Hybrid Crowd-AI Learning for Human-Interpretable Symbolic Rules in Image Classification

Shimizu Ayame

Master's Program in Informatics
Degree Programs in Comprehensive Human Sciences
Graduate School of Comprehensive Human Sciences
University of Tsukuba
March 2023

Hybrid Crowd-AI Learning for Human-Interpretable Symbolic Rules in Image Classification

Name: Shimizu Ayame

Explainable AI (XAI) aims to create AIs that can explain the reason for their prediction. AIs are quite useful in many situations since they enable processing complex tasks in massive data requirement. Although in many cases, it is difficult for human to comprehend the decision process. The mainstream AI nowadays called deep neural network basically lacks interpretability, because the inner process using neuron firings is beyond human-interpretation. This black-box problem prevents applying AIs on high-stake tasks such as medical diagnosis, and for an AI-based society with trust, an interpretable model is required. Deriving human-interpretable symbolic rules is one of the promising ways for people to verify whether the decision is appropriate or not.

This thesis explores a hybrid crowd-AI approach to develop ML models associated with human-interpretable symbolic rules, and answers to two research questions: (RQ1) Is it possible to identify the components of ML models that correspond to predicates for symbolic rules? (RQ2) If so, what method generates good results?

The proposed method extracts subsets of data instances that activate neurons similarly from the black-box decision process of trained neural networks to enable human abductive reasoning. Crowd workers are asked to conduct abductive reasoning to provide semantics of the extracted data instance subsets in a natural language, which serve as predicates to explain the data instance subsets. The obtained semantics connects the recognition processes of AI and humans, in terms of a set of predicates with natural language descriptions that comprise symbolic rules to define target classes.

In chapter 2, the mainstream of XAIs and the discussion about black-box model interpretability, XAI evaluation, and XAI works using subsets of data instances are shown as related works. In chapter 3, the proposed method and the crowdsourcing settings are explained. In chapter 4, experiments using crowdsourcing were conducted to demonstrate the effectiveness of the system. The interpretability evaluation by crowdsourcing showed that the system enables obtaining interpretable symbolic rules. In chapter 5, the interpretability of the obtained rules and explanation is discussed. In chapter 6, contributions and answers to the two research questions are summarized as a conclusion.

This thesis provides experimental results showing that the proposed approach can obtain interpretable symbolic rules and explanations based on them.

Main Academic Advisor: Kei Wakabayashi
Secondary Academic Advisor: Masaki Matsubara

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning is an intuitive system successful in many domains, and they are the mainstream of current artificial intelligence research. However, a well-known drawback is that black-box models such as neural network models do not explain why they made the decision.

The lack of explainability in machine learning models prevents the use of AIs in high-stakes decision-making, and many methods were created to make explainable machine learning models. Explainable AI (XAI) approaches look at either the inside of the machine learning model or the input-output relationship of the model, and explanations may be made on a specific prediction or to the model itself.

The interpretation quality of an explanation by XAI models can be defined by faithfulness (how accurately the explanation represents the model behavior) and plausibility (how reasonable the explanation seems to be) [1, 2, 3]. Jacovi and Goldberg [1] point out that it is possible to generate a plausible explanation lacking faithfulness and many works on textual explanation [4, 5] have not guarantee faithfulness.

As some plausible explanations lack faithfulness, some faithful explanations are concerned about their plausibility. For example, the plausibility of explanation using attention mechanisms [6] is under discussion. There is an opinion that no explanation suitable for human cognition is obtained from internal model structure analysis [7, 8]. Since black-box models are essentially not human-interpretable, we should use explainable models instead if an explanation is necessary [8].

White-box models such as deliberative and reasoned artificial intelligence driven by knowledge provide explanation interpretable for people, and was one of the most successful AIs before deep learning arose. One of the approaches for them is to develop expert systems, based on the expert-generated rules. However, such expert systems are relatively costly since they require experts in the domain to decide on the knowledge to encode, pushing them aside from the mainstream nowadays.

In such white-box models, *logical rules* are the core components. For example, a white box model concludes that the object is a car *if* it has four wheels and a handle. The elements of such logical rules is *predicates*, which are boolean functions that return true when a particular condition holds for a given instance. For example, "x has four wheels" returns true when a taxi is given for x, but it returns false when a bicycle is given for x.

There have been attempts to connect the two models - the data-driven black box

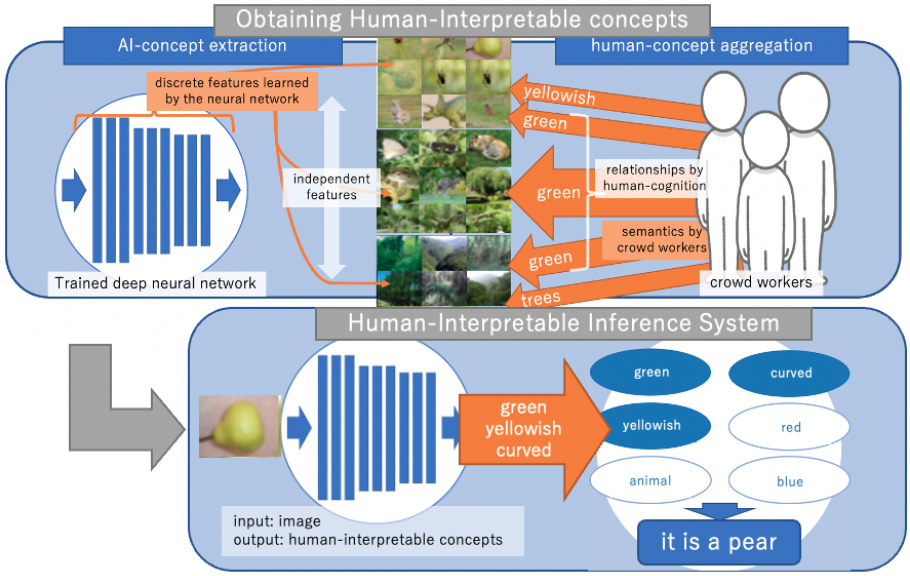

Figure 1.1: Abduction-based rule generation (ARG): We first exploits crowdworkers' abductive reasoning ability to derive predicates that comprises symbolic rules and then uses them to generate the explanations.

models and the human interpretable predicates. For example, TCAV [9] proposes a method to explain how a neural network classifier works in terms of a predefined set of predicates to explain the target classes. However, identifying predicates to be included in the pre-defined set is not easy especially when there are a large number of target classes.

This thesis addresses the problem of how to *identify* the set of predicates that explain the cognitive process of AI models for the target classes. The proposed framework, called the *abduction-based rule generation (ARG)*, first exploits crowd workers' abductive reasoning ability to derive predicates that comprises symbolic rules and then uses them to generate the explanations. The challenge is how to identify components of the ML model that correspond to appropriate predicates; for example, the predicate must not be too specific (i.e., "$\mathbf{x}$ has four wheels" instead of "$\mathbf{x}$ is a car" for a car classifier), and must be connected to easy-to-interpret semantics (i.e., "$\mathbf{x}$ is red" instead of $f_1(\mathbf{x}) > 0.001$).

Figure 1.1 shows the outline of our framework. For each subset of data instances that activate neurons similarly from the black-box decision process of trained neural networks, we ask crowd workers to conduct abductive reasoning to provide the semantics for it to provide predicates. Then, the predicates are aggregated, and the connection between the predicates and AI's cognitive process is used to explain the results of AI's output.

Research Questions. Our research questions are as follows. (RQ1) Is it possible to identify the components of ML models that corresponds to predicates for symbolic rules? (RQ2) If so, what method generates good results?

Contributions and Key Findings. The contributions and key findings are as follows.

