Generating and Modifying Natural Language Explanations

Abdus Salam, Rolf Schwitter and Mehmet A. Orgun

{abdus.salam, rolf.schwitter, mehmet.orgun}@mq.edu.au

Department of Computing, Macquarie University

Motivation

- Machine learning (ML) models are remarkably successful, especially sub-symbolic models in solving prediction tasks.
- Sub-symbolic ML models are mostly black-box models that are not immediately interpretable. Therefore, it is difficult to explain to a user why a machine learning model makes a particular prediction.
- Most explanation systems use existing information of the datasets to explain predictions.
- Sometime information that is not present directly in the dataset such as relation information can play an important role for an explanation; especially, in image prediction tasks.
- HESIP is a hybrid explanation system for image predictions that combines sub-symbolic and symbolic ML techniques to explain the predictions of image classification tasks.
- For an input image, HESIP makes a prediction using a sub-symbolic ML model and after that, uses a symbolic ML technique to learn symbolic probabilistic rules that are used to explain the prediction.
- The explanations are generated in a controlled natural language (CNL) using a logic programming based bi-directional grammar.
- The HESIP system aims to generate an explanation for the predicted image that represents the object information together with the relation information.
- It is important that a user can modify an incorrect explanation so that the system can learn a better explanation taking the feedback from the user into consideration.
- In this paper, we present a method that involves a human-in-the-loop who can fix incorrect explanations by modifying them.
- To the best of our knowledge, there is no explanation system that allows a user to modify an explanation in order to improve the explanation generation process of the system.

