A Knowledge-based Strategy For XAI: The Explanation Graph

Mauro Dragoni [a,*] and Ivan Donadello [b]

[a] *Digital Health Research Center, Fondazione Bruno Kessler, Trento, Italy*
E-mail: dragoni@fbk.eu

[b] *The KRDB Research Centre for Knowledge and Data, Free University of Bozen, Bolzano, Italy*
E-mail: ivan.donadello@unibz.it

Abstract. The interest in Explainable Artificial Intelligence (XAI) research is dramatically grown during the last few years. The main reason is the need of having systems that beyond being effective are also able to describe how a certain output has been obtained and to present such a description in a comprehensive manner with respect to the target users. A promising research direction making black boxes more transparent is the exploitation of semantic information. Such information can be exploited from different perspectives in order to provide a more comprehensive and interpretable representation of AI models. In this paper, we focus on one of the key components of the semantic-based explanation generation process: the explanation graph. We discuss its role and how it can work has a bridge for making an explanation more understandable by the target users, complete, and privacy preserving. We show how the explanation graph can be integrated into a real-world solution and we discuss challenges and future work.

Keywords: Explainable AI, Explanation Graph, Natural Language Explanations

1. Introduction

The role of Artificial Intelligence (AI) within real-world applications has significantly grown in the last years with the increasing pervasiveness of AI-based algorithmic decision making in many disciplines like Digital Health (e.g. diagnostics, digital twins) and Smart Cities (e.g. transportation, energy consumption optimization) among the others.

Unfortunately, the usage of AI-based models that rely on deep neural networks (DNN) algorithms introduced the issue that such models are *black box* in nature or, in general, it is often hard to understand why an AI-based system provides a specific output. This aspect undermines the trustworthy of such systems since in several domains it is mandatory to understand how AI-based systems work and generate decisions due to their impact on human interests, rights, and lives (e.g., decision support in credit approval or digital diagnostics).

Explainable AI (XAI) is a research area born in the sixties [1] with the aim of providing justifications about the behavior of rule-based systems. Later, the focus of the explanation systems shifted towards human-computer systems (e.g., intelligent tutoring systems) to provide better cognitive support to users [2]. In the last decade, it re-attracted a lot of attention as the need to advocate the principles mentioned above in order to promote the requirements of transparency and trustworthiness that data-driven AI-based decision-making system must have for actually supporting experts in their activities [3].

[*] Corresponding author. E-mail: dragoni@fbk.eu.

The primary reason for the renewed interest in XAI research has stemmed from recent advancements in AI and ML, and their application to a wide range of areas, as well as prevailing concerns over the unethical use, lack of transparency and undesired biases in the models. Many real-world applications in the Industrial Control System (ICS) greatly increase the efficiency of industrial production from the automated equipment and production processes [4]. However, in this setting, the use of *black boxes* is still not in a favorable position due to the lack of explainability and transparency of the model and decisions. According to [5], XAI encompasses ML or AI systems/tools for demystifying black models internals (e.g., what the models have learned) and/or for explaining individual predictions. In general, explainability of an AI model's prediction is the extent of transferable qualitative understanding of the relationship between model input and prediction (i.e., selective/suitable causes of the event) in a recipient friendly manner. The term *interpretability*, on the other hand, is more related to the mathematical relationships between between input and output. Such relationship is more suitable for AI-system developers than to the final users of an AI-system. The term *explainability* and *interpretability* are being used interchangeably throughout the literature even if they are slightly different. To this end, in the case of an intelligent system (i.e., AI-based system), it is evident that explainability is more than interpretability in terms of importance, completeness, and fidelity of prediction [6]. Based on that, we will use these terms accordingly where appropriate.

The work towards the design of a trustworthy AI-based system is not only a technical challenge due to the need of controlling if generated explanations contain sensitive data that cannot be accessed by the user consuming the explanations (e.g., medical or economic information of other people). For this reason, strategies for generating explanations have to take into account also regulators' laws like "European Union's General Data Protection Regulation" (GDPR) [1] [7] or the "US government's Algorithmic Accountability Act of 2019" [2].

This aspect is fundamental for addressing the desiderata of *Fairness*, *Privacy*, *Usability*, *Causality*, *Trust*, and *Ethics* within such automated decision making systems [8, 9]. The use of semantics would lead to meaningful explanations to users and provide insights into the rationale the AI-based system used to draw a conclusion [9]. Indeed, four main aspects of semantic technologies lead to the generation of meaningful explanations:

- **Shared human-machine vocabulary**: machine uses the same concepts used in natural language.
- **Reasoning**: humans can trust the machine as reasoning services ensure that explanations respect ethical and legal norms.
- **Rules**: explain the relation between an object and its attribute detected by a machine, i.e, not only correlation but causality. Fairness: rules can formalizes the concepts accepted by bias, Trust: some rules can encode the type of representation that best fits user's needs/features.
- **Structured knowledge**: symbols linked between knowledge bases (or even knowledge graphs) ensure that explanations have a structure that can be queried and represented into different ways.

In this paper, we target the challenges described above, as well as the benefits, of integrating knowledge into the design of strategies supporting the generation of explanations adhering to the principles of the transparency of the AI-based system, the understandability of the content with respect to the end user, and the privacy preservation of the delivered content. Among the different types of explanation generation strategies mentioned in Section 2, we want to highlight how the usage of a knowledge-based solution, may be one of the most suitable alternative for satisfying the requirements mentioned above. We present a use-case showing how an ecosystem of ontologies can be linked together for enabling the generation of effective, appropriate, and privacy-by-design explanations in complex multi-actors scenarios.

The paper starts by surveying, in Section 2, the most significant categories of explanation generation and some strategies for their rendering into a human understandable format. Then, we present in Section 3 the concept of **explanation graph** and its crucial role towards the design of transparent AI-based systems. While, in Section 4 we show how the **explanation graph** can be rendered into a comprehensive natural language message. Section 5 describes a use cases where the **explanation graph** has been applied by discussing pros and cons of their integration for supporting a transparent-by-design solution. In Section 6, we summarize lessons learned and future challenges about the integration of explanation graphs within AI-based system. Finally, Section 7 concludes the paper.

[1] https://www.eugdpr.org
[2] https://www.senate.gov

2. Related Work

Explainable AI (XAI) has been widely investigated in the last years [10], but most of the contributions limits only on the analysis of how learning models (or black boxes) learn or predict. This limited view rarely exploits domain knowledge that, if properly integrated with data and model analysis, is able to achieve a full-fledged explainable system [11]. Recent works further discussed the topics by stating that the integration of Semantic Web technologies [12–14] with machine learning systems is the key for designing a completely explainable AI system.

Explanations are often categorized along two main aspects [15, 16]: (i) *local explanations* versus *global explanations*, and (ii) *self-explaining* versus *post-hoc explanations*.

Local explanations relate to individual prediction and they provide information or justification for the model's prediction on a specific input. *Global explanations* concern to the whole model's prediction process and they provide similar justifications by revealing how the model's predictive process works, independently of any particular input. Instead, *self-explaining* explanations emerge directly from the prediction process which may also be referred to as directly interpretable [17] and they are generated at the same time as the prediction by using information emitted by the model as a result of the process of making that prediction. Examples of models belonging to this category are decision trees and rule-based models. Finally, *post-hoc explanations* require post-processing since additional operations to generate the explanations are performed after the predictions are computed. LIME [18] is an example of producing a local explanation using a surrogate model applied following the predictor's operation.

Beyond the challenge of generating an explanation, and AI-based system has also to decide how to present it depending on the end user that will consume it. The capability of deciding which is the most appropriate way to render an explanation is crucial for the overall success of a XAI approach. The literature presents three main categories of approaches for rendering the generated explanations. The first category is *saliency-based representations* [3] that are primarily used to visualize the importance scores of different types of elements in XAI learning systems, such as showing input-output word alignment [19], highlighting words in input text [20] or displaying extracted relations [21]. Saliency-based representations were the first strategies used for rendering explanations and they became very popular since they are frequently used across different AI domains (e.g., computer vision [22] and speech [23]).

The second category is represented by *raw declarative representations*. This visualization technique directly presents the learned declarative representations, such as logic rules or trees, by using, as suggested by the name, the corresponding raw representation [24]. The usage of these techniques implies that end users can understand the adopted specific representations.

Finally, the third category concerns the exploitation of *natural language explanations*. This type of explanations consists in their verbalization by using human-comprehensible natural language. The actual content of each explanation can be generated by using data-driven strategies (e.g. deep generative models) [25] or by using simple template-based approaches [26]. The latter are often combined with knowledge-based techniques where, based on the target users, a proper terminology is selected [27].

Semantic Web technologies enable the design of strategies for the rendering of explanations in natural language format [28, 29]. Explanations here are provided through textual rule-like notation. In addition, Natural Language Generation (NLG) has been exploited also for generating natural language utterances from triples [30] and for translating SPARQL queries into a natural language form that can be understood by non-expert users [31]. Here, we focused on the linking of machine learning with semantic information as enabler for both improving the comprehensiveness of XAI systems and underlying machine learning itself.

Explanations in the knowledge representation and reasoning community are implemented with two orthogonal approaches: *justifications* and *proofs*. The former computes the minimal subset of the axioms in an ontology that logically entails a given axiom. The latter computes also all the inference steps [32].

As explanations are necessary for improving the trust of users in an AI system, some works deal with user studies. The work in [33] deals with explanations for entailments of OWL ontologies. The authors investigated the effectiveness of different types of explanations for explaining unsatisfiable classes in OWL ontologies. Experiments proved that the subjects receiving full debugging support performed best (i.e., faster) on the task, and that users approved

[3] Within many works in the literature are referred also as feature importance-based explanations

of the debugging facilities. Similarly, in [34] a user study to evaluate an explanation tool is performed. However, the authors did not report any detailed analysis of the difficulty users had at understanding the provided explanations. In [35] the authors presented a user study that evaluates a model-exploration based approach to explanation in OWL ontologies. The study revealed that the majority of participants could solve specific tasks with the help of the developed model-exploration tool, however, there was no detailed analysis of which aspects of the ontology the subjects struggled with and how they used the tool. The work in [36] presents a set of algorithms for computing all the justifications of an entailment in a OWL-DL ontology. However, the capabilities of the computed justifications of the logical entailments are not assessed through any study or user evaluation The authors of [37] developed and implemented a framework that translates SWRL rules inconsistencies into natural language utterances. This returns explanations, through justifications, of the disclosure of personal data to patients and staff of hospitals. The SWRL rules translation is performed axiom by axiom, thus generating a quite long sentence. The side effect is that this representation could require too much user's time for reading and understanding.

The other form of explanation in the knowledge representation reasoning community regards the formal proofs. The work in [38] developes a tool that provides proof-based explanations for entailments of the CLASSIC system. All the intermediate steps are omitted but further filtering strategies are provided in order to generate short and simple explanations. In [39] a proof-based explanation system for knowledge bases in ALC [40] Description Logic is proposed. Here, proofs in sequent calculus style are generated by using an extension of a tableaux reasoning algorithm. The proofs are then enriched to create natural language explanations. However, no user studies to assess the effectiveness of these proofs are performed. Explanations can be rendered also with the use of visualizations (tree, graphical, logical and hybrid), as performed in [41]. Here, defeasible logic proofs are rendered through several visualizations and a user study is performed in order to assess the impact of the different approaches. However, these representations are hard to understand for non-expert users. Indeed, the participants to the study have notions of logic representation. They have attended a Semantic Web course or come from the research staff. In general, Tableau techniques [40] are used for proof algorithms for Description Logic whereas the field of Automated Reasoning [42] provides proof algorithms for other families of logics.

This set of approaches to explanation of logical entailments focuses more on the study of efficient algorithms than on effective algorithms for common users. In all the mentioned works, the computed explanations consist of sets of logical axioms that can be understood only by expert users. The aim of our work is to develop a framework whose explanations are in a semantic format that can be easily rendered in an effective representation for all users. This representation can be a verbalization in natural language (performed with methods that translate axioms of an OWL ontology in Attempto Controlled English [43, 44] or in standard English [45]) or a graphical representation. In addition, the synergies between learning models and semantic information in the presented framework allow the automatic post-hoc analysis of the explanations in order to extract from the semantic features training bias or errors. This information can be used as a feedback for refining the output classification.