(1) This thesis gives a human-in-the-loop framework for a novel problem of how to derive white-box symbolic rules from black-box ML models. The key idea is to exploit the human's ability for abductive reasoning. To the best of our knowledge, this work is the first to make natural language explanations based on the idea.

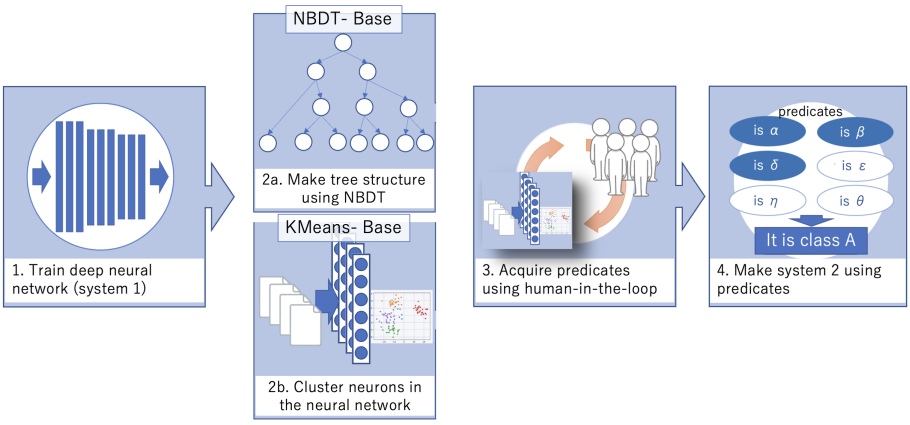

Figure 1.2: Outline of the Proposed Method

(2) This thesis gives two contrastive methods to identify appropriate components of ML models for deriving appropriate predicates in the framework. The two methods are shown as Figure 1.2. The difference between the two methods is shown in the second from the left box. The first method, which we call the NBDT-based Method, goes through the upper box process in Figure 1.2. This method uses NBDT [10], which generates a tree structure classification model from a deep learning multi-class classification model. In NBDT, each node in the tree structure classification model is a binary classifier. The other method, which we call the KMeans-based Method, goes through the the lower box process in Figure 1.2. This method makes neuron clusters using all outputs of neurons in every layer. NBDT-based Method uses the weights of the last fully-connected layer to make a hierarchical structure, while the KMeans-based Method uses all outputs of neurons in every layer. We expect to discover a suitable method for this task by exploring the two contrastive methods.

(3) This thesis reports the result of an experiment with an open-world data set and real-world crowd workers. The result shows that a model with both opacity and interpretability can be achieved by our framework.

(4) The experimental results using crowdsourcing show that ML model components using all layer output generate helpful abstract predicates, while components using only layers close to the output lead to predicates too specific.

Chapter 2

Related Work

There are various approaches in the Explainable AI (XAI) field, making it difficult to determine the pros and cons of each work [11]. Our proposed method explains the model in global (the model's behavior on the whole dataset) and local (the model's prediction of a particular instance) and performs high human-interpretation and explainability. Interpretability and explainability are used interchangeably in many cases. In this thesis, we consider interpretability as the ability to generate a convincing explanation for humans and explainability as the ability to show the inference process. As interpretability quality can be divided into faithfulness and plausibility [2, 1], we run evaluation experiments on each measurement independently.

Rudin [8] argues that the explanation of deep learning models with opaque internal structure by the attention mechanism is a posterior explanation for the inference. To avoid generating posterior reasonings as explanation, our approach does not try to explain how the deep learning model works, but rather performs its own inference by using the extracted features (predicates).

Our approach can be explained in terms of the dual-process theory [12] in psychology. In the dual-process theory, human thoughts are modeled by the interaction between black-box implicit cognition and white-box explicit cognition. The implicit cognition is called System 1; in human decision-making, it is the intuitive answer we get. The explicit cognition is called System 2, and in human decision-making, it is the logical thought we get after System 1 decisions. In human cognition, System 1 provides predictions and System 2 generates coherent explanations [13].

In our work, we define a deep learning multi-class classification model as System 1 and conduct a human-in-the-loop process with System 1 to construct a symbolic logical rules for System 2. As mentioned in the Introduction, TCAV [9] proposes a method to explain how a neural network classifier works in terms of a predefined set of predicates given in advance to explain the target classes. However, identifying the set of predicates for each class is not practical when there are a large number of target classes, such as in CIFAR100 [14].

Wan et al. [10] proposed a method called Neural-Backed Decision Trees (NBDT) that generates a tree structure classifier from a pretrained deep learning multi-class classification model. The inner nodes of the generated tree structure are binary classifiers, and the leaf nodes are the classification classes. The tree structure is made by hierarchical clustering of the last fully-connected layer weights of the pretrained deep learning model. After con-

structing the tree structure, all inner nodes are trained to perform the same outputs as the original deep learning model. NBDT can be viewed as a derivation of System 2 from System 1 since it extracts the interpretable tree structured model from the deep learning model. However, it is unclear what kind of decision is represented by each node of the tree structure constructed by NBDT. NBDT names nodes by computing the earliest common ancestor for all leaves in a subtree using WordNet [15]. As Wan et al. mentioned in their work [10], WordNet lacks concepts like object attributes and context (e.g., clocks and plates are circular, clouds and skyscrapers are both in the sky), despite the fact that such visual information is littered across NBDT. In our work, we incorporate the tree structure model constructed by the NBDT into human-in-the-loop to extract explainable features for decision-making. Our work proposes a method to reconstruct a machine learning model into a model that uses features that can be explained in natural language for decision-making.

NBDT creates a hierarchical model using the last fully-connected layer weights. As a comparison, we also propose a non-hierarchical model using all the layer weights. The model we propose uses neuron clusters, resulting from neurons clustered by their firing. Berry and Tkačik claim that brain neuron activity is organized into discrete clusters [16]. As deep neural networks imitate brain neurons, we assume that neurons of deep network classifiers function as clusters.

Chapter 3

Proposed Method

In this chapter, we clarify our design goals, present the framework overview, and describe two variations of implementation of our framework.

3.1 Problem and Framework Overview

We aim to extract components from deep learning machine and acquire human-interpretable symbols using human abductive reasoning. We name the human-interpretable symbols predicates and define them as the following.

Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ be training dataset where $\mathbf{x}_i$ is an image (data instance) and y_i is the corresponding class label. Our problem is to derive a set $\{Q_1, \ldots, Q_n\}$ of *computable*, *interpretable*, and *meaningful* predicates.

Computable When a data instance $\mathbf{x}$ is provided, we can compute the probability that $Q_i(\mathbf{x})$ is true. We denote this probability by $P_{Q_i}(\mathbf{x})$.

Interpretable Each predicate Q_i is associated with a natural language description (e.g., "$\mathbf{x}$ is mainly red", "$\mathbf{x}$ is a flower") denoted by $I(Q_i)$.

Meaningful The set of predicates are meaningful as they explain the process of deriving the conclusion.

To this end, we propose a framework that extracts predicates from deep neural network parameters trained by using the target classification dataset D.

Figure 1.2 shows two variations of implementation of our framework. In either variation, the deep neural network (DNN) is used for evaluating the predicates (i.e., calculating $Q_i(\mathbf{x})$) and for building the crowdsourcing tasks to obtain human-interpretable descriptions (i.e., obtaining $I(Q_i)$). Then, we induce symbolic rules that connect the predicates and the target classes. In this thesis, we apply white-box model analysis such as logistic regression and decision tree that statistically infers the relationship between the explanatory variables (predicates) and the target variable (class). We also apply predicate logic that gives logical rules of predictions. Logical propositions are obtained by converting the decision tree input to boolean and disassembling the decision tree. In the following sections, we explain the two methods in more detail.

