Bridging Signals to Natural Language Explanations With Explanation Graphs

Ivan Donadello[1], Mauro Dragoni[2]

[1] *Free University of Bolzano, Bolzano, Italy*

[1] *Fondazione Bruno Kessler, Trento, Italy*

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 significantly grown in the last years. Let us consider how much pervasive became the presence 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).

Unfortunately, the usage of AI-based models, built by machine learning algorithms, introduced the issue that such models are *black box* in nature. Hence, the main challenge to tackle is to understand how the model or algorithm works and generates decisions because such decisions may affect human interests, rights, and lives. For this reason, it is crucial for high stakes applications such as credit approval in finance and digital diagnostics where AI-based systems may only support the experts during the decision making process. Beside the technical challenge, strategies for generating explanations have to take into account also regulators' laws like European Union's General Data Protection Regulation (GDPR) [1] [1], US government's "Algorithmic Accountability

XAI.it 2021 - Italian Workshop on Explainable Artificial Intelligence

ivan.donadello@unibz.it (I. Donadello); dragoni@fbk.eu (M. Dragoni)

0000-0002-0701-5729 (I. Donadello); 0000-0003-0380-6571 (M. Dragoni)

[1] https://www.eugdpr.org

Act of 2019" [2], or U.S. Department of Defense's Ethical Principles for Artificial Intelligence [3]). This aspect is fundamental for tackling the fairness, accountability, and transparency-related risks withing such automated decision making systems.

Explainable AI (XAI) is a re-emerging research trend, as the need to advocate the principles mentioned above in order to promote the requirements of transparency and trustworthiness that AI-based decision-making system must have for actually supporting experts in their activities.

In this paper, we target the challenge of designing a strategy able to support 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 type of explanation generation strategies mentioned in Section 2, we want to highlight how the usage of a knowledge-based solution, and by focusing in particular on the role of the explanation graph, may be one of the most suitable alternative for satisfying the requirements mentioned above.

The start by surveying, in Section 2, the most significant categories of explanation generation and rendering strategies. Then, we present the concept of **explanation graph**, its crucial role during the explanation generation process and we present, in Section 4, a use case where the explanation graph has been applied. In Section 5, we summarizes lessons learned and future challenges about the integration of explanation graphs within AI-based system and then we conclude the paper.

2. Related Work

Explanations are often categorized along two main aspects [2, 3]: (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 justification 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 [4] 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 made. LIME [5] 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 an XAI approach. The literature presents three main categories of approaches for rendering the generated explanations. The first category is *saliency-based representations* [4] that are primarily used to

[2] https://www.senate.gov

[3] https://www.defense.gov

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

visualize the importance scores of different types of elements in XAI learning systems, such as showing input-output word alignment [6], highlighting words in input text [7] or displaying extracted relations [8]. 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 [9] and speech [10]).

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 [11]. 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 explanation*. 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) [12] or by using simple template-based approaches [13] where, for example, knowledge-based strategies are used for selecting the proper terminology to use based on the type of communication to provide and of type of end users [14].

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

3. From Explanation Graph To Explanation Rendering

Visualization techniques mentioned in Section 2 aims to convert the output generated by *black box*, usually provided in a structured format, into a graphical representation. 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 [15]. 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.

Here, we focus on the construction of **explanation graphs** from the structured output provided by *black box* models and how to exploit it for generating Natural Language Explanations (NLE). An **explanation graph** is a conceptual representation of the structured output provided by the **black box** model where each feature of the output is represented by means of conceptual knowledge enriched with further information gathered by external sources. An **explanation graph** works as a bridge between the signals produced by the **black box** model and an understandable rendering of such signals. Producing such explanation carries a challenge, given the requirement of adopting proper language with respect to the targeted audience [16] and their context. Let us consider a sample scenario occurring within the healthcare domain where patients suffering from diabetes are monitored by a virtual coaching 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 virtual coaching system interacts with both clinicians and patients. When an undesired behavior is detected, it has to generate two different explanations: one for the clinician containing 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. In this scenario the privacy issue is limited since the clinician use to know the whole health conditions of the patients. However, 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 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. The *explanation graph* has two main features 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 [17, 18] can be regarded as the nodes of the explanation graph, whereas arcs are computed on the basis of the SHAP features values. SHAP's output is one of the possible inputs that the generator of the explanation graph can process. Indeed, such a generator is agnostic with respect to the type of model adopted by machine learning systems, since it can work with any approach providing an output that can be represented with a graph-like format. 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.

