SeXAI: Introducing Concepts into Black Boxes for Explainable Artificial Intelligence

Ivan Donadello[1] and Mauro Dragoni[1]

Fondazione Bruno Kessler, Via Sommarive 18, I-38123, Trento, Italy
{donadello|dragoni}@fbk.eu

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 present the first version of SeXAI, a semantic-based explainable framework aiming to exploit semantic information for making black boxes more transparent. After a theoretical discussion, we show how this research direction is suitable and worthy of investigation by showing its application to a real-world use case.

1 Introduction

[1] Explainable Artificial Intelligence (XAI) aims at explaining the algorithmic decisions of AI solutions with non-technical terms in order to make these decisions trusted and easily comprehensible by humans [1]. If these AI solutions are based on learning algorithms and perceived as black boxes due to their complexity, XAI makes them more transparent and interpretable too. This is of great interest for both logical reasoning in rule engines and Machine Learning (ML) methods. The explanation of a reasoning process can be very difficult, especially when a system is based on a set of complex logical axioms whose logical inferences are performed with, for example, tableau algorithms [4]. Indeed, inconsistencies in logical axioms may be not well understood by users if the system limits to just report the violated axioms. Indeed, users are generally skilled to understand neither formal languages nor the behavior of a whole system. This is crucial for some applications, such as a power plant system where a warning message to the user must be clear and concise to avoid catastrophic consequences. On the other hand, ML methods are based on statistical models of the data where some explanatory variables (i.e., the features) of the data are leveraged in order to predict a dependent variable (i.e., a class or a numeric value). Many statistical methods (e.g., the principal component analysis) are able to detect what are the main involved features in a ML task. These involved features can be used to *explain* to user the reason of a particular decision. These features are usually handcrafted by human experts and consequently

present a shared semantics. Modern Deep Neural Network (DNN) are able to learn these features with no need of human effort. However, the semantics of these learnt features is nor explicit or shareable with humans. Therefore, a human-comprehensible explanation about how and why an AI system took a decision is necessary.

A shared and agreed definition on explainability has not been reached in the AI community so far. Here we follow the definition of Adadi and Berrada [1] that argue for a distinction between interpretability and explainability. The former regards the study of a mathematical mapping between the inputs and the outputs of a black-box system. The latter regards a human comprehension of the logic and/or semantics of such a system. Doran et al. [13] refine the notion of explainability stating that an explainable (or comprehensible) system should provide a reason or justification about its output instead of focusing solely on the mathematical mapping. Moreover, they argue that *truly explainable systems* have to adopt reasoning engines that run on knowledge bases containing an explicit semantics in order to generate a human comprehensible explanation. In addition, the explainability power depends also on the background knowledge of the users.

To this extent, the logical reasoning associated to semantics is fundamental as it represents a bridge between the output machine and human concepts. This differs from other XAI works that try to analyze the activations of the hidden neurons (i.e., the learnt features) with respect to a given output without attaching a shared semantics. However, logical reasoning on the back-box output is not sufficient as it performs a post-hoc explanation of the black-box guided only by the axioms of a knowledge base. Indeed, no explicit link from the black-box learned features and the concepts in the knowledge base is used. The contribution of the paper addresses this issue.

We propose a novel semantic-based XAI (SeXAI) framework that generates explanations for a black-box output. Differently from Doran et al., such explanations are First-Order logic formulas whose predicates are semantic features connected to the classes of the black box output. Logic formulas are then easy to translate in natural language for a better human comprehension. Moreover, the semantic features are aligned with the neurons of the black box thus creating a neural-symbolic model. This allows reasoning between the output and the features and the improvement of both the knowledge base and the black box output. In addition, the semantics in the knowledge base is aligned with the annotation in the dataset. This is fundamental both for the neural-symbolic alignment and for the black box performance. The latter were tested with experiments on image classification showing that a semantic aligned with the training set outperforms a model whose semantics is deduced from the output with only logical reasoning. The rest of the paper follows with Section 2 that provides a state-of-the-art of techniques for generating explanations from logical formulas. Section 3 describes the main concepts of the SeXAI framework whereas Section 4 shows a first application and results of the framework in an image classification task. Section 5 concludes the paper.