Our work starts from the adoption of *natural language explanations* with the aim of designing explanations generation pipelines able to exploit knowledge-based representations and repositories. Such pipelines support the generation of texts tailored to the knowledge capabilities of the target users and by preserving also privacy and ethical aspects connected with the information to deliver.

3. From Knowledge Fragments to Explanation Graph

Knowledge graphs represent one suitable road towards the design of a trustworthy AI-based solution [14]. As introduced in Section 1 and discussed in Section 2, AI-based systems provide output that has to be processed before making it comprehensive by target users. This issue can have different levels of severity depending on the type of AI-based system adopted. By surveying the literature, the two antipodes are neural-based systems and rule-based systems. The former are traditionally considered as *black box* since their transparency in understanding why a specific classification has been provided is generally low. This aspect is amplified if input features do not have a conceptual meaning that can be exploited in a later stage. The latter, on the one hand, are transparent since it is possible to backtrack all possible choices performed by the system, but, on the other hand, their interpretability can be very complex given the amount of rules that have to be checked.

The scenarios in which an AI-based system is adopted can exacerbate the trustworthy need. As example, within the medical domain a clinician must know why a system provides a specific recommendation for a given patient (e.g., which medical information it used, which patient's data, how they have been combined), or, within the predictive maintenance domain where technicians must know why the system suggests to perform a specific maintenance operation. In these cases, the output of the AI-based system has to be linked with further information (e.g., external knowledge) supporting the completeness and the comprehensiveness of the explanation provided to the target users.

A graph-based visualization representation helps towards this goal since it aims to convert the output generated by AI-based systems, usually provided in a structured or semi-structured format, into a graphical representation. In turn, such a representation would enable the design of different strategies for transforming the provided outputs into a representation that can be easily understand and consumed by the target user.

Explanations generated starting from structured formats such as the one mentioned above help users in better understanding the output of an AI system. A better understanding of this output allows users to increase the overall acceptability in the system. An explanation should not only be correct (i.e. mirroring the conceptual meaning of the output to explain), but also useful. An explanation is useful or actionable if and only if it is meaningful for the users targeted by the explanation and provides the rationale behind the output of the AI system [46, 47]. Explanations are meaningful if they are easily understandable by the targeted audience depending on the context in which the explanations are received. For example, if an explanation has to be provided on a specific device, such a device represents a constraint to be taken into account for deciding which is the most effective way for generating the explanation. Such explanation can be in natural language/vocal messages, visual diagrams or even haptic feedback.

The end-to-end explanation generation process, from model output to an object usable by the target users, requires a building block in the middle supporting the rendering activity. Such rendering requires explanations having a formal representation with a logical language equipped with predicates for entities and relations. This formal representation can be directly represented as an *explanation graph* with entities/nodes and relations/arcs. An **explanation graph** is a conceptual representation of the structured output provided by the **black box** model where each element of the output is represented by means of conceptual knowledge enriched with further information gathered by external sources. The *explanation graph* has two main characteristics making it suitable to be integrated in several complex domains. First, through the connections with further possible knowledge bases, it auto-enhance itself with other concepts from domain ontologies or Semantic Web resources. Second, the adopted representation format already provides an easy render in many human-comprehensible formats that can be understood also by less expert actors. Such an explanation graph can be easily obtained from the XAI techniques explained above. The explanatory features and the output class provided, for example, by a SHAP model [48, 49] can be regarded as the nodes of the explanation graph, whereas arcs are computed on the basis of the SHAP features values. The *explanation graph* can also work as bridge for accessing different types of knowledge usable, for example, to enrich the content of natural language explanations by respecting privacy and ethical aspects connected with the knowledge to use.

Figure 1 shows the abstract representation end-to-end strategy to support the generation of comprehensive natural language explanations by starting from knowledge fragments integrated into the *explanation graph*.

The end-to-end pipeline is composed by two main phases: (i) the generation of the *explanation graph* and (ii) the exploitation of the *explanation graph* to generate natural language explanations.

In the next subsections, we show a methodology to build the *explanation graph* by integrating different knowledge fragments (Section 3.1) and a narrative showing how the methodology can be applied within a concrete setting (Section 3.2). While, in Section 4, we present the rendering process enabling the generation of a natural language explanation from the created *explanation graph*.

3.1. Explanation Graph Building Methodology

As introduced above, the *explanation graph* provides and integrated representation of the knowledge fragments produced by three main sources of information: (i) the knowledge fragment provided by the output of the AI-based system; (ii) the knowledge fragment provided by public knowledge sources; and, (iii) the knowledge fragment provided by private knowledge sources. All such knowledge fragments contribute to provide an exhaustive representation of the context needed for creating a complete explanation to the target users. We present below a three-phase methodology for building an *explanation graph* based on the knowledge fragments mentioned above.

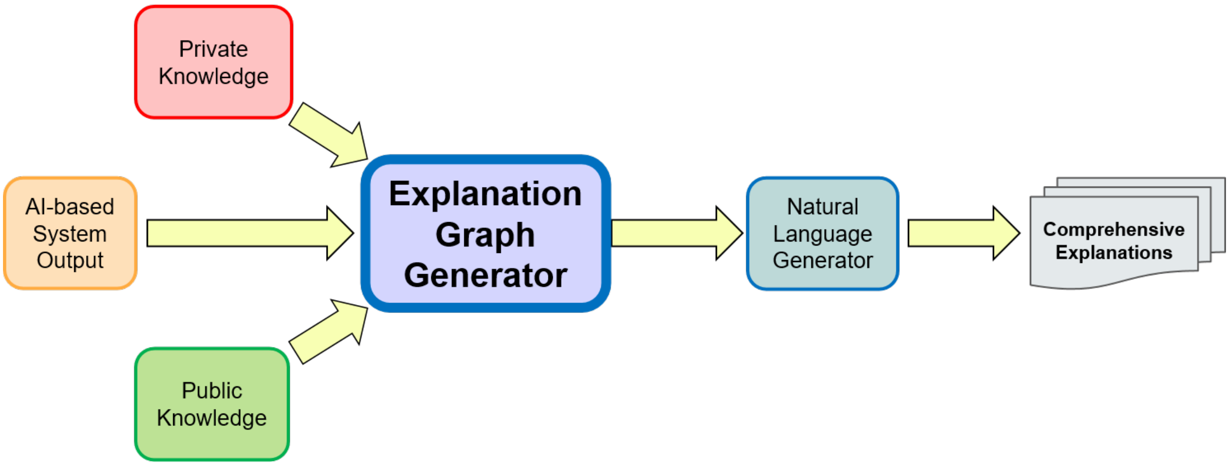
Private Knowledge
AI-based System Output
Explanation Graph Generator
Natural Language Generator
Comprehensive Explanations
Public Knowledge

Fig. 1. Summary of the end-to-end pipeline transforming knowledge fragments (i) generated by the AI-system; (ii) provided by public knowledge bases; and, (iii) provided by private knowledge bases, into the explanation graph object that is later transformed into a comprehensive natural language explanation through a NLG component.

Knowledge Fragment From The AI-based System Output The first knowledge fragment is the one used to initialize the *explanation graph* and it is provided by the transformation of the AI-based system output in a set of concepts having a precise semantic meaning within the domain. This operation can be performed thanks to *semantic features*. *Semantic features* work as bridge between the output of an AI-based system and the generation of the whole *explanation graph* describing the explanation's domain. *Semantic features* are strongly related to the output of the AI-based system due to the manual definition of relationships having a precise semantic meaning enabling the initialization of the *explanation graph*.

The definition of the relationships between the output of the AI-based system, the semantic features, and the concepts between an underlying ontology is a manual effort, in some cases time consuming, that will be the most important challenge to address in the future in order to foster the creation of this type of hybrid systems.

The amount of such an effort may vary depending on the type of AI-based system is deployed. Hence, it is necessary to highlight an important difference between the two main categories of AI-based systems: *symbolic systems* and *connectionist systems*.

The transforming operation for a *symbolic system* is a straightforward action since, by design, the semantic features used within the implemented reasoning strategy are defined unambiguously in an ontology. Hence, the construction of the *explanation graph* starts from the list of semantic features contained in the AI-based system that are directly mapped to ontological concepts.

Different and more complex is the case of *connectionist systems* since it is necessary to define alignments between the features given as input to the *connectionist systems* and the conceptual knowledge used as the basis for building the explanation graph. This operation follows the definition of *comprehensible system* provided by Doran et al. [9], that is a system that computes its output along with symbols that allow users to understand what are the main *semantic features* in the data that triggered that particular output. Here, we refine the work of Doran et al. by introducing the concept of semantic feature. These are features that can be expressed through predicates of a First-Order Logic (FOL) language and represent the common and shared attributes of an object/phenomenon that allow its recognition. Examples can be $ContainsBacon(x)$ or $ContainsEggs(x)$ indicating the ingredients of a dish in a picture. Semantic features in principle can be further explained by more fine-grained semantic features. For example, the $ChoppedBacon(x)$ feature can be explained by the $HasCubicShape(x)$ and $HasPinkColor(x)$ features. However, in a nutritional domain, these latter features do not add further comprehension to users and can represent an overload of information. Therefore, the knowledge engineering and/or domain expert have to select the right granularity of the semantic features to present to users and therefore ensuring a sort of atomic property of these features. Semantic features are different from the learnt numeric (and not comprehensible) features of a *connectionist systems*. The aim of a comprehensible system is to find an alignment between the learnt and the semantic features.

The connection between a *connectionist systems* output and its semantic features is formalized through the definition of *comprehension axiom*.

Definition 1 (Comprehension axiom)**.** *Given a FOL language with $\mathcal{P} = \{O\}_1^n \cup \{A\}_1^m$ the set of its predicate symbols, a comprehension axiom is a formula of the form*

$$\bigwedge_{i=1}^{k} O_i(x) \leftrightarrow \bigwedge_{i=1}^{l} A_i(x)$$

with $\{O\}_1^n$ the set of output symbols of a connectionist systems and $\{A\}_1^m$ the corresponding semantic features (or attributes).

A comprehension axiom formalizes the main tasks of a *connectionist systems*:

Multiclass Classification: the predicate $O_i(x)$ represents a class (e.g., pasta with Carbonara sauce or sushi) for x and $k = 1$ as a softmax is applied in the last layer of the *connectionist systems*. The semantic features represent, for example, ingredients contained in the recognized dish.

Multilabel Classification: $O_i(x)$ is part of a list of predicates being computed by the *connectionist systems* (e.g., dinner and party) for x and $k > 1$ as a sigmoid is applied in the last layer of the *connectionist systems*. The semantic features represent, for example, objects in the scene, such as, pizza, table, bottles, person and balloons.

Regression: $O_i(x)$ can be part of a list of predicates being computed by the *connectionist systems* (e.g., the asked price and the real values of house) for x. Here $k \geqslant 1$ with a sigmoid applied in the last layer of the *connectionist systems*. The semantic features are properties of interest for buying a house.

Once a set of comprehension axioms is returned by our comprehensible system, the former can be easily transformed into a graph representation where the nodes are the unary predicates O_i and A_i plus other information such as a possible neural network scores for these predicates. The edges are the logic relations between these predicates, such as implications and n-ary predicates with $n > 1$. A single comprehension axiom can be represented as a star-shape graph with O in the center, A_i at the end of the branches and the biimplications as edges.

Knowledge Fragment Public Knowledge The second knowledge fragment relates to the knowledge that can be extracted from public knowledge bases and it is used to perform a first extension of the *explanation graph*. By starting from the concepts generated from the AI-based output, ontology matching [50] strategies can be used to define alignments between such concepts and possible candidates defined within the selected external ontologies. These alignments allow to enhance the *explanation graph* with further knowledge that can be useful to generate the final explanation. Indeed, information contained within external knowledge bases may provide the proper context to better justify the content of the explanation. Examples of such knowledge bases are the Linked Open Data (LOD) Cloud [4] or other publicly available ontologies and knowledge graphs (e.g., ontologies available on BioPortal [5]).