HESIP: System Architecture

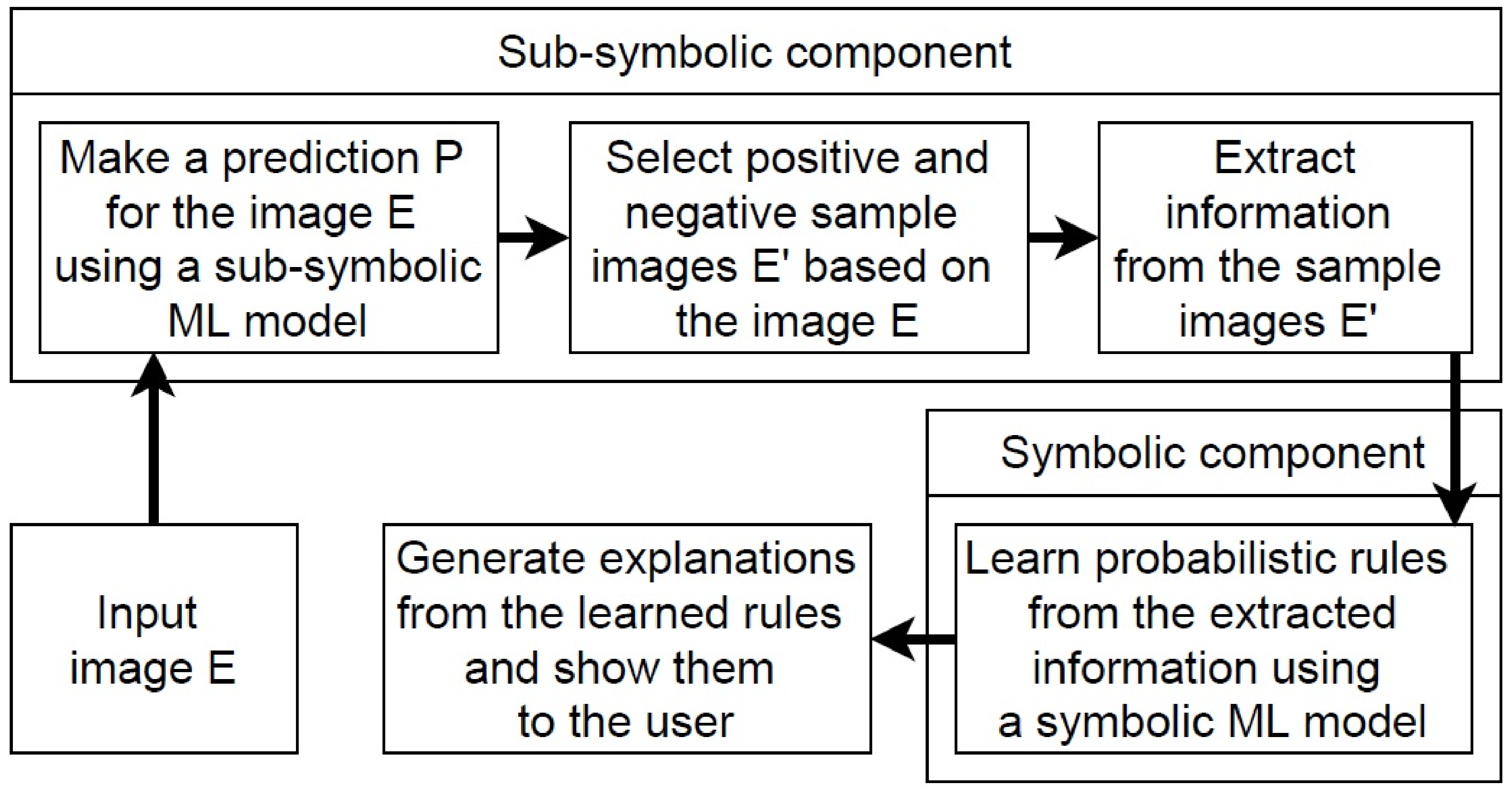
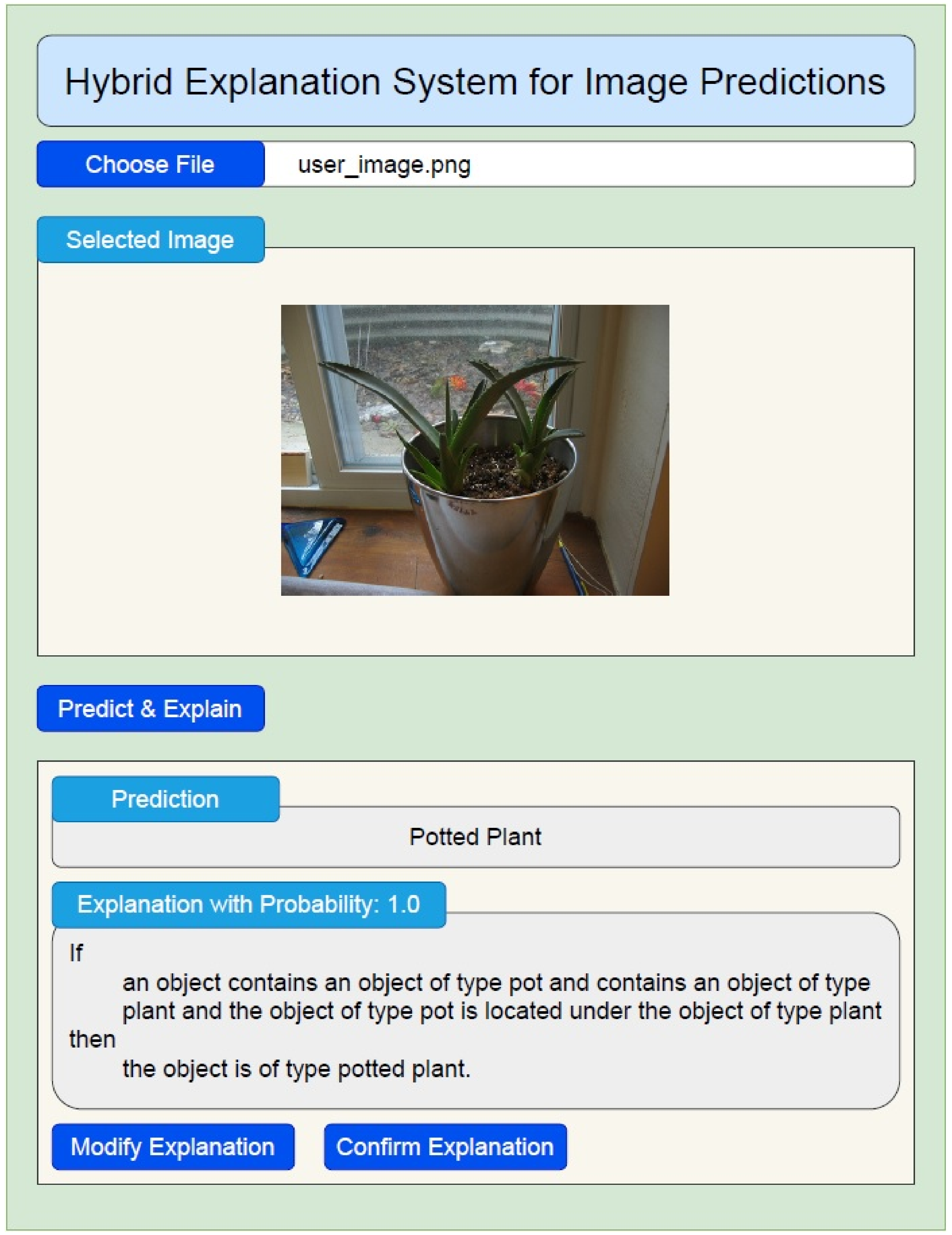