Figure 3.1: Crowdsourcing task for Node Annotation in NBDT-based Method

3.2 NBDT-based Method

The first approach to extracting interpretable predicates is to leverage an existing XAI method called NBDT [10]. NBDT generates a decision tree structure whose each node corresponds to a boundary in the latent feature space learned by a DNN. The inner nodes of the generated tree structure are binary classifiers, and the leaf nodes are the classification classes. As Figure 1.2 (2a. and 2b.) shows, the proposed method first constructs the NBDT by training on the target dataset D that contains a large number of labeled images.

Our proposed NBDT-based Method attempts to give human-interpretable descriptions to each node in the NBDT by using crowdsourcing. The node that corresponds to a latent space boundary is not easy to understand its semantics for humans. Some of the nodes may even be impossible to interpret for humans, not only just absent of descriptions. Therefore, the proposed method extracts human-interpretable nodes as predicates by applying the following two processes: (1) node annotation and (2) node selection.

3.2.1 Node annotation

The basic strategy for obtaining natural language descriptions from crowd workers is instance-based comparison. Given data instance $\mathbf{x}$ and node v, the NBDT can calculate the probability of the binary decision $p_v(\mathbf{x})$. For the comparison, we choose two sets of data instances as representative of positive and negative examples regarding the binary decision. The positive data instances for node v are denoted by $\texttt{Top}_{k,\mathbf{x}}(p_v(\mathbf{x}))$, which is a set of k instances that make the probability $p_v(\mathbf{x})$ to be highest, where k is a predefined integer that specifies the number of images to be reviewed by workers. The negative counterpart is $\texttt{Top}_{k,\mathbf{x}}(-p_v(\mathbf{x}))$.

We collect descriptions of what each node in the NBDT checks by showing the two

Figure 3.2: Crowdsourcing task for Node Selection in NBDT-based Method

sets of selected images to crowd workers. Figure 3.1 is an example of the screen shown to workers in the crowdsourcing process ($k = 30$). Workers fill in the blanks of the sentence "Is the image _____?". We make tasks from all the inner nodes in NBDT. We also make tasks switching "Yes" and "No" for all tasks. We assign multiple workers to each task to obtain multiple descriptions for each node and obtain sets of descriptions $A_{v,l,j}$ where v is the node, l is the "Yes" and "No" switching, and j is the worker number.

3.2.2 Node selection

Some of the descriptions collected from workers may be inappropriate due to both human errors and uninterpretability of the node. Uninterpretable nodes include nodes that do not contribute to image classification (nodes not used), nodes representing concepts that cannot be denoted by English, and nodes representing concepts that don't correspond to human cognition. Therefore, we evaluate the collected descriptions using crowdsourcing as well. A helpful description satisfies the two conditions: it conforms to the actual binary classifier's judgment and is human-interpretable.

Figure 3.2 is an example of the screen shown to workers in the crowdsourcing process. The positive and negative examples of images used in the node annotation $\texttt{Top}_{k,\mathbf{x}}(p_v(\mathbf{x}))$ and $\texttt{Top}_{k,\mathbf{x}}(-p_v(\mathbf{x}))$ are randomly placed, and a description $A_{v,l,j}$ is shown on the top of the screen. For each collected description $A_{v,l,j}$, we ask workers to choose all pictures that match the description. We calculate the F1 score of the worker's selection, where selected images that come from $\texttt{Top}_{k,\mathbf{x}}(p_v(\mathbf{x}))$ are true positive cases, selected images from $\texttt{Top}_{k,\mathbf{x}}(-p_v(\mathbf{x}))$ are false positive cases, non-selected images from $\texttt{Top}_{k,\mathbf{x}}(p_v(\mathbf{x}))$ are false negative cases, and non-selected images from $\texttt{Top}_{k,\mathbf{x}}(-p_v(\mathbf{x}))$ are true negative cases. We defined that descriptions (e.g., the sentence shown at the top in Figure 3.2) with high F1 scores are human-interpretable descriptions. In our experiment, we empirically decided the threshold

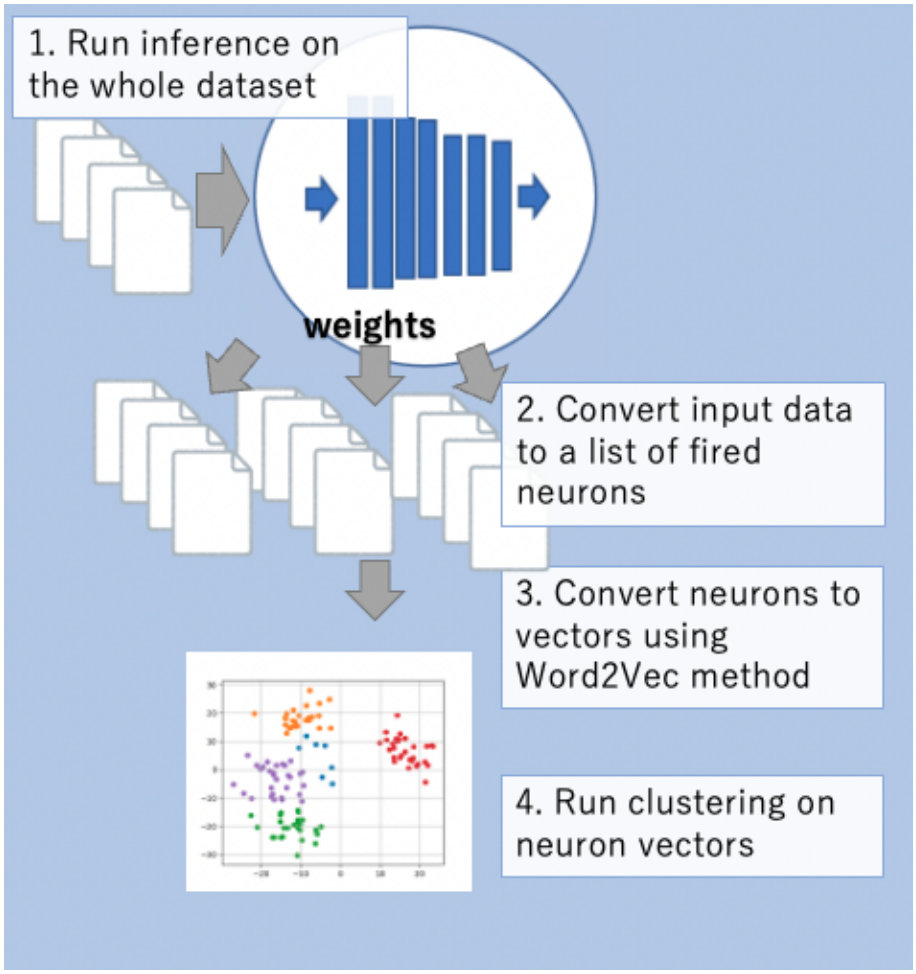

Figure 3.3: Converting Neurons to Vectors

of F1 score to be 0.67.

3.2.3 Predicates in NBDT-based Method

Using the F1 scores calculated in the previous section, we define predicates $\{Q_1, \ldots, Q_n\}$ as descriptions having F1 scores that are higher than the threshold, 0.67. The evaluation function for Q_i is based on the NBDT's function. The probability that $Q_i(\mathbf{x})$ is true, denoted by $P_{Q_i}(\mathbf{x})$, is defined as the probability of the binary decision $p_v(\mathbf{x})$ for the node v that corresponds to Q_i. The interpretable description $I(Q_i)$ is a set of descriptions having F1 scores higher than the threshold.

By using the extracted predicates, a new classification model is created. We make a logistic regression model by using the output probabilities of the inner nodes $(P_{Q_1}(\mathbf{x}), \ldots, P_{Q_n}(\mathbf{x}))$ as a feature vector. Each feature used in the logistic regression model has a corresponding natural language annotation, making the inference human-interpretable.