There are two main aspects that are important to highlight in this phase. First, the selection of the ontologies for the alignment operation. Not all ontologies publicly available are suitable for enhancing the *explanation graph*. For example, if the AI-based system has to recommend if a maintenance operation has to be performed on a weather station, the attempt of extending the *explanation graph* with concepts extracted from an ontology about agriculture is useless. Ideally, the experts in charge of deploying and activating the AI-based system should select a group of ontologies to use for performing the extension of the *explanation graph*. Like it has been mentioned above, also this operation can be time consuming. However, it is warmly recommended to perform this task manually in order to avoid the analysis of many out-of-domain ontologies.

Second, the effectiveness of the alignment algorithm. The literature discusses many ontology alignment strategies [51] performed differently on different benchmarks. Clearly, the effectiveness of the alignment algorithm adopted for extending the *explanation graph* affects, in turn, the quality and the appropriateness of the generated

[4] https://lod-cloud.net/

[5] https://bioportal.bioontology.org/

explanations. What it has been discussed in the first point, may drive the alignment algorithm to define correct alignments and, in this case, alignment strategies favoring the precision metric, rather than the recall one, are preferable.

Knowledge Fragment Private Knowledge Finally, the third knowledge fragment integrated into the *explanation graph* concerns private information associated with the user should receive the explanation. Such private information can be exploited for tailoring the generated message with respect to the user profile or to decide which user, among different alternatives, should receive the explanation (e.g., the clinician rather than the patient).

Operations performed in this last phase are the same described in the previous paragraph, i.e., the execution of ontology alignment operations, with the difference that the private knowledge associated with the user is aligned with the content of the *explanation graph*. This way, it is possible to tailoring the content of the final natural language explanation with respect to the user profile. During this phase, privacy constraints are possibly introduced within the *explanation graph* if some information are accessible by a subset of possible target users.

In the next section, we show a running example showing how these phase may be applied in a concrete setting.

3.2. Example of Explanation Graph Construction

In order to make the *explanation graph* building process more comprehensive we present in this section a running example showing how the *explanation graph* is initialized with the starting knowledge fragment and then enriched with the knowledge fragments coming from both public and private knowledge bases.

Let us consider a scenario occurring within the healthcare domain where patients suffering from a chronic nutritional disease are monitored by a digital assistant system in charge of providing recommendations about healthy behaviors (i.e., diet and physical activities) based on what patients ate and which activities they did. The digital assistant interacts with both clinicians and patients that are the two types of target users occurring in the scenario. Hence, when an explanation is sent out, depending on the target user to reach, both content and language have to be tailored with respect to her. A patient is associated with a set of guidelines that she should follow to maintain a healthy status and to avoid disease exacerbation and that are defined within the private knowledge associated with her. When an undesired behavior is detected (i.e., a guideline is violated), the digital assistant has to generate, as mentioned above, two different explanations: one for the cliniciancontaining medical information linked with the detected undesired behavior including also possible severe adverse consequences; and one for the patient omitting some medical details and, possibly, including persuasive text inviting to correct the patient's behavior in the future. Where needed, privacy issue should be managed in order to avoid the delivery of sensitive information to an unauthorized user. Indeed, in general, not all personal information of patients can be delivered in the generated explanations and the selection of the proper ones are demanded to the constraints defined within the AI-based system.

The first phase is to initialize the *explanation graph* by creating a concept for each semantic feature and by instantiating the relationships among them that are gathered by the domain ontology with which semantic features are mapped. Let us consider our running example. The user can provide a picture of a food dish through a mobile application and the back-end processes it in order to detect the recipe. The back-end implements a connectionist model trained for extracting a list of recipes together with their confidence score and a gradient-based explainer motivates the classification task by providing a set of pairs $< feature, value >$ where each feature corresponds to a specific ingredient contained within the udnerlying knowledge base [6] . The *explanation graph* associated with this specific event includes first concepts representing the event itself and the output of the classification task. Figure 2 shows how first concepts are created within the *explanation graph* by starting from the output of the AI-based system. We provided a limited set of concepts by purpose in order to preserve the readability of the image. It is important to notice that the gray concepts are the ones that are part of an underlying ontology, deployed within the AI-based system, describing the scenario itself and that it is not part of the public neither the private knowledge used for building the *explanation graph*. In this case, when an undesired behavior is detected, the *explanation graph* is populated with a concept *Violation* representing the undesired behavior and a concept *MonitoringRule* representing the guideline that has been violated. The *Violation* concept is then linked with the food category generated the

[6] For simplicity, we omit here all the aspects related to the validity of the classification provided and the fact that from the ingredients that have been detected it would be possible to assess the confidence of the recipe.

Violation. In this example, the *ColdCuts* food category that has been detected during the recipe image classification. Then, since the AI-based system is in charge of monitoring people behavior, the *User* concept is generated as well since the *Violation* refers to her.

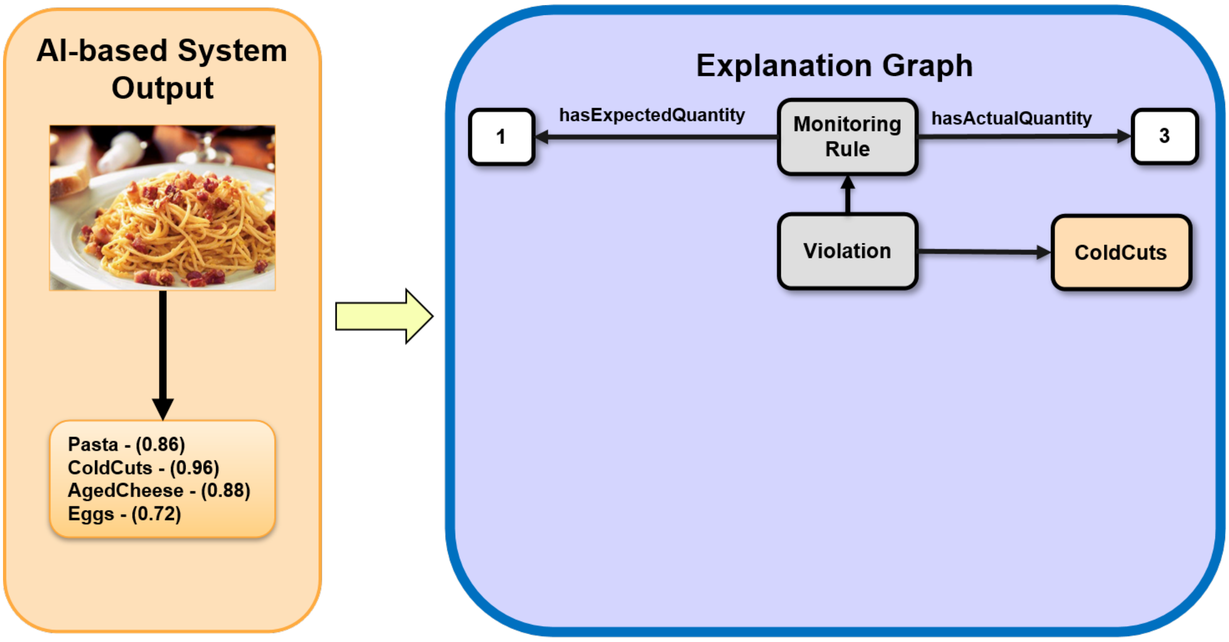

Fig. 2. Transformation of the output provided by the AI-based system in the first knowledge fragment included in the final explanation graph.

The second phase consists in the defining the alignments between one or more concepts contained in the *explanation graph* and concepts defined into external knowledge bases publicly available. Here, the concepts of interest are the ingredients detected by the classifier. Hence, for each concept representing one of the ingredients contained in the recognized recipe, it is possible (i) to associate nutritional information about both positive and negative properties of them, (ii) to extract correlations between each ingredient and the onset or exacerbation of specific nutritional diseases, and (iii) to include nutritional-related relationships between the detected recipe and other foods based on their nutritional information. This information can be extracted from knowledge sources like the Linked Open Data (LOD) cloud, the UMLS[7] knowledge base, or from domain-specific ontologies providing nutritional-related information with a high level of granularity, like, in this specific case, the HeLiS ontology [52]. Within our example, by starting from the *ColdCuts* concept it is possible to perform the following operations:

- to extract from the HeLiS ontology the list of the nutrients contained within the *ColdCuts* food category (e.g., *AnimalFats*, *Salt*);
- to find within the LOD cloud other knowledge bases defining the *ColdCuts* concept (e.g., AGROVOC [8]) that can be used as starting point for exploring the cloud to look for further relevant knowledge;
- to exploit the alignments between the disease taxonomy defined in AGROVOC and the UMLS knowledge base in order to extract possible correlations between food categories and nutritional diseases (e.g., the excess of *ColdCuts* consuption may lead to *Cardiovascular* diseases.

Figure 3 shows how the explanation graph is extended with new concepts extracted from publicly available knowledge bases. For brevity, we report only the concepts that are relevant with respect to our running example.

[7] https://www.nlm.nih.gov/research/umls/index.html
[8] http://aims.fao.org/vest-registry/vocabularies/agrovoc

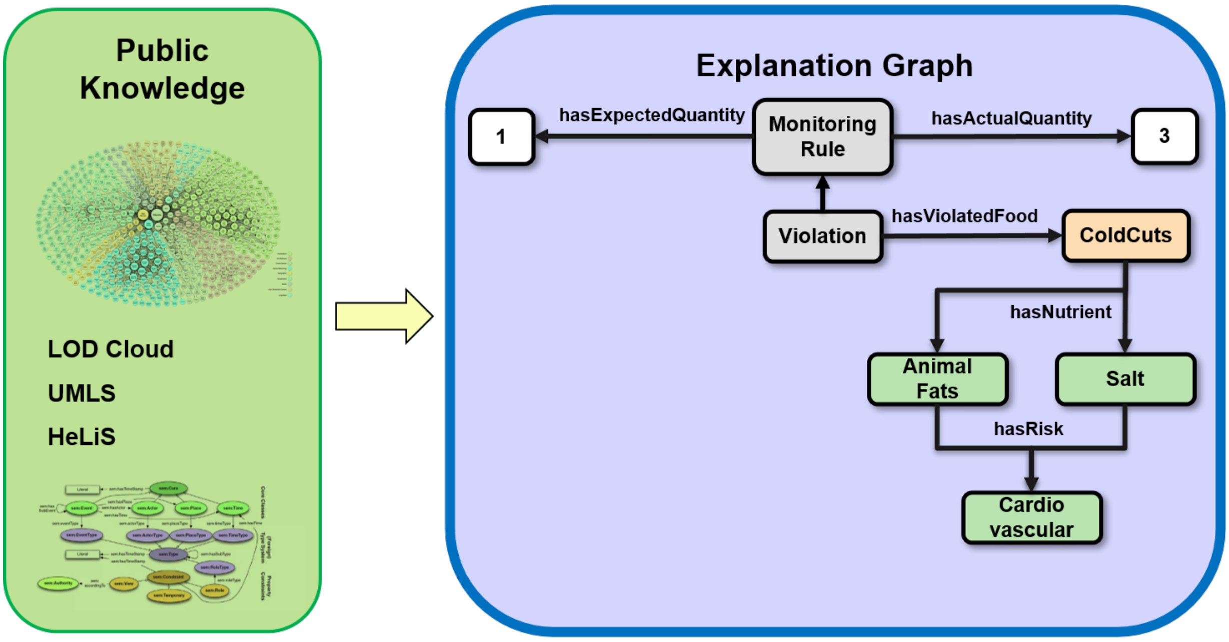

Fig. 3. Enrichment of the explanation graph by introducing knowledge coming from public resources.