2 Related Work

The research on XAI has been widely explored in the last years [17], but most of the contributions focused only on the analysis of how learning models (a.k.a. black boxes) work. This is a limited view of the topic since there is a school of thought arguing that an effective explainability of learning models cannot be achieved without the use of domain knowledge since data analysis alone is not enough for achieving a full-fledged explainable system [8]. This statement has been further discussed recently by asserting that the key for designing a completely explainable AI system is the integration of Semantic Web technologies [19,20,29]. Semantic Web technologies enabling the the design of strategies for providing explanations in natural language [2,26] where explanations are provided through textual rule-like notation. NLG strategies have been designed also for generating natural language text from triples [34] and for translating SPARQL queries into a natural language form understandable by non-experts [15]. Here, we focused on the integration of semantic information as enabler for improving the comprehensiveness of XAI systems. Our aim is to generate natural language explanations as result of the synergies between neural models and logic inferences for supporting end-users in understanding the output provided by the systems.

The explanation of the logical reasoning in an ontology is implemented with two two orthogonal approaches: *justifications* and *proofs*. The former computes the minimal subset of the ontology axioms that logically entails an axiom. The latter computes also all the inference steps [25].

One of the first user studies dealing with explanations for entailments of OWL ontologies was performed by [24]. The study investigated the effectiveness of different types of explanation for explaining unsatisfiable classes in OWL ontologies. The authors found that the subjects receiving full debugging support performed best (i.e., fastest) on the task, and that users approved of the debugging facilities. Similarly, [28] performed a user study to evaluate an explanation tool, but did not carry out any detailed analysis of the difficulty users had with understanding these explanations. While, [5] presents a user study evaluating 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 [23] presents several algorithms for computing all the justifications of an entailment in a OWL-DL knowledge base. However, nor study or user evaluation is performed to assess the capability of the computed justifications of the logical entailments. The work in [18] focuses on the explanation, through justifications, of the disclosure of personal data to users (patients and staff) of hospitals. This is performed by translating SWRL rules inconsistencies into natural language utterances. Moreover, the SWRL rules translation is performed axiom by axiom, thus generating a quite long sentence. This could require too much time for reading and understanding. Whereas, our method returns only a single utterance summarizing the whole justification.

Formal proofs are the other form of explanation for logical reasoning. In [31] the authors present an approach to provide proof-based explanations for entailments of the CLASSIC system. The system omits intermediate steps and provides further filtering

strategies in order to generate short and simple explanations. The work proposed in [7] first introduced a proof-based explanation system for knowledge bases in the Description Logic ALC [4]. The system generates sequent calculus style proofs using an extension of a tableaux reasoning algorithm, which are then enriched to create natural language explanations. However, there exists no user studies to explore the effectiveness of these proofs. In [27] the authors proposed several (tree, graphical, logical and hybrid) visualizations of defeasible logic proofs and present a user study in order to evaluate the impact of the different approaches. These representations are hard to understand for non-expert users. Indeed, the study is based on participants from a postgraduate course (who have attended a Semantic Web course) and from the research staff. In general, proof algorithms for Description Logic are based on Tableau techniques [4] whereas proof algorithms for other logics are studied in the field of Automated Reasoning [32].

This wide range of approaches to explanation of logical entailments is more focused on the development of efficient algorithms than on effective algorithms for common users. Indeed, all the computed explanations are sets of logical axioms understandable only by expert users. The aim of our work is to provide and effective representation to explanation for all users. This representation is based on the verbalization of the explanation in natural language. This verbalization can be performed by using methods that translate axioms of an OWL ontology in Attempto Controlled English [22,21] or in standard English [3] with the use of templates. This last work also presents some users' studies on the quality of the generated sentences. However, these works do not handle with the reasoning results (justifications or proofs), indeed, no strategy for selecting and rendering an explanation is studied.