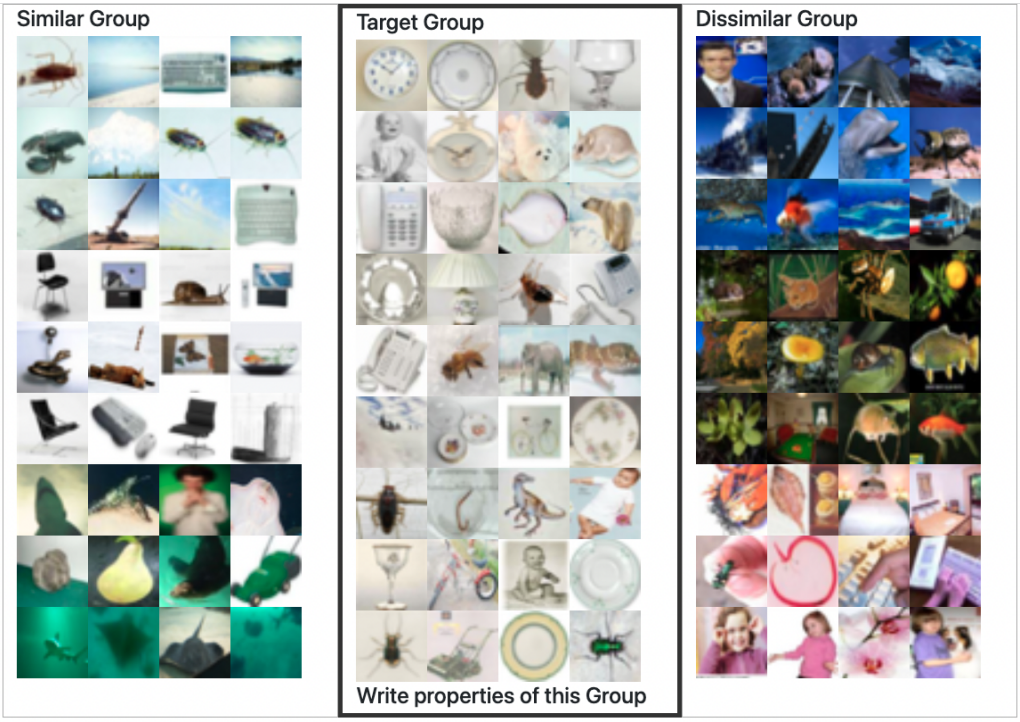

Figure 3.4: Crowdsourcing Task giving Properties to Neuron Cluster

3.3 KMeans-based Method

Another approach to extracting predicates is to examine the neuron activities in a trained DNN. While deep networks contain millions of neurons, it is not probable that all the neurons work differently and play independent roles from each other. Rather, we assume that there are some groups of neurons that are activated in response to similar visual features. This consideration motivates us to propose the KMeans-based Method that makes neuron clusters and treats them as predicates.

In this method, we first train a DNN with training data D as Figure 1.2 (1.) shows. (In our experiment, we used a pretrained ResNet50 [17] and fine-tuned it with D.) Then, we make neuron clusters by using the trained DNN. Assuming that each neuron cluster represents an image feature, we make neuron vectors from all neuron firings when the whole dataset passed the network.

Figure 3.3 shows the process to compute neuron vectors.

1. Run inference and obtain neuron firings on the whole dataset.
2. Split the neuron firings into three sections by the network layers; top layers, which are close to the input; middle layers, which come next to the top layers; and bottom layers, close to the output.
3. Defining neurons firing above certain levels as fired neurons, convert each input data in each layer section to a list of fired neurons. The strength of the firing orders the neurons listed.

4. Considering each neuron as words and neuron lists as documents, use the Word2Vec [18] algorithm to convert neurons to vectors.

5. Run k-means clustering on neuron vectors.

We set $k = 100$ for clustering, meaning that 3 (number of layer sections) $\times 100$ (number of clusters per layer sections) $= 300$ neuron clusters are made.

3.3.1 Collecting Predicates

We collect the properties of the image using crowdsourcing. Figure 3.4 is an example of the screen shown to workers in the crowdsourcing process.

Images in the center represent the target neuron cluster in which we want workers to write properties. Images on the left side represent neuron clusters close to the target image cluster, and images on the right side represent neuron clusters far from the target neuron cluster.

3.3.2 Aggregating Predicates

Each description collected in crowdsourcing corresponds to a neuron cluster, although the predicate may not be unique to a neuron cluster. Figure 1.1 shows an example of predicate "green" corresponding to multiple neuron clusters.

By embedding the descriptions by pretrained BERT [19], we calculate cosine similarity between every description. The description pairs with cosine similarity above a threshold are connected as similar descriptions, and descriptions connected with at least one path are grouped as a predicate. Figure 3.5 shows how descriptions are aggregated to predicates.

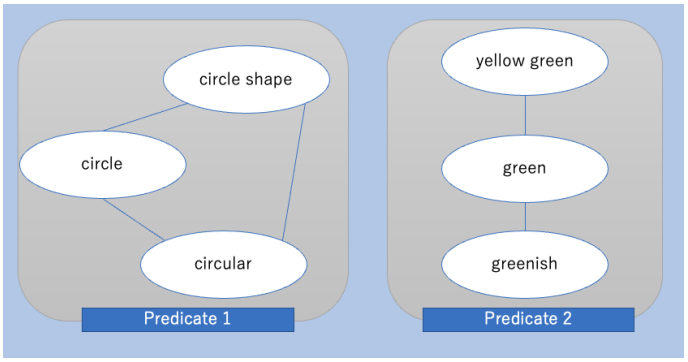

Figure 3.5: Aggregating Descriptions

3.3.3 Dual Process Model

In this section, we explain how we exercise the predicates and extract logical rules.

Decision Tree Model

Using crowdsourcing results, we make a new model combining System 1 and System 2 architecture. The input of this new model is the firing of neuron clusters when an image

is given to deep neural network model (System 1). We standardized the neuron firings, and defined neurons with firing above the threshold as fired neurons. From the predicate aggregation process, each neuron cluster is attached to one or more predicates. Using the list of neuron cluster firings, we obtain predicate firings. The input of the new model is made by the following steps.

1. Obtain a list of neurons l_i fired by passing data $\mathbf{x}_i$ to the deep neural network.
2. Convert the neuron IDs in the list of fired neurons l_i to the corresponding neuron cluster IDs.
3. Make neuron cluster firing vector v_i from the number of times each neuron cluster appeared in list l_i. The size of v_i is the number of neuron clusters, and each element is the number of each neuron cluster that appeared in the list.
4. Using the correspondence of neuron clusters and predicates, aggregate v_i by predicates and make predicate activation vector p_i. The size of p_i is the number of predicates, and each element is the number of each predicate that appeared in v_i.

The tree decision model uses p_i, the firing of predicates as the input feature (System 2). Each feature used in the tree decision model has a corresponding natural language annotation, making the inference human-interpretable. We also made a tree decision model using neuron cluster vector v_i as the input vector as a comparison of human-interpretability and machine interpretability.

Logic Model

By converting the input of the decision tree model to a boolean and disassembling the decision tree to a logical proposition, we make a logic model. As the decision tree model, the input of this new model is the firing of predicates when an image is given to deep neural network model (System 1). Each neuron cluster is attached to one or more predicates, so using the list of neuron cluster firings, we obtain predicate firings. We convert the predicate firing to boolean by rounding up the predicate firing above a threshold as *True Predicates* and others as *False Predicates*.

The input of the new model is made by the following steps.