Sub-symbolic component
Make a prediction P for the image E using a sub-symbolic ML model
Select positive and negative sample images E' based on the image E
Extract information from the sample images E'
Symbolic component
Learn probabilistic rules from the extracted information using a symbolic ML model
Generate explanations from the learned rules and show them to the user
Input image E

Fig. 1: Architecture of the HESIP system.

HESIP: User Interface



Hybrid Explanation System for Image Predictions

Choose File user_image.png

Selected Image

Predict & Explain

Prediction

Potted Plant

Explanation with Probability: 1.0

If
an object contains an object of type pot and contains an object of type plant and the object of type pot is located under the object of type plant
then
the object is of type potted plant.

Modify Explanation Confirm Explanation

Fig. 2: The HESIP system is displaying the prediction and the explanation together with the probability for the selected image.

- A user clicks on the “Choose File” button to select an image.
- HESIP displays the image and the “Predict & Explain” button.
- When the user presses the “Predict & Explain” button, HESIP shows the prediction for the image and the explanation of the prediction together with the probability.
- If the user feels that there is something wrong with the generated explanation, then they can fix the incorrect information so that the HESIP system can learn a better one.
- The user can confirm or modify the generated explanation in a CNL.

Generating Explanations

- HESIP makes a prediction in the sub-symbolic component for an image selected by the user. Afterwards, HESIP selects sample images for the predicted image, extracts information of the sample images and represents the sample image information using an ontology.
- Therefore, HESIP uses the information of the sample images as example instances in the symbolic component to learn the explanatory rule for explaining the image prediction.
- Finally, HESIP generates an explanation for the image prediction from the learned rule using a bi-directional logic grammar.

```
type(A, potted_plant):1.0 :-
   type(B, pot), object(B),
   type(C, plant), object(C),
   relation(B, C, under),
   relation(A, C, contain),
   relation(A, B, contain),
   object(A).
```

⇩

If an object contains an object of type pot and contains an object of type plant and the object of type pot is located under the object of type plant then the object is of type potted plant.

Fig. 3: A generated explanation from the learned rule.