Finally, in the third phase, the *explanation graph* is extended with knowledge fragments extracted from private knowledge bases, e.g., the structured representation of a personal health record (PHR) or the financial status of a person. By considering our running example, the knowledge included within the patient's PHR can be linked to the nutritional information already included in the *explanation graph*. By analyzing the PHR of our user, we may extract both the age and possible ongoing diseases (e.g., *Hypertension*). Then, thanks to the integration of the UMLS knowledge base, we may infer that the *Hypertension* is a sub-concept of *Cardiovascular* diseases. Hence, we can instantiate the *isA* relationship and, in turn, to exploit it to reinforce the explanation provided to the target user by including further reasons about why the detected behavior is undesired. In particular, as it will be shown in the use case provided in Section 5, the natural language generator relies this knowledge to decide which linguistic strategy to adopt for generating the explanation. Finally, depending on the target user, all the content generated by starting from this information may be excluded from the rendered explanation due to privacy constraints defined within the target user's profile. Figure 4 shows the last two concepts added into the *explanation graph* and the relationships instantiated with existing concepts.

The result of this process is the *explanation graph* object containing the structured, and semantically validated, knowledge generated by starting from the output of the AI-based system extended with further knowledge coming from both public and private knowledge bases. Both the domains to consider for building the *explanation graph* as well as the amount of knowledge to include is left to the AI-system engineer depending on the desired granularity of the explanations. Figure 5 shows the *explanation graph* generated for our running example.

In the next Section, we discuss a possible strategy to convert the *explanation graph* into a natural language explanation. This aspect emphasizes the role of the knowledge into this process since, without having a complete understanding of the seed information to use for generating explanations, it would not be possible to design a trustworthy AI-based system.

4. From Explanation Graph to Explanation Rendering

A described in Section 3, the **explanation graph** works as a bridge between the signals produced by a **black box** model or a symbolic approach to an understandable rendering of such signals. Producing such explanation carries

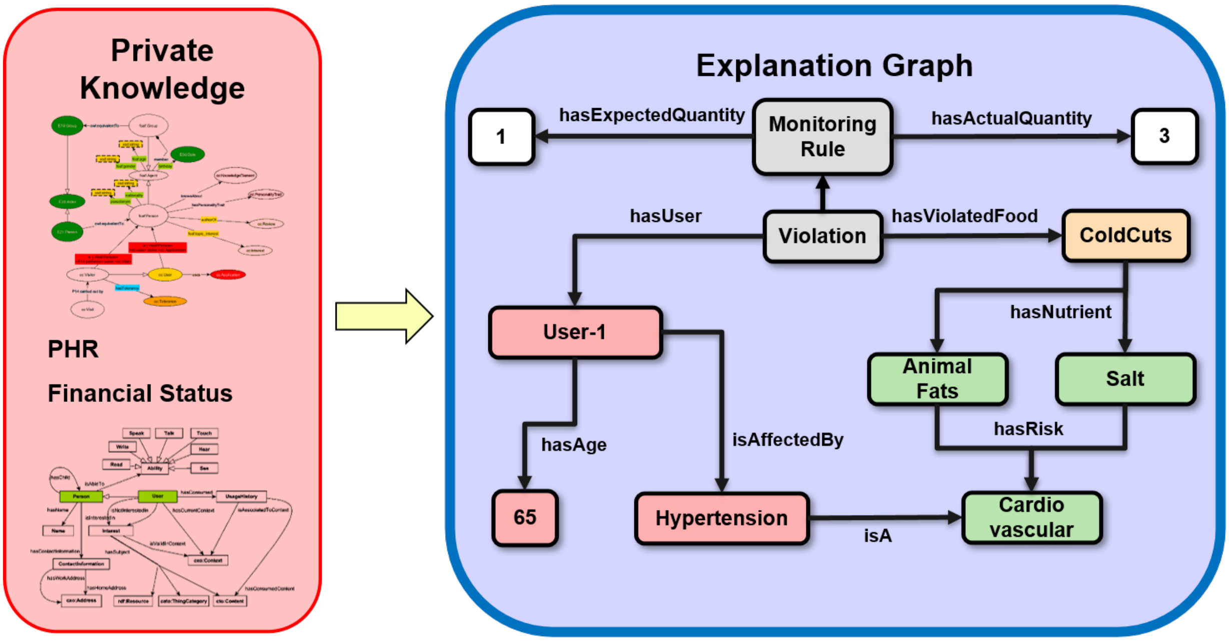

Fig. 4. Enrichment of the explanation graph by introducing knowledge coming from private resources.

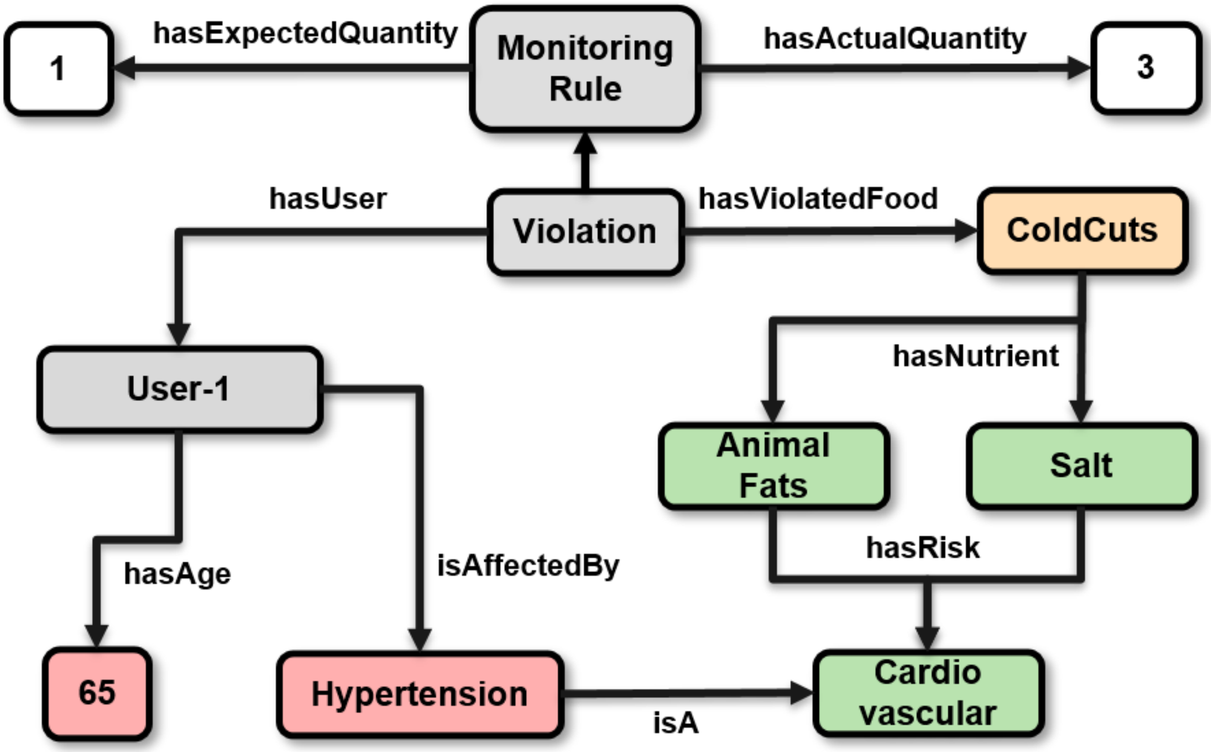

Fig. 5. *Explanation graph* for users exceeding in cold cuts consumption in the diet & healthy lifestyle adherence application. Concepts with the orange background belong to the knowledge fragment generated from the AI-based system output. Concepts with the green background belong to the knowledge fragment generated from public knowledge bases. Finally, concepts with the red background belong to the knowledge fragment generated from private knowledge bases.

a challenge, given the requirement of adopting a proper language with respect to the targeted audience [47] and the related context. The *explanation graph* provides a formal (graph-like) representation to be easily rendered and personalized through natural language text [53]. Terms in the *explanation graph* encode the rationale behind the AI-based system decision, whereas the domain knowledge base encodes further terms that help the user's comprehension by: (i) enhancing the final rendered explanation with further information about the output; and, (ii) using terms or arguments that are tailored to that particular user and increase the comprehension of the explanation. The generation of such natural language explanations can rely on a generation pipeline composed by three elements described below and shown in Figure 6.

1. The rendering component. This component is in charge of transforming the *explanation graph* into its equivalent natural language form enhanced with contextual information tailored to both the domain and the target user.
2. The *explanation graph* build by starting from the output of the AI-based system as described in Section 3.
3. The strategy to adopt. A strategy encodes the *what*, *when*, and *how* of a natural language explanation.

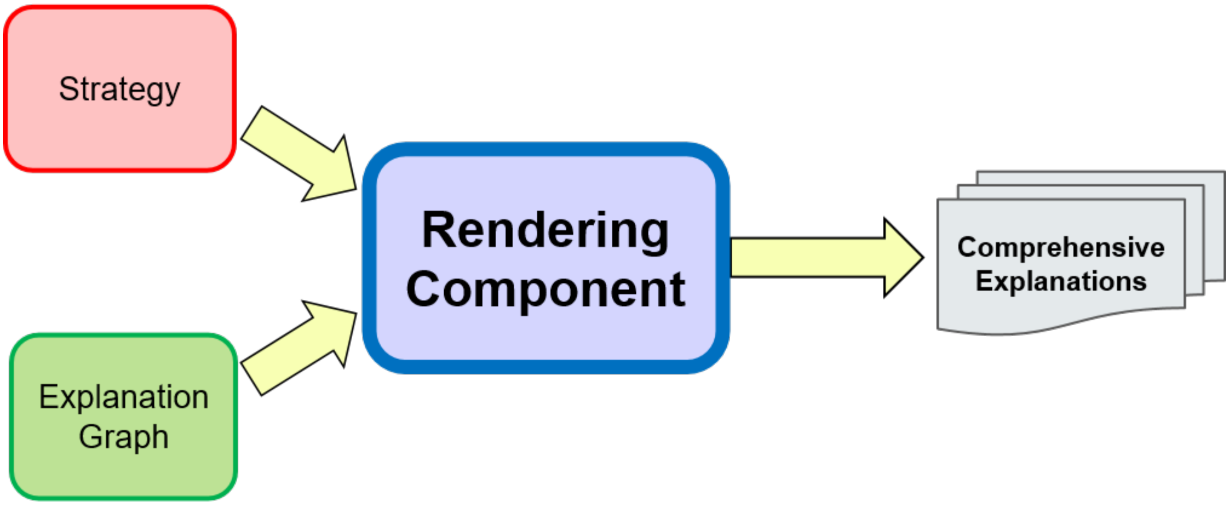

Fig. 6. Abstract view of the rendering process.

The rendering component is a machinery receiving as input the *explanation graph* to render and the strategy to adopt. The rendering component itself may be considered as a flexible container that is agnostic with respect to both the domain, the content of the explanation to render, and the strategy to adopt. Depending on the internal structure of the rendering component and by the implemented strategy, the linguistic realization is performed through several steps that may be different from one instance to another [54]. Indeed, the rendering component integrate the methodology adopted for constructing the structure of the actual natural language explanation. One of the possible methodology to render an *explanation graph* into a natural language text is to adopt a template-based system. Templates are formal grammars whose terminal symbols are a mixture of terms/data taken from the nodes/arcs of the *explanation graph* and from a domain knowledge base.

The core of the rendering operation is represented by the strategy integrated into the rendering component. We said above that a strategy encodes the *what*, *when*, and *how* of a natural language explanation.

The *what* is the content that has to be included into the message. The content is generated by starting from the *explanation graph* and, depending on both the scenario and the target user, it may vary in terms of which part of the *explanation graph* should be published or not. For example, if there are privacy constraints associated with the target user, some nodes of the *explanation graph* are not considered during the rendering process. Other examples are emergency scenarios where the generation process has to be as fast as possible: here, only the core nodes of the *explanation graph* are took into account in order to include the minimum amount of knowledge able to provide a complete explanation to the target user.

The *when* represents the *timing* adopted to provide the natural language explanation to the target user. The timing aspect is directly connected to the context of the explanation and it is used to decide when a generated explanation as to be provided to the target user. For example, in an emergency scenario it is straightforward to guess that the explanation has to be sent as soon as possible. In other scenarios the timing aspect may consist in a precise schedule that the AI-based system must follow to communicate with the target user, e.g.: within a predictive maintenance scenario, the AI-based system may provide every two hours a report containing the status of the monitored device by explaining if a maintenance operation is required (or not) and why.