Explanations require a starting formal (graph-like) representation to be easily rendered and personalized through natural language text [19]. The generation of such natural language explanations can rely on pipelines which takes the structured explanation content as input and, through several steps, performs its linguistic realization [20]. The work in [19] injects in such a pipeline a template system that implements the text structuring phase of the pipeline. Figure 1 shows the explanation generation process starting from a SHAP analysis of a model output.

Features contained within the SHAP output are transformed into concepts linked with a knowledge base related to the problem's domain. Such a knowledge base is exploited also for extracting relationships between the detected concepts. This preliminary explanation graph can be enriched with further knowledge extracted from publicly available resources (e.g. the Linked Open Data cloud) as well as with private data (e.g. personal health records). Finally, the explanation graph, through the NLE rendering component is transformed into a natural language explanation.

As mentioned above, generating natural language explanations starts from the creation of the explanation graph, since it provides a complete structured representation of the knowledge that has to be transferred to the target user. As first step, the features of the SHAP output are transformed into concepts of the explanation graph and they are, possibly, aligned with entities contained within the knowledge base related to the problem's domain. Such entities represent the first elements composing the explanation graph that can be used as collector for further knowledge exploited for creating the complete message. Beside the alignment of SHAP output features with the domain knowledge, such a knowledge base is exploited for extracting the relationships among the identified concepts. The extraction of such relationships is fundamental for completing the explanation graph as well as for supporting its transformation into its equivalent natural language representation. Once the alignment between the SHAP output and the domain knowledge has been completed, the preliminary explanation graph can be extended in two ways. First, public available knowledge can be linked to the preliminary explanation graph for completing the domain knowledge. Second, personal information of the specific end user can be attached to the explanation for improving the context of the explanation itself and for supporting the motivation of the explanation, if any. It is important to highlight that the usage of personal information can be restricted to specific target users. Hence, a privacy check has to be performed before to include it.

Let us consider as example the explanation graph shown in Figure 2. Some medical information associated with the identified food category may not be contained in the domain knowledge integrated into the local system. Hence, by starting from the concept representing the food category, we may access, through the Linked Open Data cloud, the UMLS[5] knowledge base for extracting information about the nutritional disease risks connected with such a food category. Beside public knowledge, the explanation graph can be enriched with user information provided if and only if they are compliant with respect to possible privacy constraints. User information can be provided by knowledge bases as well as probabilistic models. Also in this case the explanation graph generator is agnostic with respect to the external source to exploit. In the use case we present below, the generator relies on an external user-oriented knowledge base containing facts that can be exploited for deciding which kind of linguistic strategy to adopt.

Finally, the created explanation graph can be rendered in a natural language form through a template system for natural language explanations (TS4NLE) [19] that leverages a Natural Language Generation (NLG) pipeline. 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. Terms in the explanation graph encode the rationale behind the AI 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. Generally, the user's information are encoded in a user model is previously given, in form of an ontology or knowledge graph.

TS4NLE is structured as a decision tree where the first level contains high-level and generic