Modifying Explanations

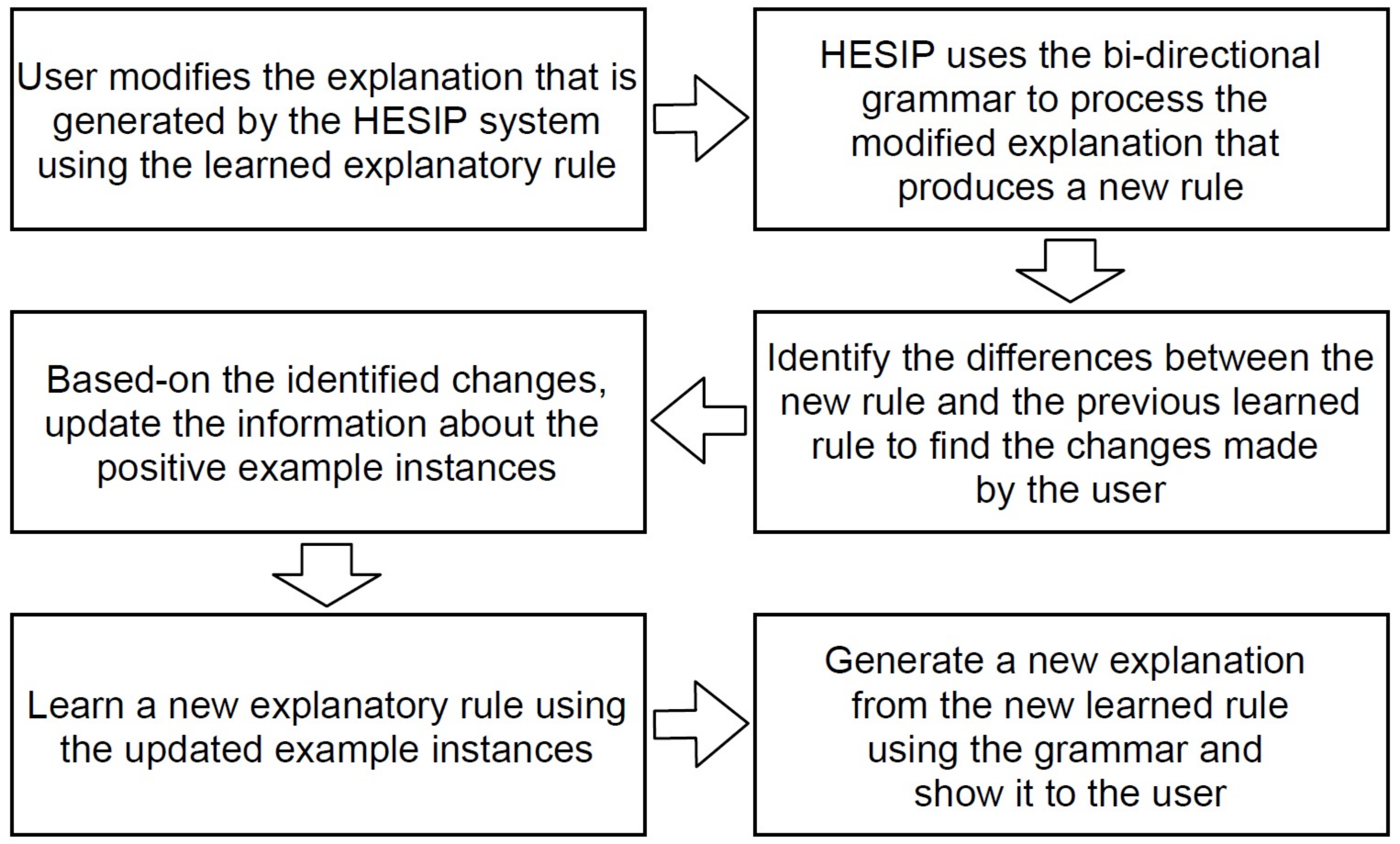

User modifies the explanation that is generated by the HESIP system using the learned explanatory rule
HESIP uses the bi-directional grammar to process the modified explanation that produces a new rule
Identify the differences between the new rule and the previous learned rule to find the changes made by the user
Based-on the identified changes, update the information about the positive example instances
Learn a new explanatory rule using the updated example instances
Generate a new explanation from the new learned rule using the grammar and show it to the user

Fig. 4: The explanation modification process of the HESIP system.