3 The Framework

In the fields of Machine Learning and Pattern Recognition, a feature is a characteristic or a measurable property of an object/phenomenon under observation [6]. Features can be numeric or structured and they are crucial in tasks such as pattern detection, classification or regression as they serve as explanatory variables. Indeed, informative and discriminating features are combined in a simple or complex manner by the main ML algorithms. This also holds in our everyday experience, a dish composed by pasta, bacon, eggs, pepper and aged cheese (features) is recognized as pasta with Carbonara sauce (the class). Diseases are recognized according to the symptoms (features), the price of the houses is computed according to the features of, e.g., location, square meters and years of the real estate. However, with the rise of Deep Neural Networks (DNN), features are learnt by the system from the raw data without the necessity of handcrafting from domain experts. This has improved the performance of such systems with the drawback of loosing comprehensibility from users. Indeed, DNNs embed the data in a vector space in the most discriminating way without any link to a formal semantics. The aim of SeXAI is to link a DNN with a formal semantics in order to provide a comprehensible explanation of the DNN output to everyday users.

Following the definitions of Doran et al.[13], we ground the notion of explainable system into the concept of a *comprehensible system*, 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 therefire ensuring a sort of atomic property of these features. Semantic features are different from the learnt numeric (and not comprehensible) features of a DNN. The aim of a comprehensible system is to find an alignment between the learnt and the semantic features.

The connection between a DNN 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 DNN and $\{A\}_1^m$ the corresponding semantic features (or attributes).

A comprehension axiom formalizes the main tasks of a DNN:

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 DNN. 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 DNN (e.g., dinner and party) for x and $k > 1$ as a sigmoid is applied in the last layer of the DNN. 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 DNN (e.g., the asked price and the real values of house) for x. Here $k \geq 1$ with a sigmoid applied in the last layer of the DNN. The semantic features are properties of interest for buying a house.

We present the SeXAI framework for comprehensible systems in Figure 3. The knowledge base $\mathcal{KB}$ contains both the predicate symbols in $\mathcal{P}$ for annotating the data and the comprehension axioms. These latter are passed to the symbolic system that is in charge of i) analyzing the output of the DNN and the associated semantic features; ii) reasoning about them according to the comprehension axioms; iii) returning a, possibly

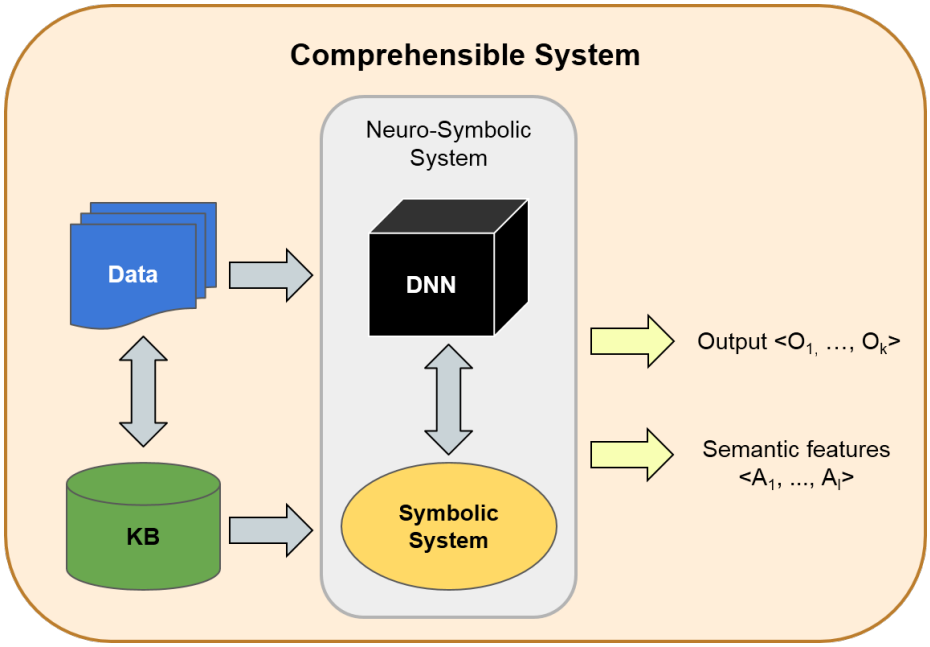

Fig. 1. In the SeXAI framework data are annotated with symbols of a knowledge base. A symbolic system is aligned with a DNN in order to provide an output and a set of semantic features consistent with the comprehension axioms in the knowledge base.