1. Obtain neuron firings n_i by passing data $\mathbf{x}_i$ to the deep neural network.
2. Apply standardization on elements of n_i by each neuron layer to make each layer of n_i follow a normal distribution.
3. Make vector p_i by aggregating the list of neurons n_i to predicate firings using the correspondence of neuron clusters and predicates. The size of p_i is the number of predicates, and each element is the aggregated firing of predicates.
4. Convert p_i to binary predicate activation vector b_i by rounding up the firing of each element in p_i to 1 if the amount of the aggregated firing is above 0.6 and otherwise to 0. The size of b_i is the number of predicates, and each element is the aggregated firing of predicates in boolean.

The logic model uses b_i, the firing of predicates in boolean as the input feature (System 2). A tree decision model using b_i is made to obtain the relationships between class predictions and b_i. As each element in b_i represents a predicate, and predicates correspond to concepts in natural language, the relationships between b_i and the predicted class is compatible with predicate logic.

Chapter 4

Experiments

We conducted an experiment to compare the methods in terms of meaningfulness and interpretability of the extracted predicates. In evaluating the results, we divided the evaluation measurements in human-interpretability.

4.1 Experiment Settings

We used the CIFAR100 dataset [14] in our experiment because we expected that it would generate not a few number of predicates. CIFAR100 dataset is a set of 32×32 color images in 100 classes. We first used the training data of CIFAR100 for extracting predicates and generate the models with the extracted predicates, and compared the performance with its test data.

Table 4.1: The Number of Crowd Workers for each Task

Method	Task Type	The Number of Workers
NBDT-based	Node Annotation	7
NBDT-based	Node Selection	3
KMeans-based	Collect Neuron Property	7

The crowdsourcing platform we used in this experiment is Amazon Mechanical Turk[1]. Table 4.1 summarises the number of crowd workers who perform each task in the experiment. Since the total number of tasks and workers depends on the method, we will show them in the following sections. We paid 0.2 USD to each worker for each task. We set 3 layer sections and 100 clusters per layer section in neuron clustering.

Table 4.2: The Number of Crowd Workers: Evaluation Tasks

Method	Evaluation Measurement	The Number of Workers
NBDT-based	Human-Interpretation	3
KMeans-based	Human-Interpretation	3
Logic-model	Faithfulness	3
Logic-model	Plausibility	3

In evaluating the faithfulness of predicates processed in Logic-model, we showed crowd workers the predicate (natural language annotations) and group of images that return large

[1] https://www.mturk.com/mturk

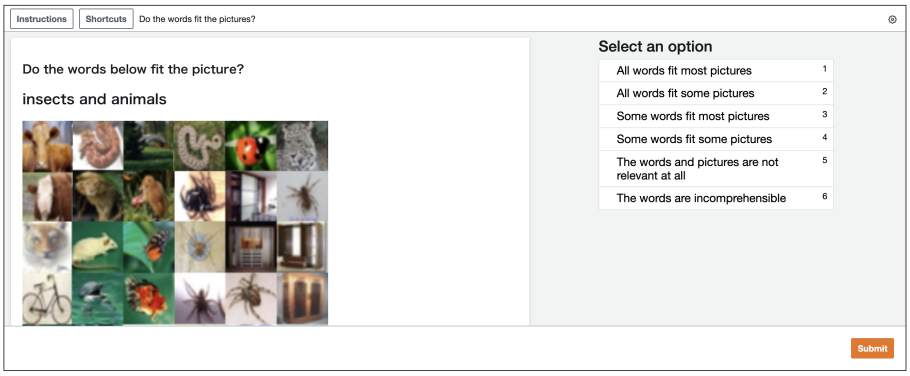

Figure 4.1: Crowdsourcing Experiment for Evaluating Predicates

p_i value. We asked the workers if the predicate fits the group of images.

Table 4.2 summarises the number of crowd workers who perform each task in the experiment. Figure 4.1 is an example of the screen shown to workers in the crowdsourcing process. In evaluating the plausibility of logic processed in Logic-model, we selected two classes ("rose" and "tulip") in CIFAR100 since explanations are relative and the generated logic becomes too redundant for workers to evaluate the plausibility. Figure 4.2 is an example of the screen shown to workers in the crowdsourcing process.

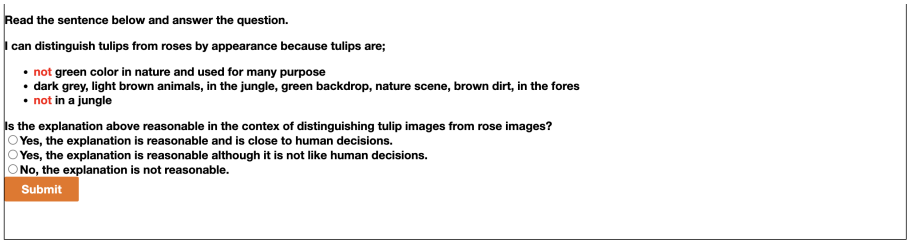

Figure 4.2: Crowdsourcing Experiment for Evaluating Logic

4.2 Results

4.2.1 Accuracy of the Obtained Models with Extracted Predicates

Table 4.3 shows the accuracy of each result of the original neural network models and the generated predicate-based models (we put the circle in the Explainable column to denote it is the generated model using the extracted predicates) for the test data. For the models, higher accuracy means having more meaningful predicates.

In the NBDT-based Method, the accuracy of the classifier using human-interpretable nodes was 66.64%, which is about 11 points lower than NBDT scores. In the KMeans-based Method, the accuracy of the classifier differed by the number of predicates. When aggregating the 2100 semantics annotating the 300 neuron clusters to bigger groups such as 100 predicates, the classifier accuracy decreased to 55.70%, which is over 25 points lower than the base DL model. Although, the classifier accuracy is 78.53%, which is only 3 points lower than the base DL model, when aggregating the semantics to 500 predicates. The

Table 4.3: Test Data Accuracy in CIFAR100

Method	Explainable	Accuracy
NBDT (deep tree structure)	△	77.09
NBDT-based (all nodes)	△	73.83
NBDT-based (interpretable)	○	66.64
KMeans-based (DL model)	×	81.83
KMeans-based (100 predicates)	○	55.70
KMeans-based (200 predicates)	○	61.88
KMeans-based (300 predicates)	○	71.78
KMeans-based (400 predicates)	○	76.45
KMeans-based (500 predicates)	○	78.53
KMeans-based (neuron clusters)	△	77.97

result using 500 predicates is 0.56 points higher than the KMeans-based Method using the original non-aggregated 300 neuron clusters as an input, meaning human annotations are informative not only for humans but also for AI classification.

Table 4.4: Accuracy in 2 Class Classification("rose" and "tulip")

Data type	Max tree depth	Accuracy
train data	5	67.5
train data	12	90.4
test data	5	60.2
test data	12	73.5

Table 4.4 shows the accuracy of each result of the logic generating KMeans-based models for the training data and test data. The *max tree depth* column shows the number of nodes along the longest path from the root of the tree to its farthest leaf node. A large max depth amount makes large trees and enables sensitive sorting of the given data set. Although, this leads to generating an overfitted model, which cannot predict well on unknown data. Considering human-interpretability, logic generated from large tree model becomes long and difficult to comprehend.

On the training data, the accuracy of the classifier set to max tree depth 5 was 67.5% and the accuracy of the classifier set to max tree depth 12 was 90.4%. On the test data, the accuracy of the classifier set to max tree depth 5 was 60.2% and the accuracy of the classifier set to max tree depth 12 was 73.5%.

The scores on the classifier set to max tree depth 12 were higher than those of the classifier set to max tree depth 5, but the difference in the accuracy was smaller on test data. This means that generalization ability is not so high on the classifier set to max tree depth 12.