[5] https://www.nlm.nih.gov/research/umls/index.html

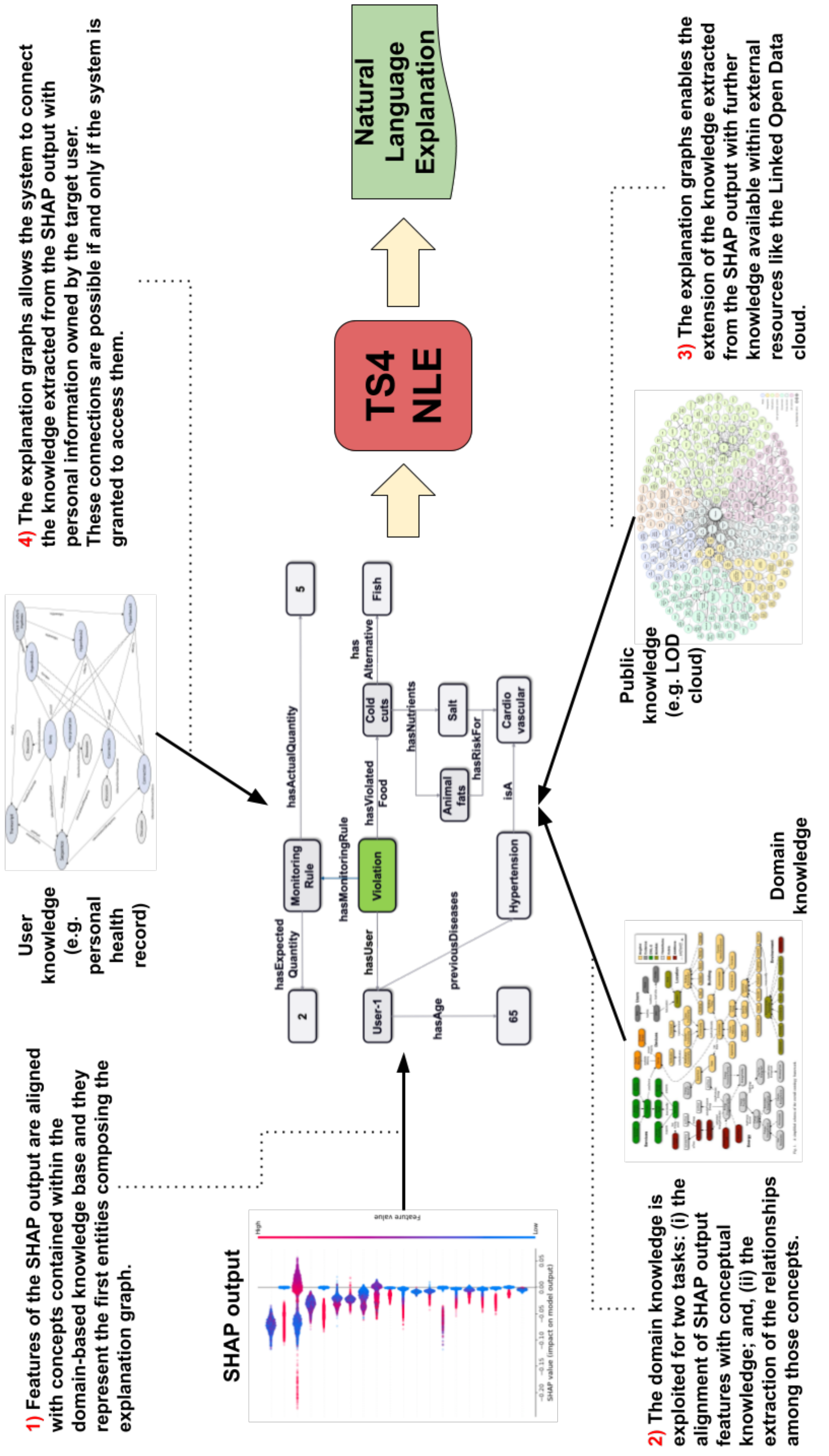
1) Features of the SHAP output are aligned with concepts contained within the domain-based knowledge base and they represent the first entities composing the explanation graph.
SHAP output
Feature value
High
Low
SHAP value (impact on model output)
User knowledge (e.g. personal health record)
4) The explanation graphs allows the system to connect the knowledge extracted from the SHAP output with personal information owned by the target user. These connections are possible if and only if the system is granted to access them.
2
hasExpectedQuantity
Monitoring Rule
hasActualQuantity
5
User-1
hasUser
hasMonitoringRule
Violation
hasViolatedFood
Cold cuts
hasAlternative
Fish
hasAge
previousDiseases
hasNutrients
Animal fats
Salt
hasRiskFor
65
Hypertension
isA
Cardio vascular
TS4 NLE
Natural Language Explanation
Domain knowledge
2) The domain knowledge is exploited for two tasks: (i) the alignment of SHAP output features with conceptual knowledge; and, (ii) the extraction of the relationships among those concepts.
Public knowledge (e.g. LOD cloud)
3) The explanation graphs enables the extension of the knowledge extracted from the SHAP output with further knowledge available within external resources like the Linked Open Data cloud.