refined, output along with the related semantic features. This architecture extends the ones in [13], where a reasoner computes the explanation of the output, with a semantic module that enables several tasks that improve the comprehension and the transparency (i.e., the interpretation) of the DNN:

Output and semantic features refinement: The DNN is trained to return both the output and the semantic features in a multitasking learning setting. Then, with the use of fuzzy reasoning or neural-symbolic systems [11,12,9,30], both outputs can be refined according to the comprehension axioms and to the evidence coming from the scores of the DNN.

Feature Alignment: Once a DNN is trained in a multitasking learning setting, it is possible to analyze which are the most activated neurons of the last hidden layer [16] for each semantic feature. In this manner, we can align the high-level features of the DNN with the semantic features in $\mathcal{KB}$.

Knowledge base improvement: Once the features alignment is performed, the system can turn off the neurons corresponding to a given semantic feature and check the performance degradation with respect to the output. No degradation of the performance means that the particular semantic feature has just a correlation with the output and, therefore, it can be removed from the corresponding comprehension axiom or stated as a simple correlation. On the other hand, a degradation of the performance indicates a causality of the semantic features with respect to the output.

The more the performance degrades the higher the causality degree for that feature is. Therefore, we can enrich $\mathcal{KB}$ with some priors in the comprehension axioms about the importance of the semantic features.

Model improvement: Analyzing the semantic features returned by wrong output predictions allows the system to detect the presence of some common semantic features that alter some predictions. Therefore, the model can assign a lower weight to the neurons aligned with that semantic features.

The symbolic system in SeXAI extends the framework of Doran et al. [13] by computing the alignment of semantic and DNN features that enables the improvement of both $\mathcal{KB}$ and of the model. Differently, in [13] the reasoner module is able to only generate the output and the semantic features.

4 SeXAI in Action

Section 3 provided the general description of the SeXAI framework that we proposed for increasing the overall comprehensiveness of AI models. In this Section, we show how the SeXAI framework can be instantiated within a real-world scenario. In particular, we applied the SeXAI framework to image classification with the aim of demonstrating how the integration of semantics into an AI-based classification systems triggers both the generation of explanations and, at the same time, an improvement of the overall effectiveness of the classification model.

As described in Section 3, the SeXAI framework is composed by different modules that, depending on the scenario in which the framework is deployed, can be instantiated or not. Let us consider a scenario where the goal is to classify food images with respect to the food categories contained by the represented recipe instead of the recipe itself. Information about food categories are particularly useful in scenario where physicians are supported by information systems concerning the diet monitoring of people affected by nutritional diseases (e.g., diabetes, hypertension, obesity, etc.). By starting from the SeXAI architecture shown in Figure 3, we instantiated the modules as follows.

- The "Data" module contains our dataset of recipe images we used for training the classification model. A more detailed description of the dataset is provided in Section 4.1.
- The "Knowledge Base" contains, beyond a taxonomy of recipes and food categories, the composition of each recipe in terms of its food categories. Recipes compositions are described by object properties within the knowledge base. More specifically, in our scenario we adopted the HeLiS ontology [14] where we have the food category-based composition of more than 8,000 recipes[2].
- As "Black-box model", we implemented a DNN trained with recipe/food images annotated with the list of related food categories. Given a recipe image x, the recipe label represents the $O(x)$ output neuron, while the food categories represent the

[2] In the remaining of the paper, we will refer to some concepts defined within the HeLiS ontology. We leave to the reader the task of checking the meaning of each concept within the reference paper.

semantic features $A(x)$ output neurons. In our scenario we decided to not include the $O(x)$ output neurons and to classify each image by its semantic features $A(x)$. Hence, each neuron of the DNN output layer indicates if one of the food categories contained in the dataset has been detected within the images or not.

– Finally, in our scenario the "Symbolic System" links together the "Knowledge Base" and the output of the DNN for generating natural language explanations of the classification results.