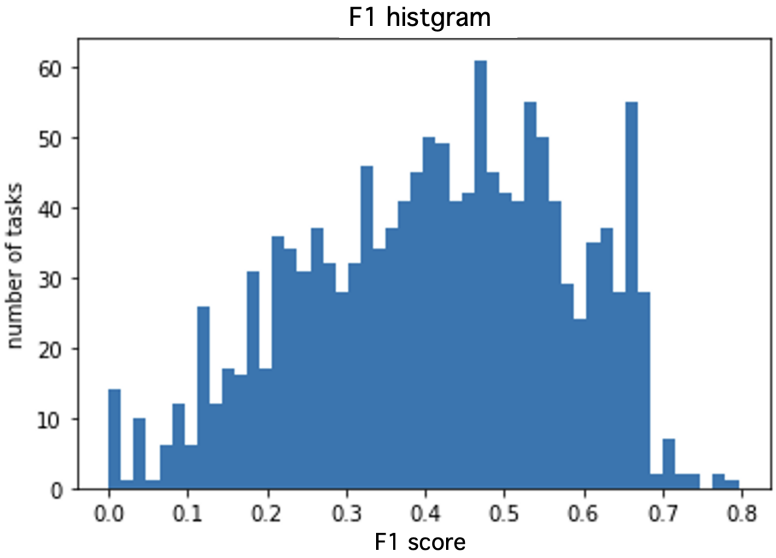

Figure 4.3: Distribution of F1 scores in Node Selection Experiment

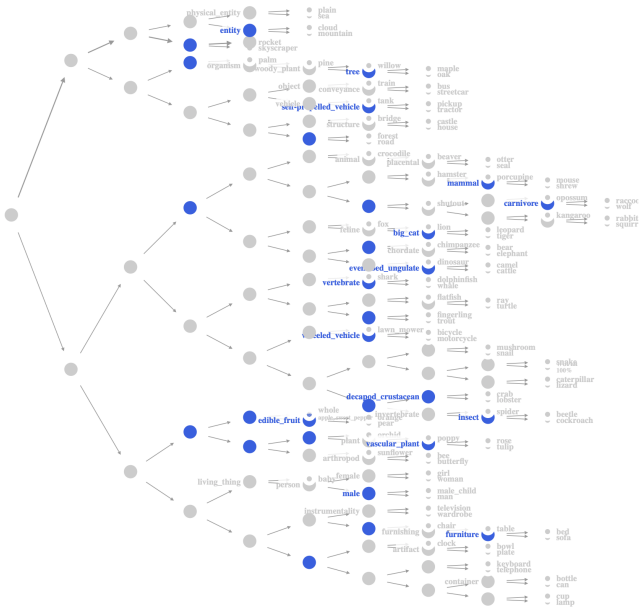

Figure 4.4: Human-interpretable Nodes in NBDT

4.2.2 Crowdsourcing Cost

Table 4.5 shows the crowdsourcing costs. NBDT-based Method, taking 2 steps in obtaining explainable annotations was costly in both time and money. Table 4.6 shows the time taken when an expert (the author) worked on the tasks. Note that the task aims to acquire a widely accepted explanation. We do not mean to say that an expert can give always give the best answer; Cognition such as color category boundaries may differ by culture and biological differences. The overall time for the whole task is estimated from the average of

the time taken to finish 20 tasks. The overall time is slightly shorter when an expert does the whole task.

Table 4.5: Crowdsourcing Cost Summary

Method	Step	Time	Money
NBDT-based	Collecting Annotations	3.25 hours	$504.00
NBDT-based	Selecting Annotations	7.57 hours	$141.12
NBDT-based	Overall	10.82 hours	$645.12
KMeans-based	Collecting Annotations	4.03 hours	$216.00

Table 4.6: Time Cost: An Expert Annotating

Method	Step	Number of Tasks	Time(20 tasks)	Estimated Time
NBDT-based	Collecting Annotation	196	8.1 minutes	1.32 hours
NBDT-based	Selecting Annotation	1372	5.7 minutes	6.48 hours
NBDT-based	Overall			8.78 hours
KMeans-based	Collecting Annotation	300	14.1 minutes	3.53 hours

4.2.3 Detailed Analysis

We then show the detail of how NBDT-based Method, KMeans-based Method, and the logic generation using KMeans-based Method worked and how interpretable the extracted predicates and logic are. Since the annotation is done by crowdsourcing, the predicates may include grammatical mistakes or may not make sense due to human error and spammers.

NBDT-based Method

The number of inner nodes (binary classifiers) in the tree structure generated by NBDT was 98, and 98×2 (for each Yes/No) $\times 7$ (number of workers per task) $= 1,372$ annotations were collected through crowdsourcing. 29 inner nodes with an F1 value of 0.67 or higher were used as features in the human-interpretable logistic regression model. Figure 4.3 shows the distribution of F1 scores in the node selection experiment. Figure 4.4 shows the 29 human-interpretable nodes in NBDT. Figure 4.4 shows that although the extracted nodes were only those at the fourth level or lower, within the lower levels, the extracted nodes are not concentrated on specific classes.

Table 4.7: Example of weights in NBDT-based Method (The weights use answers from crowdsourcing and may include grammatical mistakes)

Class		Explanation	Weight
tulip	top weights	flowers	0.2456
		contains green color	0.0858
		in trees	0.0800
	bottom weights	mainly pink	-0.2934
		fruits	-0.1102
		yellow round	-0.0946
rose	top weights	flowers	0.1382
		in a chairs	0.1309
		mainly pink	0.1161
	bottom weights	fruits	-0.1707
		a piece of living room furniture	-0.1251
		blurred	-0.1040

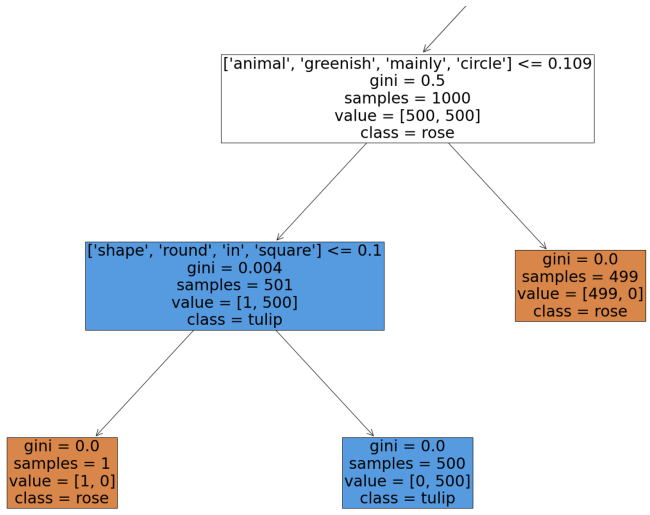

Figure 4.5: Example of Decision Tree using Predicates (Due to the limited space, only the most frequent words in the predicate is shown in the figure)

As an example of what the logistic regression model learned, we show the top 3 and bottom 3 weights of the "tulip" class and the "rose" class in Table 4.7. Tulip and rose are relatively close classes in CIFAR100 dataset, and they both have description "flowers" as their most considerable weight. The second largest weight in rose is "in a chairs", which probably comes from the fact that rose include pictures of cut flowers taken indoors. The

Table 4.8: Example of weights in KMeans-based Method (The collected properties use answers from crowdsourcing and may include grammatical mistakes)

Class		Collected properties	Weight
tulip	top weights	Mainly warm coloring with a green background.	0.1937
		greenish; in forest;	0.1791
		Mostly cool shades of grey; green; blue and brown.	0.0840
	bottom weights	Predominant gray colors	-0.0658
		Mostly focusing vehicles.	-0.0528
		glass and curved	-0.0474
rose	top weights	mainly red;	0.2609
		Outdoor and greenish background.	0.1088
		Mainly focusing Furniture.	0.0886
	bottom weights	in the jungle; green backdrop;	-0.0308
		Mostly focusing vehicles.	-0.0305
		Mainly brown; contains animals.	-0.0277

second largest weight in tulip is "contains green color", which can be assumed as a result of large number of tulip pictures taken in gardens. The weights learned also show that "mainly pink" is positive weight in rose but negative in tulip, also showing that classifiers focus on colors when distinguishing rose and tulip.