Figure 1: The images show the process of transforming a SHAP output into an explanation graph that is then transformed into its equivalent natural language explanation.

templates that are progressively specialized and enriched according to the user's feature specified in the user model. Once templates are filled with non-terminal terms, the lexicalization[6] and linguistic realization of the pipeline are performed with standard natural language processing engines such as RosaeNLG[7].

4. Natural language explanations use case: persuasive messages for healthy lifestyle adherence

In this section, we provide the description of a complete use case related to the generation of persuasive natural language explanation within the healthcare domain.

Given as input a user lifestyle (obtained with a diet diary or a physical activity tracker), AI systems are able to classify the user behavior in classes ranging from *very good* to *very bad*. The explanation graph contains the reason for such a prediction and suggestions for reinforcing or changing the particular lifestyle. According to the user model (e.g., whether the user has to be encouraged or not, the users' barriers or capacities), the template system is explored in order to reach a leaf containing the right terms to fill the initial non-terminal symbols of the template. 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. A further tutorial of this use case is available online[8].

The explanation graph contains entities connected by relations encoding the rationale of the AI system decision. Figure 2 contains the explanation graph for a 65 years old user that consumes too much cold cuts. Such a graph is rendered with TS4NLE 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"*.

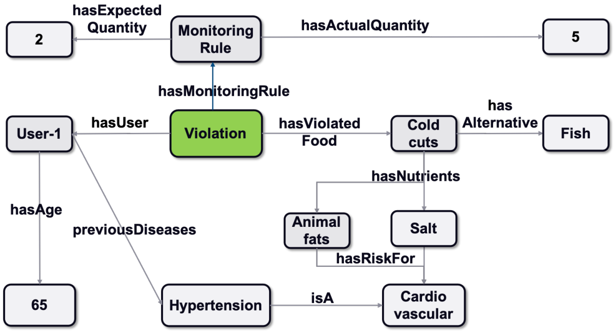

Figure 2: Explanation graph for users exceeding in cold cuts consumption in the diet & healthy lifestyle adherence application.

[6] *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.

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

[8] https://horus-ai.fbk.eu/semex4all/

The generation of the natural language explanation shown above is performed by TS4NLE by following the steps below. After the generation of the explanation graph, the *message composition* component of TS4NLE starts the generation of three textual messages for the feedback, the argument and the suggestion, respectively. This is inspired by the work in [21] and expanded taking into consideration additional strategies presented in [22]. These consist of several persuasion strategies that can be combined together to form a complex message. Each strategy is rendered through natural language text with a template. 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 reference ontology. 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[9]. Below we describe the implemented strategies to automate the message generation, focusing also on linguistic choices. The template model together with an example for instantiating it, is represented in Figure 3.

Explanation Feedback: 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.

Explanation Argument: it 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.

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 daily fruit juice drinking to less than 200 ml (a water glass) since it contains too much sugar. In the presence of an excess in juice 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.

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

[9] 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[10]) to generate well-formed sentences starting from the constraints written in the template (e.g., tenses and subject consistency for verbs)