The evaluation of explanations quality is still an open topic within the AI research area [19]. Moreover, in our scenario, explanations aim to provide a comprehensive description of the output rather than being a vehicle for improving the model. Hence, the evaluation of their language content is not of interest. Instead, the SeXAI framework evaluation provided in this work focuses on the effectiveness of exploiting semantic features for both training and classification purposes. As baseline, we used a post-hoc semantic-based strategy where images used for training the DNN were annotated only with the corresponding recipe label. Here, the list of food categories has been extracted after the classification of each images by exploiting the predicted recipe label. Figure 4 shows the building blocks of the baseline. For readability, hereafter we will refer to the instantiation of the SeXAI framework as "multi-label classifier", while the baseline will be labeled as "single-label classifier".

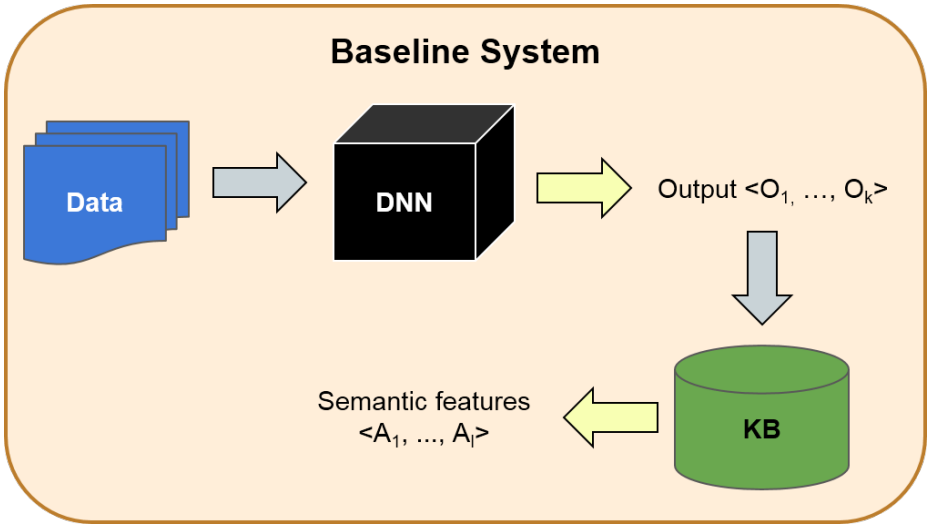

Fig. 2. The architecture of the baseline system we used for comparing the effectiveness of the SeXAI framework concerning the recipe images classification task.

4.1 Quantitative Evaluation

In the considered scenario, a good performance on recognizing food categories is important as the misclassification of images could trigger wrong behaviors of the systems in which the classifier is integrated. For example, if the framework would be integrated into a recommendation system, a misclassification of a recipe image would lead to the generation of wrong messages or even no message to the target user.

The Food and Food Categories (FFoCat) Dataset [3] We leverage the food and food category concepts in HeLiS for the multi-label classification. However, current food image datasets are not built with these concepts as labels, so it was necessary to build a new dataset (named FFoCat) with these concepts. We start by sampling some of the most common recipes in *Recipe* and use them as food labels. The food categories are then automatically retrieved from *BasicFood* with a SPARQL query. Examples of food labels are *Pasta with Carbonara Sauce* and *Baked Sea Bream*. Their associated food categories are *Pasta, AgedCheese, VegetalOils, Eggs, ColdCuts* and *FreshFish, VegetalOils*, respectively. We collect 156 labels for foods (*Recipe* concept) and 51 for food categories (*BasicFood* concept). We scrape the Web, using Google Images as search engine, to automatically download all the images related to the food labels. Then, we manually clean the dataset by checking if the images are compliant with the related labels. This results in 58,962 images with 47,108 images for the training set and 11,854 images for the test set (80-20 ratio of splitting). Then, by leveraging HeLiS properties, we enrich the image annotations with the corresponding food category labels to perform multi-label classification. The dataset is affected by some natural imbalance, indeed the food categories present a long-tail distribution: only few food categories labels have the majority of the examples. On the contrary, many food categories labels have few examples. This makes the food classification challenging.