KMeans-based Method

We collected 3 (layer sections) $\times 100$ (neuron clusters) $\times 7$ (number of workers per task) $= 2100$ annotations were collected through crowdsourcing.

We made a logistic regression model using neuron cluster firings. As an example of what the logistic regression model learned, we show the top 3 and bottom 3 weights of the "tulip" class and the "rose" class in Table 4.8. The "Collected properties" are answers from crowdsourcing. We instructed workers to write a sentence or a word, but we did not give strict rules in the annotation task. Some answers from workers include grammatical mistakes. Unlike results in Table 4.7, tulip and rose did not have common neuron clusters in top 3 weights. Although, the fact that brown and green is helpful in distinguishing tulip, and looking for red colors and furniture in distinguishing rose seem to be shared in both NBDT-based Method and KMeans-based Method.

Figure 4.5 shows a part of the decision tree model using aggregated predicates. Due to the limited space, only the most frequent words in the predicate are shown in the figure. The "tulip" class and the "rose" class are divided by the sensitivity of predicate represented

by words such as ("animal", "greenish", "mainly", "circle") and ("shape", "round", "in", "square"). By using decision tree model and human-interpretable predicates, we can extract symbolic rules such as:

- ("animal", "greenish", "mainly", "circle") $\wedge$ ("shape", "round", "in", "square") $\rightarrow$ tulip
- ("animal", "greenish", "mainly", "circle") $\vee$ *not* ("shape", "round", "in", "square") $\rightarrow$ rose

Each set of words in parentheses corresponds to a single predicate. The words are the most frequent words in the set of workers' descriptions given to the predicate.

Logic Model

We collected 173 predicates by aggregating the 2100 annotations given from crowd workers. The predicates may include some grammatical mistakes since they are answers directly from crowdsourcing. We made a logistic tree decision model using predicates and neuron firings corresponding to the predicates. By disassembling the decision tree, we obtained logic showing the path to predict the classification class.

Examples of the logic generated are shown in Table 4.9. Since the logic is based on tree decision model, all logic generated from the same decision tree start from the same predicate. As results in Table 4.7 and Table 4.8 show, tulip and rose seem to be distinguished by colors. As the top 2 predicates in the original decision trees are "green color in nature and used for many purpose" and "dark grey, light brown animals, in the jungle, green backdrop, nature scene, brown dirt, in the fores", which seems to represent green and brown, the fact that brown and green helps distinguish tulip does seem to be shared in the logic model. Although, generated logic shows rose may contain green and brown in some cases.

Table 4.9: Generated Logic (The predicates use answers from crowdsourcing and may include grammatical mistakes)

Predicted Class	max tree depth	Logic
tulip	5	green color in nature and used for many purpose dark grey, light brown animals, in the jungle, green backdrop, nature scene, brown dirt, in the fores **not** black **not** leg **not** white rectangular
	12	green color in nature and used for many purpose dark grey, light brown animals, in the jungle, green backdrop, nature scene, brown dirt, in the fores black **not** citrus fruits **not** in a jungle
rose	5	green color in nature and used for many purpose dark grey, light brown animals, in the jungle, green backdrop, nature scene, brown dirt, in the fores black **not** citrus fruits **not** brown, squared shaped objects
	12	green color in nature and used for many purpose dark grey, light brown animals, in the jungle, green backdrop, nature scene, brown dirt, in the fores black **not** citrus fruits in a jungle insects and animals dark background object in the center lives in australian continent poisonous venom rounded shapes with lines tallest buildings

4.3 Interpretability Evaluation

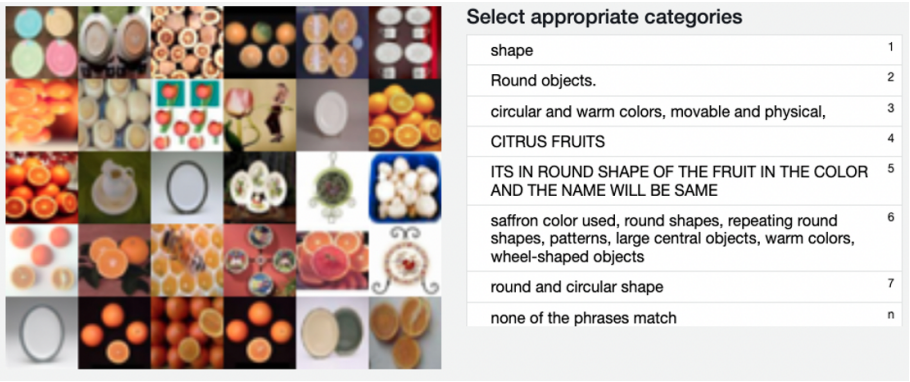

Figure 4.6: Crowdsourcing Experiment for Evaluating Annotations

We ran a crowdsourcing task to evaluate the human-interpretability of annotations associated to the predicates.

Figure 4.6 is an example of the evaluation task of annotation interpretability. A set of images and annotations are shown, and workers select all annotations out of seven annotations that suites the sets of images. Each set of images is shown to 3 workers, and we define the annotation as interpretable when at least one worker selects it.

Table 4.10: Ratio of approved Predicates

	Interpretable predicates(%)
NBDT-based Method	64.72
KMeans-based Method	68.14

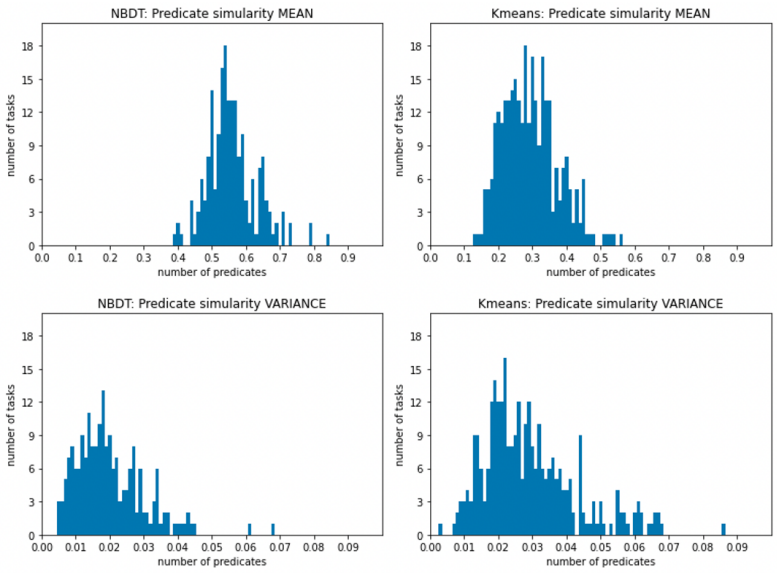

Figure 4.7: Predicate similarity in each tasks

Table 4.10 shows the result of the evaluation task. The interpretable predicates score

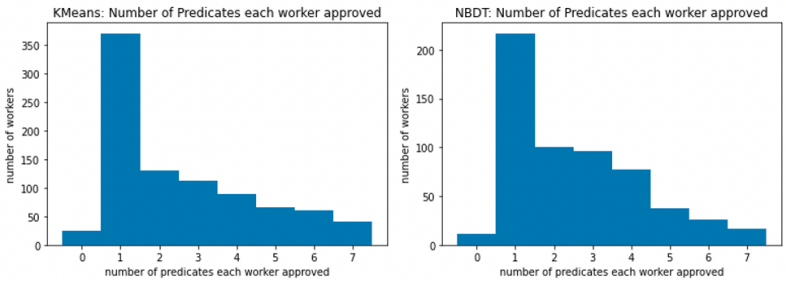

Figure 4.8: The Number of Predicates each worker approved

shows the ratio of sets of images having at least one interpretable predicate.