Finally, the *how* is related to the the communication technique to adopt when the actual generation of explanation's content is performed. Communication techniques may range from short and prescriptive messages, used for example in case of an emergency, to more complex, persuasive, and argumentative messages when the explanation does not have only the goal of providing a specific information, but also to introduce elements aiming to trigger a behavior change in the target user.

Before presenting the use case described in Section 5, we mention some possible scenarios giving an idea about how the encoding of the strategy can be different with respect to their context and aim.

- High-risk scenarios (e.g., volcanoes, earthquakes, flooding). An AI-based system is in charge of processing real-time data provided by a network of sensors and to inform target users about emergency situations. The content of the explanations should be as much short and clear as possible. The *when* is definitely in real-time. The communication strategy has to be prescriptive.
- Predictive maintenance scenarios (e.g., industrial equipment, smart lifts). An AI-based system is in charge of collecting data from IoT devices concerning the status of a specific machinery and to process them at a regular basis in order to detect possible situations of interested that have to be notified to proper users. The content of the explanations should be as much complete as possible in order to properly report every detail about the status of the monitored machinery. The *when* is defined through a precise schedule (e.g., every hour) and it may work as trigger to the AI-based system to process the collected data and to generate the explanation. Also in this case the communication strategy may be mostly prescriptive. In some case, if some danger level is reached, the explanation may include suggested actions for the target user.
- Event-based scenarios (e.g., data input, external trigger). An AI-based system is in charge of reacting to a specific event (e.g., data provided by a user, temporal trigger) and to process data and generate explanations based on the data snapshot collected at the time the trigger is received. Here, both the content and the communication strategy are adapted depending on the specific scenario. The important aspect to manage is the *when* one since the effectiveness of the strategy strongly depends on the appropriateness of the events defined to trigger the generation of explanations.
- Privacy-based scenarios (e.g., situations where any kind of personal and sensitive data are involved in the generation of the explanations). The *explanation graphs* generated by the AI-based system includes sensitive information (e.g., economic, medical) of the user which data refers to. In these scenarios, the important aspect to manage is to decide which content to include into the generated explanation by taking into account which is the access control policies of the target users with respect to each single information. Here, the use of knowledge bases is very important since it is possible to define data access constraints directly at ontological level [55]. Hence, when part of the explanation content is generated by starting from a specific node of the *explanation graph*, the rendering component can check if the target user has the grants to access that specific node in order to decide if the textual fragment generated from that node can be included in the explanation or not. A more complex scenarios, where explanations have to be provided to different target users, this process is performed for all of them.
- Behavior change scenarios (e.g., digital therapeutics). An AI-based system is responsible of monitoring the behavior of patients and to detect if undesired situations are detected. Here, the important aspect to analyze is the communication strategy. Indeed, within behavior change scenarios, *how* the message is presented to the target user is crucial for determining the effectiveness of the entire system. The use case we present in Section 5 targets this specific scenario.

Now, in Section 5, we present a complete use cases showing how the *explanation graph* concept has been instantiated into a specific scenario by showing also the impact that it had on the behavior of target users.

5. Use Case: Natural Language Persuasive Explanations For Healthy Lifestyle Adherence

In this section, we describe a use case exploiting the *explanation graph* to generate natural language explanations towards persuading users about adhering to suggested healthy behaviors. In this use case, with the term *user* we intend both healthy citizens or patients since the use case fits well for both of them. Indeed, in the first case the goal is to preserve healthy conditions, while in the second case such guidelines can be used to monitor the behaviors of patients affected by chronic diseases or that are recovering from severe medical issues and that have to avoid relapses.

In this use case, the rendering component we integrated is the one described in [53], hereafter called *Template System For Natural Language Explanations (TS4NLE)*, and we refer to it throughout this section as a candidate solution, but not limited to it, to adopt for the actual generation of a natural language explanation. TS4NLE realizes templates that can be structured as a decision tree where the first level contains high-level and generic information that is progressively specialized and enriched according to the user's features specified in the user model. Once templates are filled with terminal terms, the lexicalization [9] and linguistic realization of the filled template are performed with standard natural language processing engines such as RosaeNLG [10].

In this use case, the TS4NLE has been instantiated integrating as strategy the persuasion model proposed by den Akker [56], and expanded taking into consideration additional strategies presented in [57], where generated messages are composed by three parts: *Feedback*, *Argument*, and *Suggestion*. Below, we present the description of the scenario and we report the results concerning the impact of the generated explanations on the behavior of a selected group of users.

As support external knowledge base, we rely on the HeLiS ontology [52]. HeLiS is a state-of-the-art ontology that formalizes the food and recipes composition, the rules of the Mediterranean diet, the physical activities domain and user preferences and habits in order to support the promotion of healthy lifestyles. The relevance of this ontology with respect to this use case pivots around the integrated model representing a fine-grained description of food (i.e., nutrients) and the correlation between nutrients and the onset or exacerbation of possible allergies or diseases.

Preamble. Each user is associated with a profile containing a set of guidelines, related to both nutrition and physical activity behaviors, that she has to follow. Once the user provides her behavioral data, the AI-based system classifies the user behavior in classes ranging from *very good* to *very bad* depending on the implemented granularity. When undesired situations (hereafter in this section called *violations*) are detected, the system has to inform all the involved stakeholders (e.g., users, clinicians, caregivers) about such violations through the use of a natural language generation (NLG) template-based strategy [27]. The communication provided has to include all information relevant to explain (i) why the violation has been detected, (ii) which data led to such a violation, and, (iii) why such a violation is dangerous for the user. Finally, in the case in which a multi-stakeholder scenario is foreseen, privacy-wise filters have to be applied to avoid the communication of sensitive information to people do not have the permission to see them. *Explanation graphs* were thought for supporting these kind of interactions in order to make the system more trustworthy by the involved stakeholders. According to the user profile (e.g., whether the user has to be encouraged or not or according to the users' barriers or capacities), the NLG component explores all possible options in order to reach the one it considers to be most effective.

A user study regarding the Mediterranean diet states that such tailored explanations are more effective at changing users' lifestyle with respect to a standard notification of a bad lifestyle [53].

As example, we still consider the *explanation graph* shown in Figure 5. Such a graph can be rendered through the NLG component as: *"This week you consumed too much (5 portions of a maximum 2) cold cuts. Cold cuts contain animal fats and salt that can cause cardiovascular diseases. People over 60 years old are particularly at risk. Next time try with some fresh fish"*.

The generation of the natural language explanation shown above is performed by the TS4NLE component by following the steps below. After the generation of the *explanation graph*, the *message composition* component

[9] *Lexicalization* is the process of choosing the right words (nouns, verbs, adjectives and adverbs) that are required to express the information in the generated text, it is extremely important in NLG systems that produce texts in multiple languages. Thus, the template system chooses the right words for an explanation, making it tailored.

[10] https://rosaenlg.org/rosaenlg/3.0.0/index.html

of TS4NLE starts the generation of three textual messages for the feedback, the argument and the suggestion, respectively.

The generation of the whole message leverages several persuasion strategies in order to compose a complex persuasive message through templates. Templates are then rendered through natural language text. A template is formalized as a grammar whose terminal symbols are filled according to the data in the violation package and new information queried in the *explanation graph* . Once templates are filled, a sentence realizer (i.e., a producer of sentences from syntax or logical forms) generates natural language sentences that respect the grammatical rules of a desired language[11]. Below we describe the implemented strategies to automate the message generation, focusing also on linguistic choices.

Explanation Feedback [13]: is the part of the message that informs the user about the not compliant behavior, hereafter called *violation*, with the goal that has been set up. Feedback is generated considering data included in the explanation graph starting from the violation object: the food entity of the violation will represent the object of the feedback, whereas the level of violation (e.g., deviation between food quantity expected and that actually taken by the user) is used to represent the severity of the incorrect behavior. The intention of the violation represents the fact that the user has consumed too much or not enough amount of a food entity. Feedback contains also information about the kind of meal (breakfast, lunch, dinner or snack) to inform the user about the time span in which the violation was committed.

The violation aligned with the terminal symbols of the template in our running example is in Figure 7.

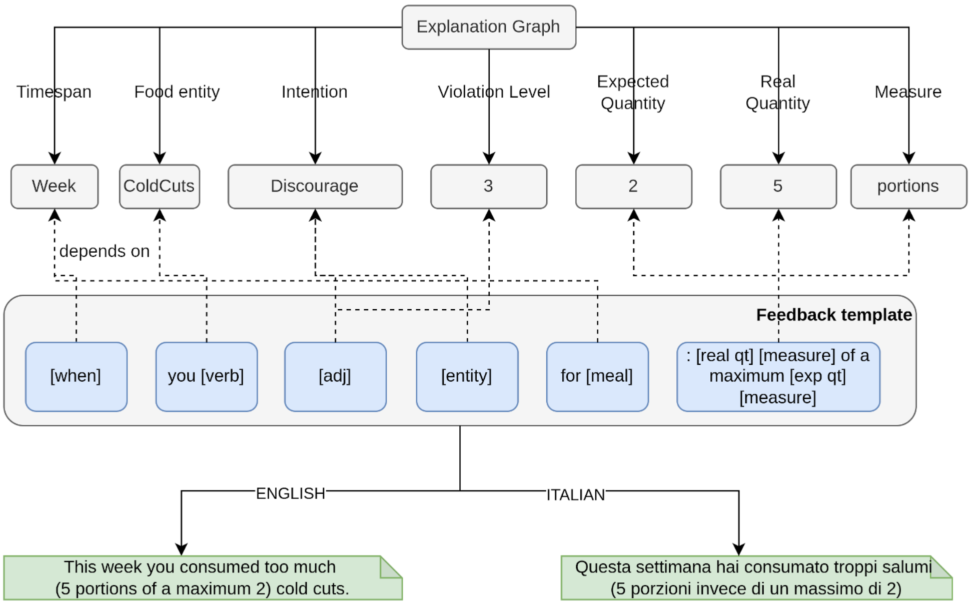

Fig. 7. TS4NLE model (template and example of violation) for generating the text of the feedback. Choices on template and message chunks depend on the violation package. This holds also for both the argument and suggestion. Dashed lines represent a dependency relation. A template of type “informative” has been used in the example.

From a linguistic point of view, choices in the feedback type are related to the verb and its tense: e.g., beverages imply use of the verb *to drink* while for solid food we use *to eat*. To increase the variety of the message, verbs *to consume* and *to intake* are also used. Past simple tense is used when violation is related to a specific moment (e.g.

[11] Current version of TS4NLE supports the generation of messages in English and Italian. In particular, Italian language requires a morphological engine (based on the open-source tool called morph-it[12]) to generate well-formed sentences starting from the constraints written in the template (e.g., tenses and subject consistency for verbs)

[13] The feedback concept in the message generation model of [56] must not be confused with the behavior change strategy element *feedback* in the BIT model.

You drank a lot of fruit juice for lunch), while present continuous is used when the violation is related to a period of time of more days and the period is not yet ended (e.g., *You are drinking a lot of fruit juice this week*).

Explanation Argument: is the part of the message informing users about possible consequences of a behavior. For example, in the case of diet recommendations, the *Argument* consists of two parts: (i) information about nutrients contained in the food intake that caused the violation and (ii) information about consequences that nutrients have on human body and health. Consequences imply the positive or negative aspects of nutrients. A A template example for supporting the generation of an explanation argument is shown in Figure 8.