Experimental Settings and Metrics For both multi and single-label classification we separately train the Inception-V3 network [33] from scratch on the FFoCat training set to find the best set of weights. The fine tuning using pre-trained ImageNet [10] weights did not perform sufficiently. This is probably due to the fact that the learnt low-level features of the first layers of the network belong to a general domain and do not match properly with the specific food domain. For the multi-label classification, we use a sigmoid as activation function of the last fully-connected layer of the Inception-V3 and binary cross entropy as loss function. This is a standard setting for multi-label classification. Regarding the single-label classification, the activation function of the last fully-connected layer is a softmax and the loss function is a categorical cross entropy. We run 100 epochs of training with a batch size of 16 and a learning rate of 10^{-6}. At each epoch images are resized to 299x299 pixels and are augmented by using rotations, width and height shifts, shearing, zooming and horizontal flipping. This results in a training set 100 times bigger than the initial one. We used early stopping to prevent overfitting. The training has been performed with the Keras framework (TensorFlow as backend) on a PC equipped with a NVIDIA GeForce GTX 1080.

[3] The dataset, its comparison and the code are available at `https://bit.ly/2Y7zSWZ`.

As performance metric we use the mean average precision (MAP) that summarizes the classifier precision-recall curve: $MAP = \sum_{i=1}^{n}(R_n - R_{n-1})P_n$, i.e., the weighted mean of precision P_n achieved at each threshold level n. The weight is the increase of the recall in the previous threshold: $R_n - R_{n-1}$. The macro AP is the average of the AP over the classes, the micro instead considers each entry of the predictions as a label. We preferred MAP instead of accuracy as the latter for sparse vectors can give misleading results: high results for output vectors with all zeros.

Results Given an (set of) input image(s) $\boldsymbol{x}$, the computing of the precision-recall curve requires the predicted vector(s) $\boldsymbol{y}$ of food category labels and a score associated to each label in $\boldsymbol{y}$. In the multi-label method this score is directly returned by the Inception-V3 network (the final logits). In the single-label and inference method this score needs to be computed. We test two strategies: (i) we perform *exact inference* of the food categories from HeLiS and assign the value 1 to the scores of each $y_i \in \boldsymbol{y}$; (ii) the food categories labels inherit the *uncertainty* returned the DNN: the score of each y_i is the logit value s_i returned by $DNN(\boldsymbol{x})$. Results are in Table 1. The direct multi-

Method	Micro-AP (%)	Macro-AP (%)
Multi-label (SeXAI framework)	**76.24**	**50.12**
Single-class without uncertainty (baseline)	50.53	31.79
Single-class with uncertainty (baseline)	60.21	42.51

Table 1. The multi-label classification of food categories outperforms in average precision (AP) the methods based on single-label classification and logical inference.

label has very good performance (both in micro and macro AP) in comparison with the single-label models. The micro-AP is always better than the macro-AP as it is sensible to the mentioned imbalance of the data. This means that errors in the single recipe classification propagate to the majority of the food categories the recipe contains. That is, the inferred food categories will be wrong because the recipe classification is wrong. On the other hand, errors in the direct multi-label classification will affect only few food categories. We inspected in more detail some of the errors committed by the classifiers in order to have a better understanding of their behaviors. In some cases, the single-label method misclassified an image with *Backed Potatoes* as *Backed Pumpkin* thus missing the category of *FreshStarchyVegetables*. Another image contains a *Vegetable Pie* but the single-label method infers the wrong category of *PizzaBread*. In another image, this method mistakes *Pasta with Garlic, Oil and Chili Peppers* with *Pasta with Carbonara Sauce*, thus inferring wrong *Eggs* and *ColdCuts*. Here the multi-label method classifies all the categories correctly. Therefore, the multi-label method allows a more fine grained classification of the food categories w.r.t. the single-label method. The latter has better results if the score returned by the DNN is propagated to the food categories labels w.r.t. the exact inference.

4.2 Discussion

The experience of designing the SeXAI framework and the analysis of results obtained from a preliminary validation within a real-world use case highlighted two important directions towards the long-term goal of achieving a fully-explainable AI system.