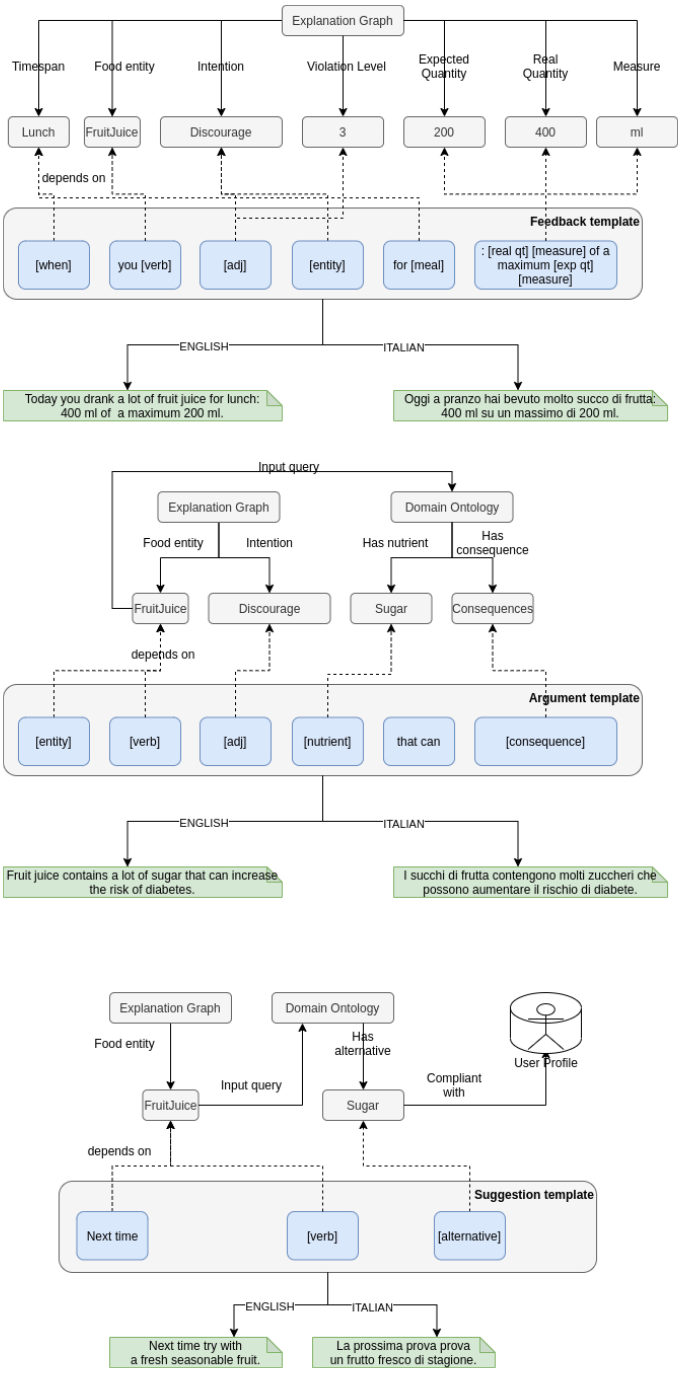

Figure 3: TS4NLE model (template and example of violation) for generating the text of an explanation. The top part represents the generation of the feedback. Choices on template and message chunks depend on the violation package. This holds also for both the argument and suggestion. Dashed lines represent a dependency relation. A template of type “informative” is used in the example. The middle part represents the generation of the argument, which is given as part of the explanation when violating diet restrictions. Finally, the bottom part represents the generation of the suggestion.

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.

Continuing with the running example, first TS4NLE queries the domain knowledge base through the reasoner 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. Both tools[11] and the colaboratory (Colab notebook) session are online[12] for freely creating new use cases using the TS4NLE approach.

5. Discussion And Conclusions

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 TS4NLE approach. Such bottleneck can be mitigated by using machine learning with human-in-the-loop techniques to increase variability in the generated natural language explanations.

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 machine learning 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.

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 for making it more general across domains. Then, we intend to enhance the template-based approach by designing a

[11] https://github.com/ivanDonadello/TS4NLE

[12] https://colab.research.google.com/drive/1iCVSt7TFMruSzeg5DswLOzOR1n7xATbz

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 [23]) can be linked with conceptual knowledge in order to support the generation and exploitation of explanation graphs.

References

[1] G. Ras, M. van Gerven, P. Haselager, Explanation methods in deep learning: Users, values, concerns and challenges, CoRR abs/1803.07517 (2018). URL: http://arxiv.org/abs/1803.07517. arXiv:1803.07517.

[2] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2019) 93:1–93:42. URL: https://doi.org/10.1145/3236009. doi:10.1145/3236009.

[3] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160.