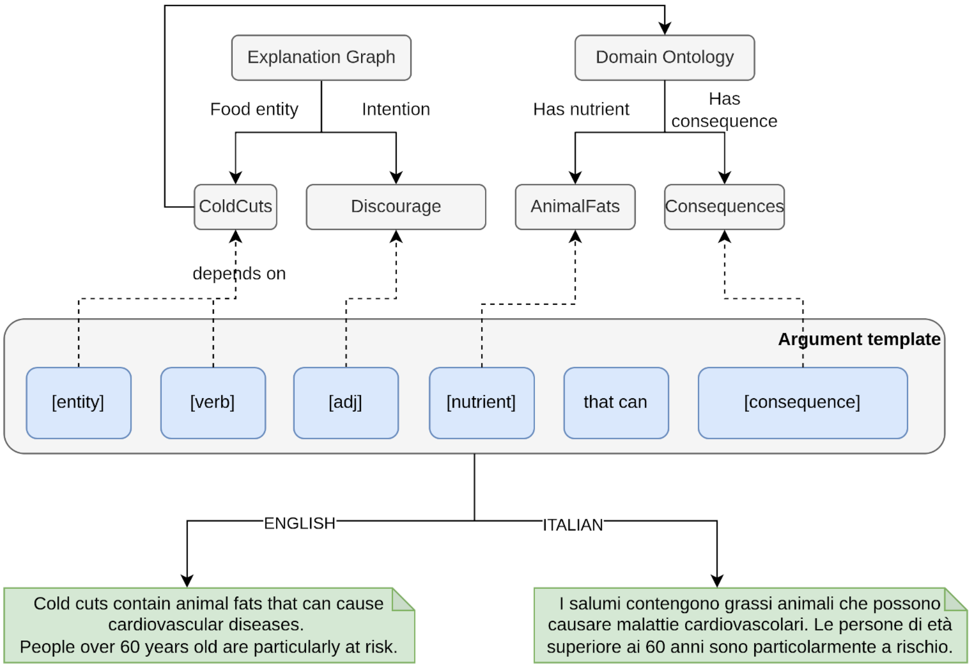
Explanation Graph
Domain Ontology
Food entity
Intention
Has nutrient
Has consequence
ColdCuts
Discourage
AnimalFats
Consequences
depends on
Argument template
[entity]
[verb]
[adj]
[nutrient]
that can
[consequence]
ENGLISH
ITALIAN
Cold cuts contain animal fats that can cause cardiovascular diseases. People over 60 years old are particularly at risk.
I salumi contengono grassi animali che possono causare malattie cardiovascolari. Le persone di età superiore ai 60 anni sono particolarmente a rischio.

Fig. 8. TS4NLE model (template and example of violation) for generating the text of the argument given as part of the explanation when violating diet restrictions.

In this case, TS4NLE uses the intention element contained in the selected violation package to identify the type of argument to generate. Let us consider the violation of our running example where the monitoring rule limits the weekly cold cuts consumption to maximum 2 portions per week since. In the presence of an excess in cold cuts consumption (translating to a discouraging intention) the argument is constituted by a statement with the negative consequences of this behavior on user health. On the contrary, the violation of a rule requiring the consumption of at least 200 gr of vegetables per day brings the system to generate an argument explaining the many advantages of getting nutrients contained in that food (an encouraging intention). In both cases, this information is stored within the explanation graph.

Moreover, TS4NLE analyzes the message history to decide which property of the *explanation graph* to use in the *Argument*, to generate a message content that depends on e.g., content sent in the past few days, ensuring a certain degree of variability. With respect to linguistic choices, the type of nutrients and their consequences influence the verb usage in the text. Finally, to emphasize different aspects of the detected violation, templates encode the use of appropriate parts of speech. For example, for stressing the negative aspects of the violated food constraint, the verb *contain* (nutrients) and *can cause* (for consequences) were used. On the other hand, positive aspects are highlighted by the verb phrase *is rich in* and verb *help* are used for nutrients and consequences, respectively.

Explanation Suggestion: this part represents an alternative behavior that TS4NLE delivers to the user in order to motivate him/her to change his/her lifestyle. Exploiting the information available within the *explanation graph*, and possibly collected from both public and private knowledge, TS4NLE generates a *post* suggestion to inform the user about the healthy behavior that he/she can adopt as alternative. To do that, the data contained in the *explanation graph* are not sufficient. TS4NLE performs additional meta-reasoning to identify the appropriate content that depends on (i) qualitative properties of the entities involved in the event; (ii) user profile; (iii) other specific violations; (iv) history of messages sent.

The model for generating a suggestion message is shown in Figure 9 where, for the sake of readability, we report only the second point of the list: compliance with the user profile.

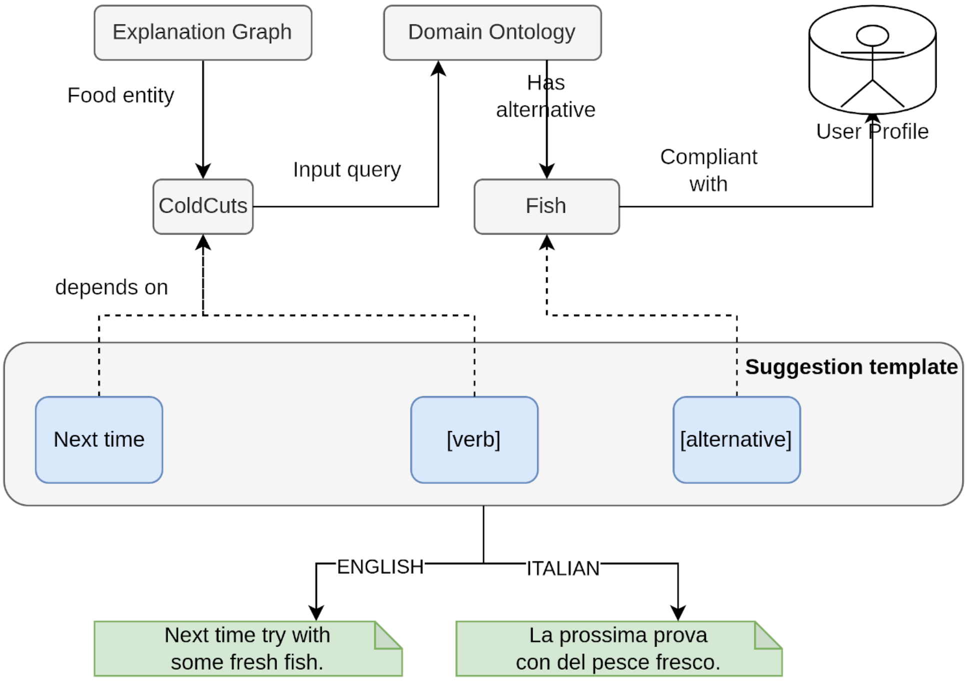

Fig. 9. TS4NLE model (template and example of violation) for generating text of the suggestion.

Continuing with the running example, first TS4NLE queries the *explanation graph* to provide a list of alternative foods that are valid alternatives to the violated behavior (e.g., similar-taste relation, list of nutrients, consequences on user health). These alternatives are queried according to some constraints: (i) compliance with the user profile and (ii) compliance with other set up goals. Regarding the first constraint, the reasoner will not return alternative foods that are not appropriate for the specific profile. Let us consider a vegetarian profile: the system does not suggest vegetarian users to consume fish as an alternative to meat, even if fish is an alternative to meat by considering only the nutrients. The second constraint is needed to avoid alternatives that could generate a contradiction with other healthy behavior rules. For example, the system will not propose cheese as alternative to meat if the user has the persuasion goal of cheese reduction.

Finally, a control on message history is executed to avoid the suggestion of alternatives recently proposed. Regarding the linguistic aspect, the system uses appropriate verbs, such as *try* or *alternate*, to emphasize the alternative behavior.

Evaluation The strategy described above has been applied in the context of the *Key to Health* project and tested during a user study last forty-nine days. This user study consisted in providing a group of users with a mobile application we created based on the services included into our solution. We analyzed the usage of a mobile application connected with our solution for the project timespan by monitoring the information provided by the users and the associated violations, if any. One of our goals was to measure the effectiveness of the persuasive messages generated by TS4NLE module by observing the evolution of the number of generated violations by the users.

A total of 120 users have been involved in the *Key to Health* project and they have been split in two groups. We used a non-randomized experiments setup, as described in [58].In particular, we relied on the setting using Intervention and Control groups with post-test design. This design involves two groups where the intervention is implemented in one group and compared with a second group without the intervention, based on a post-test measure from both groups. A first group of 92 users (the intervention group) received messages according to all the strategy implemented in the TS4NLE module. Whereas a second group of 28 users, that was our control group, did not receive messages implementing the arguments about the consequences of a violation. Indeed, they received only canned text messages with the feedback notifying when a violation was detected. An example of canned text is "Today you have drunk too much (300 ml of maximum 200 ml) fruit juice" notified as soon as the related violation is detected. Our hypothesis was that a persuasive message exploiting the strategy implemented in the TS4NLE component allows a higher decrease in the number of violations along with the usage of the application. For the *Key to Health* project, the domain experts validated and adopted three kinds of dietary rules:

- QB-Rules (Quantity-Based rules related to single meals) that check the proper amount of a given food category to be consumed in a meal. Users were asked to insert 4 meals everyday: breakfast, lunch, snack, dinner. A pair (meal, day), e.g., breakfast at day 1, is associated with an identifier number.
- DAY-Rules (related to a single day) that check the maximum (or minimum) amount (or portion) of a given food category that can (or should) be daily consumed.
- WEEK-Rules (related to a single week) that check the maximum (or minimum) amount (or portion) of a given food category that can (or should) be weekly consumed.

Figures 10, 11, and 12 present the evolution of the average number of violations detected per user related to the QB-Rules, DAY-Rules, and WEEK-Rules sets respectively.

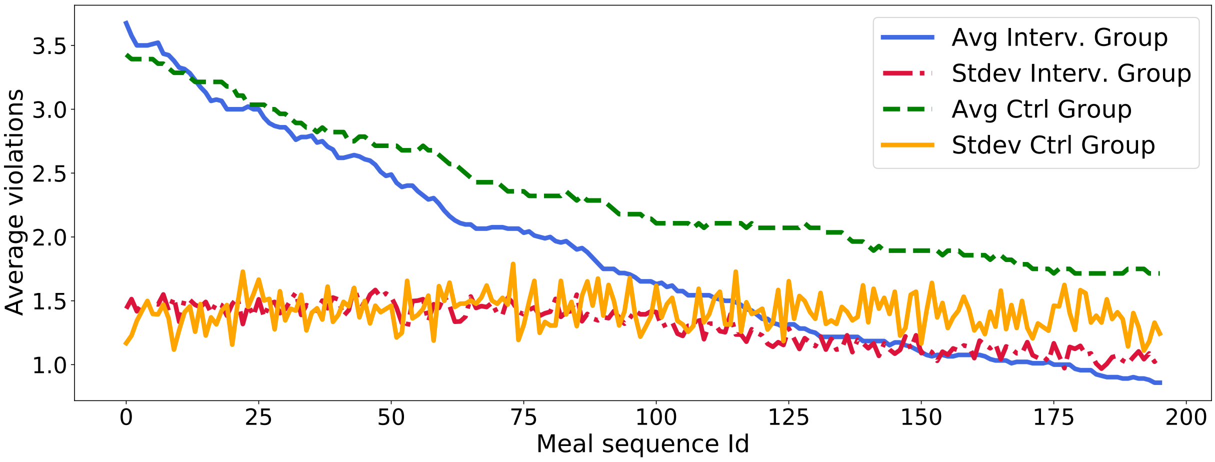

Fig. 10. Evolution of the average number of detected violations through the *Key To Health* project timespan concerning the QB-Rules. The meal sequence IDs is the sequence of the identifier numbers for each pair (meal, day).