First, the integration of semantic features with black-box models enabled the generation of comprehensive explanations. SeXAI can be considered a neuro-symbolic framework conjugating the effectiveness of black-box models (e.g., DNN) with the transparency of semantic knowledge that, where possible, can support the generation of explanations describing the behavior of AI systems. This aspect opens to a very interesting and innovative research direction centered on the content of the generated explanations. Indeed, the integration of semantic features for generating explanations can be exploited for refining the statistical model itself (as described in Section 3). For instance by analyzing correlations between the presence of specific semantic features within explanations and the performance of the black-box model. Future work will focus on strengthening this liaison within the SeXAI framework in order to validate if an inference process could improve the classification capability and, at the same time, to observe how inference results could be exploited for refining the black-box model.

Second, the integration of semantic features can lead to better classification performance. Results presented in Table 1 show that through the integration of semantic features, it is possible to improve the overall effectiveness of the black-box model. This is a very interesting finding since it confirms the importance of a by-design integration of semantic features. Future activities will further investigate this hypothesis within other scenarios with the aim of understanding which are the boundaries and if there exist some constraints in the application of this strategy. For instance, the granularity of semantic features with respect to the entities that have to be classified could play an important role. Hence, a trade-off has to be found in order to maintain the explainable capability of the system and, at the same time, an acceptable effectiveness of the classification model.

5 Conclusions

The aim of Explainable Artificial Intelligence is to provide black-box algorithms with strategies to produce a reason or justification for their outputs. This is fundamental to make these algorithms trusted and easily comprehensible by humans. A formal semantics, provided by knowledge bases, encoded in a logical language allows the connection between the numeric features of a black box and the human concepts. Indeed, a justification in a logical language format can be easily translated in natural language sentences in an automatic way.

In this paper, we presented the first version of SeXAI, a semantic-based explainable framework aiming at exploiting semantic information for making black boxes more comprehensible. SeXAI is a neural-symbolic system that analyses the output of a black box and creates a connection between the learnt features and the semantic concepts of a knowledge base in order to generate an explanation in a logical language. This allows reasoning on the black box and its explanation, the improvement of the knowledge base

and of the black box output. The semantics in the knowledge base is aligned with the annotations in the dataset. This improves the performance of SeXAI on a task of multi-label image classification with respect to a system that performs solely logical reasoning on the black box output.

As future work, we will perform some experiments on the quality of the alignment between the learnt and the semantic features. In particular, we will evaluate the degree of causality of the semantic features with respect to the output and how the attention of a black box can be moved towards the semantic features in order to improve the model performance.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms **11**(9), 137 (2018)
3. Androutsopoulos, I., Lampouras, G., Galanis, D.: Generating natural language descriptions from OWL ontologies: the naturalowl system. J. Artif. Intell. Res. **48**, 671–715 (2013)
4. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
5. Bauer, J., Sattler, U., Parsia, B.: Explaining by example: Model exploration for ontology comprehension. In: Description Logics. CEUR Workshop Proceedings, vol. 477. CEUR-WS.org (2009)
6. Bishop, C.M.: Pattern recognition and machine learning, 5th Edition. Information science and statistics, Springer (2007)
7. Borgida, A., Franconi, E., Horrocks, I.: Explaining ALC subsumption. In: Horn, W. (ed.) ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000. pp. 209–213. IOS Press (2000)
8. Cherkassky, V., Dhar, S.: Interpretation of Black-Box Predictive Models, pp. 267–286. Springer International Publishing, Cham (2015)
9. Daniele, A., Serafini, L.: Neural networks enhancement through prior logical knowledge. CoRR **abs/2009.06087** (2020)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
11. Diligenti, M., Gori, M., Saccà, C.: Semantic-based regularization for learning and inference. Artif. Intell. **244**, 143–165 (2017)
12. Donadello, I., Serafini, L.: Compensating supervision incompleteness with prior knowledge in semantic image interpretation. In: IJCNN. pp. 1–8. IEEE (2019)
13. Doran, D., Schulz, S., Besold, T.R.: What does explainable AI really mean? A new conceptualization of perspectives. In: Besold, T.R., Kutz, O. (eds.) 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. CEUR Workshop Proceedings, vol. 2071. CEUR-WS.org (2017)
14. Dragoni, M., Bailoni, T., Maimone, R., Eccher, C.: Helis: An ontology for supporting healthy lifestyles. In: International Semantic Web Conference (2). Lecture Notes in Computer Science, vol. 11137, pp. 53–69. Springer (2018)