[4] 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, Y. Zhang, One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques, CoRR abs/1909.03012 (2019). URL: http://arxiv.org/abs/1909.03012. arXiv:1909.03012.

[5] M. T. Ribeiro, S. Singh, C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, in: B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, R. Rastogi (Eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, ACM, 2016, pp. 1135–1144. URL: https://doi.org/10.1145/2939672.2939778. doi:10.1145/2939672.2939778.

[6] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1409.0473.

[7] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, J. Eisenstein, Explainable prediction of medical codes from clinical text, in: M. A. Walker, H. Ji, A. Stent (Eds.), Proceedings of NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), Association for Computational Linguistics, 2018, pp. 1101–1111. URL: https://doi.org/10.18653/v1/n18-1100. doi:10.18653/v1/n18-1100.

[8] Q. Xie, X. Ma, Z. Dai, E. H. Hovy, An interpretable knowledge transfer model for knowledge base completion, in: R. Barzilay, M. Kan (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, Association for Computational Linguistics, 2017, pp. 950–962. URL: https://doi.org/10.18653/v1/P17-1088. doi:10.18653/v1/P17-1088.

[9] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: Y. Bengio, Y. LeCun (Eds.), 2nd

International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings, 2014. URL: http://arxiv.org/abs/1312.6034.

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

[11] P. Pezeshkpour, Y. Tian, S. Singh, Investigating robustness and interpretability of link prediction via adversarial modifications, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, 2019, pp. 3336–3347. URL: https://doi.org/10.18653/v1/n19-1337. doi:10.18653/v1/n19-1337.

[12] N. F. Rajani, B. McCann, C. Xiong, R. Socher, Explain yourself! leveraging language models for commonsense reasoning, in: A. Korhonen, D. R. Traum, L. Màrquez (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, Association for Computational Linguistics, 2019, pp. 4932–4942. URL: https://doi.org/10.18653/v1/p19-1487. doi:10.18653/v1/p19-1487.

[13] A. Abujabal, R. S. Roy, M. Yahya, G. Weikum, QUINT: interpretable question answering over knowledge bases, in: EMNLP (System Demonstrations), Association for Computational Linguistics, 2017, pp. 61–66.

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

[15] D. Doran, S. Schulz, T. R. Besold, What does explainable AI really mean? A new conceptualization of perspectives, in: CEx@AI*IA, volume 2071 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2017, pp. 1–8.

[16] 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, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115.

[17] S. M. Lundberg, S. Lee, A unified approach to interpreting model predictions, in: NIPS, 2017, pp. 4765–4774.

[18] N. D. Rodríguez, A. Lamas, J. Sanchez, G. Franchi, I. Donadello, S. Tabik, D. Filliat, P. Cruz, R. Montes, F. Herrera, Explainable neural-symbolic learning (x-nesyl) methodology to fuse deep learning representations with expert knowledge graphs: the monumai cultural heritage use case, CoRR abs/2104.11914 (2021).

[19] I. Donadello, M. Dragoni, C. Eccher, Persuasive explanation of reasoning inferences on dietary data, in: PROFILES/SEMEX@ISWC, volume 2465 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 46–61.

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

[21] H. op den Akker, M. Cabrita, R. op den Akker, V. M. Jones, H. Hermens, Tailored

motivational message generation: A model and practical framework for real-time physical activity coaching, Journal of Biomedical Informatics 55 (2015) 104–115.

[22] M. Guerini, O. Stock, M. Zancanaro, A taxonomy of strategies for multimodal persuasive message generation, Applied Artificial Intelligence Journal 21 (2007) 99–136.

[23] I. Donadello, M. Dragoni, Ontology-driven food category classification in images, in: E. Ricci, S. R. Bulò, C. Snoek, O. Lanz, S. Messelodi, N. Sebe (Eds.), Image Analysis and Processing - ICIAP 2019 - 20th International Conference, Trento, Italy, September 9-13, 2019, Proceedings, Part II, volume 11752 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 607–617. URL: https://doi.org/10.1007/978-3-030-30645-8_55. doi:10.1007/978-3-030-30645-8_55.