The blue line represents the average number of violations whereas the red line the standard deviation observed for each single event in the intervention group. Then, the green line represents the average number of violations generated by the control group and the orange one the associated standard deviation. As mentioned above, QB-Rules are verified every time a user stores a meal within the solution; DAY-Rules are verified at the end of the day; while WEEK-Rules are verified at the end of each week. The increasing trend of the gap between the blue and green lines demonstrates the positive impact of the persuasive messages sent to users. We can observe how for the QB-Rules the average number of violations is below 1.0 after the first 7 weeks of the project. This means that some users started to follow all the guidelines about what to consume during a single meal. A positive result has been obtained also for the DAY-Rules and the WEEK-Rules. By considering the standard deviation lines, we can appreciate how

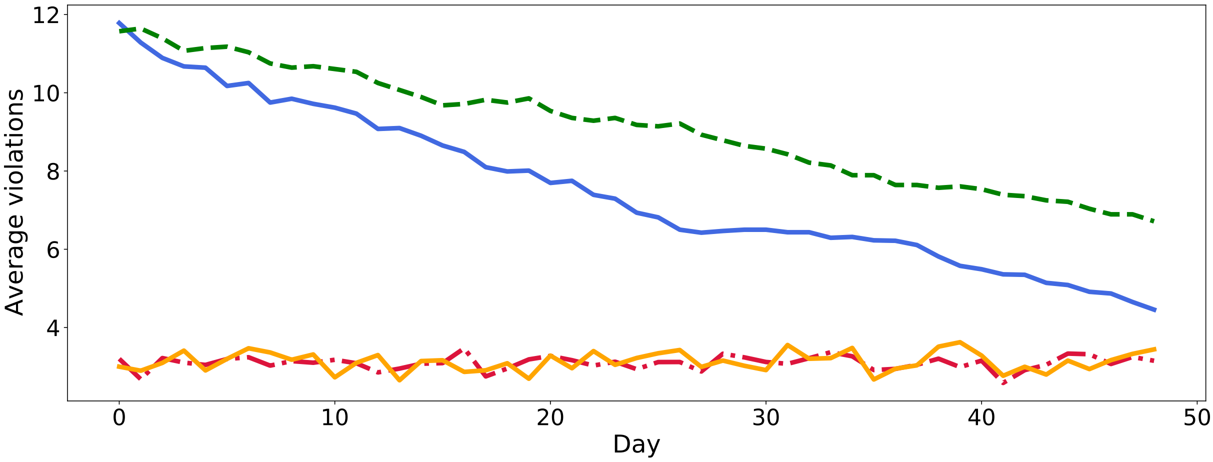

Fig. 11. Evolution of the average number of detected violations through the *Key To Health* project timespan concerning the DAY-Rules.

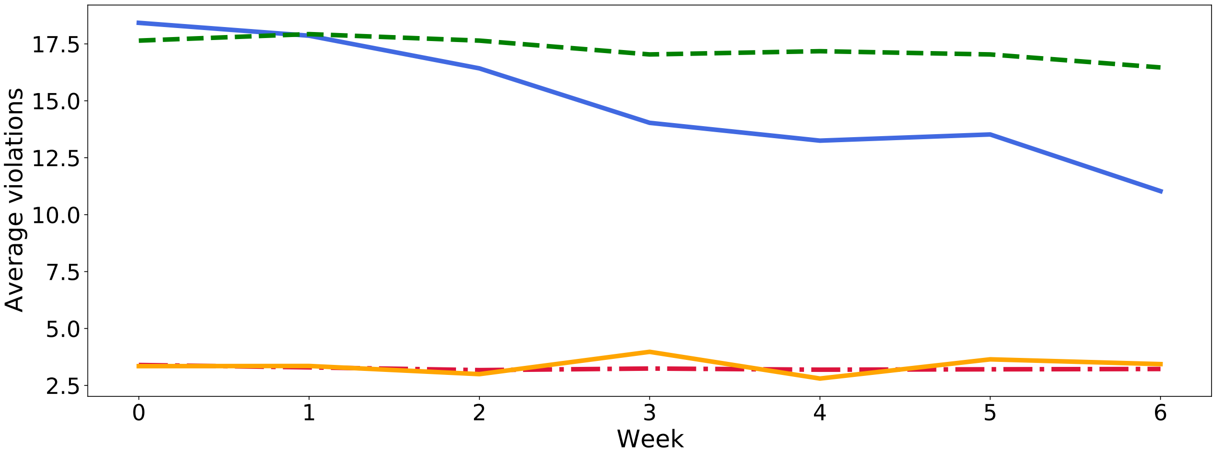

Fig. 12. Evolution of the average number of detected violations through the *Key To Health* project timespan concerning the WEEK-Rules.

both lines remain contained within low bounds and after a more in depth analysis of the data, we did not observe the presence of outliers.

The analysis of the drop of violations after the 7 weeks timespan of the project reported in Table 1 shows that that both QB and DAY rules obtained good drops. For the WEEK-Rules, however, such a drop remained limited. This can be explained with the fact that the QB and DAY-Rules are more frequently notified when violated: after every meal and day violations, respectively. Whereas the WEEK-Rules are notified once a week. As a consequence, the users pay more attention to the more frequent kind of notifications. For all the cases the intervention group has a bigger drop with respect the control group.

	Intervention Group	Control Group
QB-Rules	**76.63%**	50.00%
DAY-Rules	**62.18%**	41.98%
WEEK-Rules	**40.12%**	6.68%

Table 1

Drop of violations at the end of the project. The highest drops in violations occur with the more frequent rules.

Further quantitative analysis regard the time spent by our system to be effective. Figures 10, 11, 12 show us that the two groups tend to diverge at a certain point during the *Key To Health* time span. Here, we are interested the day/week when the two groups start to diverge with a statistical significance. We report this analysis in Table 2 with these days/week along with their p-values and average number of violations in the starting day/week for both the intervention and control group.

	Starting day/week	p-value	Violations Intervention Group	Violations Control Group
QB-Rules	29th day	0.04	1.51 ± 1.26	2.10 ± 1.57
DAY-Rules	19th day	0.007	8.09 ± 2.75	9.82 ± 2.91
WEEK-Rules	4th week	0.004	14.03 ± 3.24	17.03 ± 3.97

Table 2

Key days/week where the intervention and control groups start to diverge with statistical significance.

The DAY-Rules have the quickest starting point as the two groups diverge from the 19th day, that is, the system took less than of the 39% of the project timespan to be effective. On the other hand, the QB-Rules are the slowest to be effective taking 29 days of system usage. This is due to the fact that these rules regard strong dietary habits of users that require a constant attention and effort to be changed in order to respect the QB-rules. Indeed, for both intervention and control group the average number of violations is quite small. The WEEK-Rules have a similar starting point of the QB-Rules. This can be explained with the fact that WEEK-Rules require some organization to be respected. Indeed users need some planning of their meals for the week and consequently they have to buy the proper food with these rules in mind. This planning requires the proper effort and time.

6. Discussions and Lessons Learned

The use of *explanation graphs* is an intuitive and effective way for transforming meaningless model outputs into a comprehensive artifact that can be leveraged by targeted users. *Explanation graphs* convey formal semantics that: (i) can be enriched with other knowledge sources publicly available on the web (e.g. Linked Open Data cloud) or privacy-protected (e.g. user profiles); (ii) allow rendering in different formats (e.g. natural language text or audio); and, (iii) allow full control over the rendered explanations (i.e. the content of the explanations). Natural language rendering with a template-system allows full control on the explanations at the price of high effort in domain and user modeling by domain experts. This aspect can be considered the major bottleneck of the template-based approach described in Section 4. Such bottleneck can be mitigated by using machine learning with human-in-the-loop techniques to increase variability in the generated natural language explanations. However, in some domains (e.g., the medical one) it is necessary to keep the full control on the explanations generated by the AI-based system, thus, it is not possible to avoid the effort of building a template library.

Concerning the effort needed for improving the flexibility of the overall approach, it is important to highlight that, also on the knowledge management side, links between features provided as output by a connectionist model and ontological concepts have to be defined. Depending on the complexity of the domain (or task) in which the system is deployed, this activity may have a different impact.

In [59] as well as we reported at the beginning of the previous section, we demonstrated how working with semantic features allows the development of more comprehensible classification systems since it is possible to provide explanations about how a given instance of the dataset has been classified with a specific class. This aspect represents an interesting starting point for exploiting the generated explanations as a trigger for refining the classification model in order to increase the overall effectiveness of the system. Indeed, thanks to the links between the dataset and the semantic feature mentions contained within the explanations, we are able to detect the impact of each semantic feature on the effectiveness of the classification model. In particular when a specific semantic feature is the main actor of a wrong classification.

These aspects represent the main challenges that we aim to address in the future. In particular, we aim to abstract the conceptual model on top of the *explanation graph* to make it more general across domains. Then, we intend to enhance the template-based approach by designing a strategy for reducing the experts' effort in designing new templates. Finally, we plan to set up a use case with real users for performing the validation about the usage of explanation graphs. A candidate, and challenging, scenario is the monitoring of people affected by chronic nutritional diseases where data from both sensors and users (e.g., food images [60]) can be linked with conceptual knowledge in order to support the generation and exploitation of explanation graphs.

7. Conclusions

Explainable Artificial Intelligence aims at providing black-box algorithms with strategies that explain or justify their outputs. These algorithms need to be trusted by humans and easily understood. Knowledge bases provide a formal semantics, encoded with a logical language, that enables the connection between the concepts used by humans and the numeric features of a black box model. Indeed, an explanation in a logical language format can be automatically rendered in natural language sentences or in another formats.

In this paper, a semantic-based explainable framework based on *explanation graphs* has been presented. The framework aim is the exploitation of semantic information for making AI-based systems more human comprehensible and supports the connection between the semantic concepts of a knowledge base with the learned features in order to generate an explanation in a logical language. This allows reasoning on the black box output and its explanation, the improvement of the knowledge base and of the black box output. The annotations in the dataset are aligned with the semantics in the knowledge base.

In the future work some experiments will be performed to assess the alignment between the semantic and the learned features. This allows the evaluation of the degree of causality of the semantic features with respect to the black-box output and to study how to increase the attention of a black box towards the semantic features in order to improve the model performance.

References

[1] A. Newell, J.C. Shaw and H.A. Simon, Chess-Playing Programs and the Problem of Complexity, *IBM J. Res. Dev.* **2**(4) (1958), 320–335. doi:10.1147/rd.24.0320.

[2] W.R. Swartout, C. Paris and J.D. Moore, Explanations in Knowledge Systems: Design for Explainable Expert Systems, *IEEE Expert* **6**(3) (1991), 58–64. doi:10.1109/64.87686.

[3] M.G. Core, H.C. Lane, M. van Lent, D. Gomboc, S. Solomon and M. Rosenberg, Building Explainable Artificial Intelligence Systems, in: *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, AAAI Press, 2006, pp. 1766–1773. http://www.aaai.org/Library/AAAI/2006/aaai06-293.php.

[4] Z. Wang, Y. Lai, Z. Liu and J. Liu, Explaining the Attributes of a Deep Learning Based Intrusion Detection System for Industrial Control Networks, *Sensors* **20**(14) (2020), 3817. doi:10.3390/s20143817.

[5] A. Fernández, F. Herrera, O. Cordón, M.J. del Jesus and F. Marcelloni, Evolutionary Fuzzy Systems for Explainable Artificial Intelligence: Why, When, What for, and Where to?, *IEEE Comput. Intell. Mag.* **14**(1) (2019), 69–81. doi:10.1109/MCI.2018.2881645.

[6] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders and K. Müller, Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications, *Proc. IEEE* **109**(3) (2021), 247–278. doi:10.1109/JPROC.2021.3060483.

[7] G. Ras, M. van Gerven and P. Haselager, Explanation Methods in Deep Learning: Users, Values, Concerns and Challenges, *CoRR* **abs/1803.07517** (2018). http://arxiv.org/abs/1803.07517.

[8] A.A. Freitas, Comprehensible classification models: a position paper, *SIGKDD Explor.* **15**(1) (2013), 1–10. doi:10.1145/2594473.2594475.

[9] D. Doran, S. Schulz and T.R. Besold, What Does Explainable AI Really Mean? A New Conceptualization of Perspectives, in: *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017 co-located with 16th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2017), Bari, Italy, November 16th and 17th, 2017*, T.R. Besold and O. Kutz, eds, CEUR Workshop Proceedings, Vol. 2071, CEUR-WS.org, 2017.

[10] L.H. Gilpin, D. Bau, B.Z. Yuan, A. Bajwa, M. Specter and L. Kagal, Explaining Explanations: An Overview of Interpretability of Machine Learning, in: *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, F. Bonchi, F.J. Provost, T. Eliassi-Rad, W. Wang, C. Cattuto and R. Ghani, eds, IEEE, 2018, pp. 80–89.

[11] V. Cherkassky and S. Dhar, *Interpretation of Black-Box Predictive Models*, in: *Measures of Complexity: Festschrift for Alexey Chervonenkis*, V. Vovk, H. Papadopoulos and A. Gammerman, eds, Springer International Publishing, Cham, 2015, pp. 267–286. ISBN 978-3-319-21852-6.