15. Ell, B., Harth, A., Simperl, E.: SPARQL query verbalization for explaining semantic search engine queries. In: Presutti, V., d'Amato, C., Gandon, F., d'Aquin, M., Staab, S., Tordai, A. (eds.) The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8465, pp. 426–441. Springer (2014)
16. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341**(3), 1 (2009)
17. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining explanations: An overview of interpretability of machine learning. In: Bonchi, F., Provost, F.J., Eliassi-Rad, T., Wang, W., Cattuto, C., Ghani, R. (eds.) 5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018. pp. 80–89. IEEE (2018)
18. Hamed, R.G., Pandit, H.J., O'Sullivan, D., Conlan, O.: Explaining disclosure decisions over personal data. In: ISWC Satellites. CEUR Workshop Proceedings, vol. 2456, pp. 41–44. CEUR-WS.org (2019)
19. Holzinger, A., Biemann, C., Pattichis, C.S., Kell, D.B.: What do we need to build explainable AI systems for the medical domain? CoRR **abs/1712.09923** (2017)
20. Holzinger, A., Kieseberg, P., Weippl, E.R., Tjoa, A.M.: Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E.R. (eds.) 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. Lecture Notes in Computer Science, vol. 11015, pp. 1–8. Springer (2018)
21. Kaljurand, K.: ACE view — an ontology and rule editor based on attempto controlled english. In: OWLED. CEUR Workshop Proceedings, vol. 432. CEUR-WS.org (2008)
22. Kaljurand, K., Fuchs, N.E.: Verbalizing OWL in attempto controlled english. In: OWLED. CEUR Workshop Proceedings, vol. 258. CEUR-WS.org (2007)
23. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: ISWC/ASWC. Lecture Notes in Computer Science, vol. 4825, pp. 267–280. Springer (2007)
24. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in OWL ontologies. J. Web Semant. **3**(4), 268–293 (2005)
25. Kazakov, Y., Klinov, P., Stupnikov, A.: Towards reusable explanation services in protege. In: Description Logics. CEUR Workshop Proceedings, vol. 1879. CEUR-WS.org (2017)
26. Khan, O.Z., Poupart, P., Black, J.P.: Explaining recommendations generated by mdps. In: Roth-Berghofer, T., Schulz, S., Leake, D.B., Bahls, D. (eds.) Explanation-aware Computing, Papers from the 2008 ECAI Workshop, Patras, Greece, July 21-22, 2008. University of Patras. pp. 13–24 (2008)
27. Kontopoulos, E., Bassiliades, N., Antoniou, G.: Visualizing semantic web proofs of defeasible logic in the DR-DEVICE system. Knowl.-Based Syst. **24**(3), 406–419 (2011)
28. Lam, J.S.C.: Methods for resolving inconsistencies in ontologies. Ph.D. thesis, University of Aberdeen, UK (2007)
29. Lécué, F.: On the role of knowledge graphs in explainable AI. Semantic Web **11**(1), 41–51 (2020)
30. Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In: ICLR. OpenReview.net (2019)
31. McGuinness, D.L., Borgida, A.: Explaining subsumption in description logics. In: IJCAI (1). pp. 816–821. Morgan Kaufmann (1995)
32. Robinson, J.A., Voronkov, A. (eds.): Handbook of Automated Reasoning (in 2 volumes). Elsevier and MIT Press (2001)

33. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826. IEEE Computer Society (2016)
34. Vougiouklis, P., ElSahar, H., Kaffee, L., Gravier, C., Laforest, F., Hare, J.S., Simperl, E.: Neural wikipedian: Generating textual summaries from knowledge base triples. J. Web Semant. **52-53**, 1–15 (2018)