[12] A. Holzinger, C. Biemann, C.S. Pattichis and D.B. Kell, What do we need to build explainable AI systems for the medical domain?, *CoRR* **abs/1712.09923** (2017).

[13] A. Holzinger, P. Kieseberg, E.R. Weippl and A.M. Tjoa, Current Advances, Trends and Challenges of Machine Learning and Knowledge Extraction: From Machine Learning to Explainable AI, in: *Machine Learning and Knowledge Extraction - Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE 2018, Hamburg, Germany, August 27-30, 2018, Proceedings*, A. Holzinger, P. Kieseberg, A.M. Tjoa and E.R. Weippl, eds, Lecture Notes in Computer Science, Vol. 11015, Springer, 2018, pp. 1–8.

[14] F. Lecue, On the role of knowledge graphs in explainable AI, *Semantic Web* **11**(1) (2020), 41–51.

[15] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti and D. Pedreschi, A Survey of Methods for Explaining Black Box Models, *ACM Comput. Surv.* **51**(5) (2019), 93:1–93:42. doi:10.1145/3236009.

[16] A. Adadi and M. Berrada, Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI), *IEEE Access* **6** (2018), 52138–52160.

[17] V. Arya, R.K.E. Bellamy, P. Chen, A. Dhurandhar, M. Hind, S.C. Hoffman, S. Houde, Q.V. Liao, R. Luss, A. Mojsilovic, S. Mourad, P. Pedemonte, R. Raghavendra, J.T. Richards, P. Sattigeri, K. Shanmugam, M. Singh, K.R. Varshney, D. Wei and Y. Zhang, One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques, *CoRR* **abs/1909.03012** (2019). http://arxiv.org/abs/1909.03012.

[18] M.T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, B. Krishnapuram, M. Shah, A.J. Smola, C.C. Aggarwal, D. Shen and R. Rastogi, eds, ACM, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.

[19] D. Bahdanau, K. Cho and Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, in: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2015. http://arxiv.org/abs/1409.0473.

[20] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun and J. Eisenstein, Explainable Prediction of Medical Codes from Clinical Text, in: *Proceedings of NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, M.A. Walker, H. Ji and A. Stent, eds, Association for Computational Linguistics, 2018, pp. 1101–1111. doi:10.18653/v1/n18-1100.

[21] Q. Xie, X. Ma, Z. Dai and E.H. Hovy, An Interpretable Knowledge Transfer Model for Knowledge Base Completion, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, R. Barzilay and M. Kan, eds, Association for Computational Linguistics, 2017, pp. 950–962. doi:10.18653/v1/P17-1088.

[22] K. Simonyan, A. Vedaldi and A. Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, in: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, eds, 2014. http://arxiv.org/abs/1312.6034.

[23] Z. Aldeneh and E.M. Provost, Using regional saliency for speech emotion recognition, in: *ICASSP*, IEEE, 2017, pp. 2741–2745.

[24] P. Pezeshkpour, Y. Tian and S. Singh, Investigating Robustness and Interpretability of Link Prediction via Adversarial Modifications, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran and T. Solorio, eds, Association for Computational Linguistics, 2019, pp. 3336–3347. doi:10.18653/v1/n19-1337.

[25] N.F. Rajani, B. McCann, C. Xiong and R. Socher, Explain Yourself! Leveraging Language Models for Commonsense Reasoning, in: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, A. Korhonen, D.R. Traum and L. Màrquez, eds, Association for Computational Linguistics, 2019, pp. 4932–4942. doi:10.18653/v1/p19-1487.

[26] A. Abujabal, R.S. Roy, M. Yahya and G. Weikum, QUINT: Interpretable Question Answering over Knowledge Bases, in: *EMNLP (System Demonstrations)*, Association for Computational Linguistics, 2017, pp. 61–66.

[27] M. Dragoni, I. Donadello and C. Eccher, Explainable AI meets persuasiveness: Translating reasoning results into behavioral change advice, *Artif. Intell. Medicine* **105** (2020), 101840. doi:10.1016/j.artmed.2020.101840.

[28] Q. Ai, V. Azizi, X. Chen and Y. Zhang, Learning Heterogeneous Knowledge Base Embeddings for Explainable Recommendation, *Algorithms* **11**(9) (2018), 137.

[29] O.Z. Khan, P. Poupart and J.P. Black, Explaining recommendations generated by MDPs, in: *Explanation-aware Computing, Papers from the 2008 ECAI Workshop, Patras, Greece, July 21-22, 2008. University of Patras*, T. Roth-Berghofer, S. Schulz, D.B. Leake and D. Bahls, eds, 2008, pp. 13–24.

[30] P. Vougiouklis, H. ElSahar, L. Kaffee, C. Gravier, F. Laforest, J.S. Hare and E. Simperl, Neural Wikipedian: Generating Textual Summaries from Knowledge Base Triples, *J. Web Semant.* **52-53** (2018), 1–15.

[31] B. Ell, A. Harth and E. Simperl, SPARQL Query Verbalization for Explaining Semantic Search Engine Queries, in: *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab and A. Tordai, eds, Lecture Notes in Computer Science, Vol. 8465, Springer, 2014, pp. 426–441.

[32] Y. Kazakov, P. Klinov and A. Stupnikov, Towards Reusable Explanation Services in Protege, in: *Description Logics*, CEUR Workshop Proceedings, Vol. 1879, CEUR-WS.org, 2017.

[33] A. Kalyanpur, B. Parsia, E. Sirin and J.A. Hendler, Debugging unsatisfiable classes in OWL ontologies, *J. Web Semant.* **3**(4) (2005), 268–293.

[34] J.S.C. Lam, Methods for resolving inconsistencies in ontologies, PhD thesis, University of Aberdeen, UK, 2007.

[35] J. Bauer, U. Sattler and B. Parsia, Explaining by Example: Model Exploration for Ontology Comprehension, in: *Description Logics*, CEUR Workshop Proceedings, Vol. 477, CEUR-WS.org, 2009.

[36] A. Kalyanpur, B. Parsia, M. Horridge and E. Sirin, Finding All Justifications of OWL DL Entailments, in: *ISWC/ASWC*, Lecture Notes in Computer Science, Vol. 4825, Springer, 2007, pp. 267–280.

[37] R.G. Hamed, H.J. Pandit, D. O'Sullivan and O. Conlan, Explaining Disclosure Decisions Over Personal Data, in: *ISWC Satellites*, CEUR Workshop Proceedings, Vol. 2456, CEUR-WS.org, 2019, pp. 41–44.

[38] D.L. McGuinness and A. Borgida, Explaining Subsumption in Description Logics, in: *IJCAI (1)*, Morgan Kaufmann, 1995, pp. 816–821.

[39] A. Borgida, E. Franconi and I. Horrocks, Explaining ALC Subsumption, in: *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000*, W. Horn, ed., IOS Press, 2000, pp. 209–213.

[40] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi and P.F. Patel-Schneider (eds), *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, 2003.

[41] E. Kontopoulos, N. Bassiliades and G. Antoniou, Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system, *Knowl.-Based Syst.* **24**(3) (2011), 406–419.

[42] J.A. Robinson and A. Voronkov (eds), *Handbook of Automated Reasoning (in 2 volumes)*, Elsevier and MIT Press, 2001.

[43] K. Kaljurand and N.E. Fuchs, Verbalizing OWL in Attempto Controlled English, in: *OWLED*, CEUR Workshop Proceedings, Vol. 258, CEUR-WS.org, 2007.

[44] K. Kaljurand, ACE View — an Ontology and Rule Editor based on Attempto Controlled English, in: *OWLED*, CEUR Workshop Proceedings, Vol. 432, CEUR-WS.org, 2008.

[45] I. Androutsopoulos, G. Lampouras and D. Galanis, Generating Natural Language Descriptions from OWL Ontologies: the NaturalOWL System, *J. Artif. Intell. Res.* **48** (2013), 671–715.

[46] D. Doran, S. Schulz and T.R. Besold, What Does Explainable AI Really Mean? A New Conceptualization of Perspectives, in: *CEx@AI*IA*, CEUR Workshop Proceedings, Vol. 2071, CEUR-WS.org, 2017, pp. 1–8.

[47] A.B. Arrieta, N.D. Rodríguez, J.D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Inf. Fusion* **58** (2020), 82–115.

[48] S.M. Lundberg and S. Lee, A Unified Approach to Interpreting Model Predictions, in: *NIPS*, 2017, pp. 4765–4774.

[49] N. Díaz-Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes and F. Herrera, EXplainable Neural-Symbolic Learning (X-NeSyL) methodology to fuse deep learning representations with expert knowledge graphs: The MonuMAI cultural heritage use case, *Information Fusion* **79** (2022), 58–83. doi:https://doi.org/10.1016/j.inffus.2021.09.022. https://www.sciencedirect.com/science/article/pii/S1566253521001986.

[50] J. Euzenat and P. Shvaiko, *Ontology Matching, Second Edition*, Springer, 2013. ISBN 978-3-642-38720-3.

[51] É. Thiéblin, O. Haemmerlé, N. Hernandez and C. Trojahn, Survey on complex ontology matching, *Semantic Web* **11**(4) (2020), 689–727. doi:10.3233/SW-190366.

[52] M. Dragoni, T. Bailoni, R. Maimone and C. Eccher, HeLiS: An Ontology for Supporting Healthy Lifestyles, in: *International Semantic Web Conference (2)*, Lecture Notes in Computer Science, Vol. 11137, Springer, 2018, pp. 53–69.

[53] I. Donadello, M. Dragoni and C. Eccher, Persuasive Explanation of Reasoning Inferences on Dietary Data, in: *PROFILES/SEMEX@ISWC*, CEUR Workshop Proceedings, Vol. 2465, CEUR-WS.org, 2019, pp. 46–61.

[54] E. Reiter and R. Dale, Building applied natural language generation systems, *Nat. Lang. Eng.* **3**(1) (1997), 57–87.

[55] C. Ke, F. Xiao, Z. Huang, Y. Meng and Y. Cao, Ontology-Based Privacy Data Chain Disclosure Discovery Method for Big Data, *IEEE Trans. Serv. Comput.* **15**(1) (2022), 59–68. doi:10.1109/TSC.2019.2921583.

[56] H. op den Akker, M. Cabrita, R. op den Akker, V.M. Jones and H.J. Hermens, Tailored motivational message generation: A model and practical framework for real-time physical activity coaching, *Journal of Biomedical Informatics* **55** (2015), 104–115.

[57] M. Guerini, O. Stock and M. Zancanaro, A Taxonomy of Strategies for Multimodal Persuasive Message Generation, *Applied Artificial Intelligence Journal* **21**(2) (2007), 99–136.

[58] A.D. Harris, J.C. McGregor, E.N. Perencevich, J.P. Furuno, J. Zhu, D.E. Peterson and J. Finkelstein, Position Paper: The Use and Interpretation of Quasi-Experimental Studies in Medical Informatics, *J. Am. Medical Informatics Assoc.* **13**(1) (2006), 16–23. doi:10.1197/jamia.M1749.

[59] I. Donadello and M. Dragoni, An End-to-End Semantic Platform for Nutritional Diseases Management, in: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I.F. Cruz, A. Hogan, J. Song, M. Lefrançois and F. Gandon, eds, Lecture Notes in Computer Science, Vol. 11779, Springer, 2019, pp. 363–381. doi:10.1007/978-3-030-30796-7_23.

[60] I. Donadello and M. Dragoni, Ontology-Driven Food Category Classification in Images, in: *Image Analysis and Processing - ICIAP 2019 - 20th International Conference, Trento, Italy, September 9-13, 2019, Proceedings, Part II*, E. Ricci, S.R. Bulò, C. Snoek, O. Lanz, S. Messelodi and N. Sebe, eds, Lecture Notes in Computer Science, Vol. 11752, Springer, 2019, pp. 607–617. doi:10.1007/978-3-030-30645-8_55.

[61] A. Gatt and E. Reiter, SimpleNLG: A Realisation Engine for Practical Applications, in: *ENLG*, The Association for Computer Linguistics, 2009, pp. 90–93.