Figure 4.7 shows the similarity of the predicates in each tasks. KMeans was successful than NBDT in gathering predicates in variation. Figure 4.8 shows the number of predicates each worker approved. Although each worker were able to approve at most seven annotations, most workers approved only one predicate (Figure 4.8). In the results of both NBDT-based Method and KMeans-based Method, most sets of images had at least one human-interpretable annotation for the predicates, showing the effectiveness of our methods in extracting human interpretable predicates.

Figure 4.9 is an example of the evaluation task of predicate interpretability. A set of images and predicate is shown, and workers select the degree of the fitness of predicate and images from 4 options, “All words fit most pictures”, “All words fit some pictures”, “Some words fit most pictures”, “Some words fit some pictures”, “The words and pictures are not relevant at all”, and “The words are incomprehensible”. This task evaluates the faithfulness of the predicates. Each set of images is shown to 3 workers.

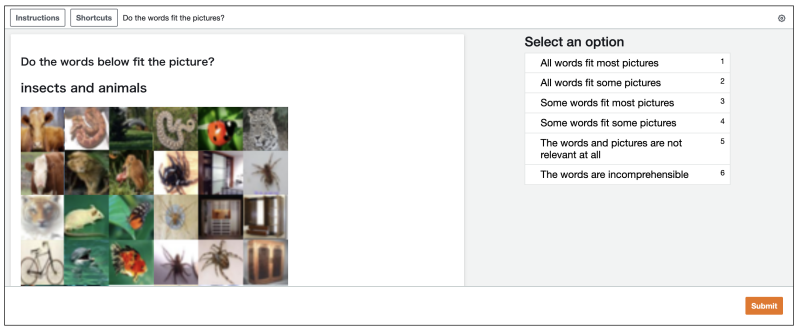

Figure 4.9: Crowdsourcing Experiment for Evaluating Predicates

Figure 4.10 shows the result of the faithfulness evaluation task. About half of the workers answered that all or some words in the predicate fit most pictures, and nearly 90% of the workers answered that all or some words fit at least some pictures. Although, considering the fact that predicates are generated by human annotations, this result does not seem surprising.

Table 4.11 shows the results of aggregating answers of each 3 workers on the same task. Defining the 6 answers, “All words fit most pictures”, “All words fit some pictures”, “Some words fit most pictures”, and “Some words fit some pictures” as “fit” answers and “The words and pictures are not relevant at all” and “The words are incomprehensible” as “not

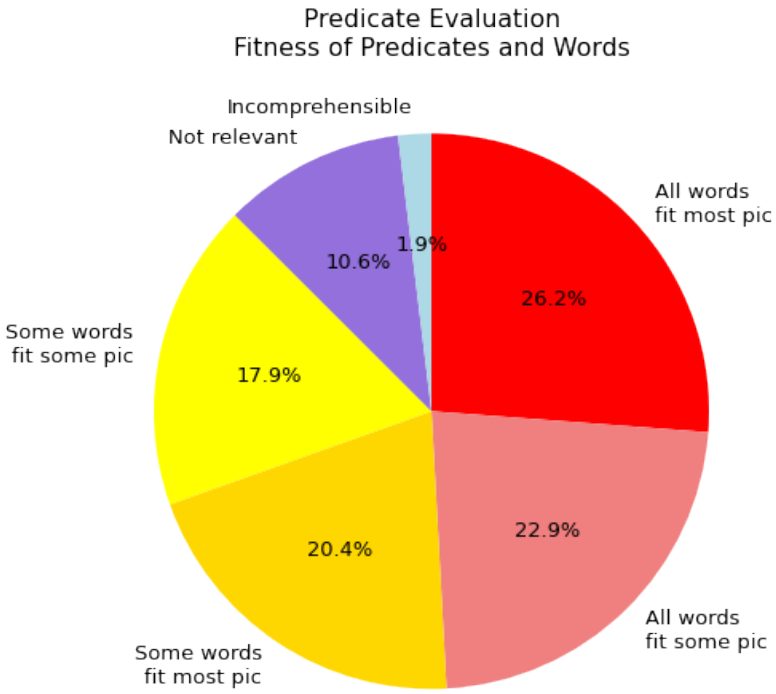

Figure 4.10: Result of Predicate Evaluation

Table 4.11: Predicate evaluation: Aggregated answers

3/3 workers answering fit	70.5%
3/3 workers answering not fit	2.9%
mixed answers	26.6%

fit" answers, majority of the predicates seem to be fitting the images, although the degree of the fitness differs within the workers. It is also shown that some predicates definitely do not fit the images, which is a result of spam answers.

Figure 4.11: Good Predicate (An annotation given from worker): “animal in sea”

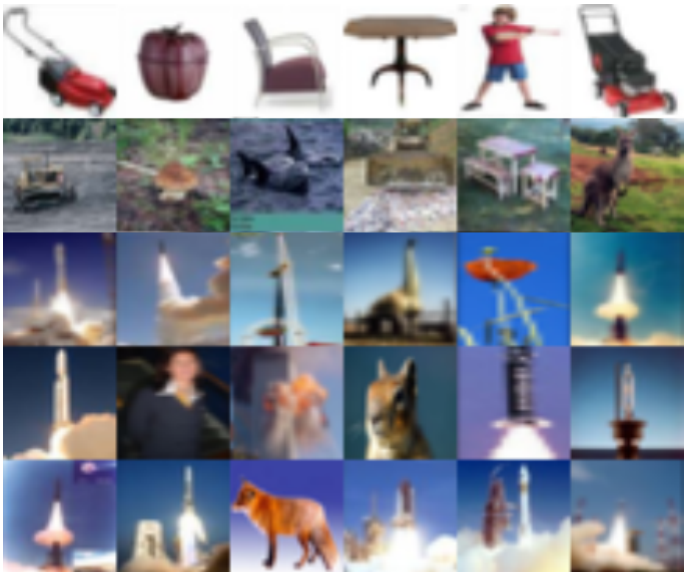

Figure 4.12: Bad Predicate (An annotation given from worker): “dolphin”

Figure 4.13: Bad Predicate (An annotation given from worker): “I am going the ocean that time I am see the tiger and men. A colour is bule.”

Figure 4.11, 4.12 and 4.13 show examples of good and bad predicates. The predicates are crowdsourced and may include grammatical mistakes. All 3 workers answered Figure 4.11 as “All words fit most pictures”, all 3 workers answered Figure 4.12 as “The words and pictures are not relevant at all”, and all 3 workers answered Figure 4.13 as “The words are incomprehensible”. Predicate “dolphin”, which is shown as Figure 4.12 is aggregation of 4 neuron clusters, and 3 out of 4 neuron clusters included few dolphin images. Since the set of images corresponding to predicates was made by choosing random images from the related neuron clusters, dolphin images included in the original neuron clusters were excluded by chance. A neuron cluster without any dolphin image is likely to be caused by a spammer giving random annotations.

Predicate “I am going the ocean that time I am see the tiger and men. A colour is bule.”, shown as Figure 4.13 is aggregation of 6 neuron clusters. The annotation is from a single crowd worker, doubtless to be a spammer giving annotations irrelevant to the task.

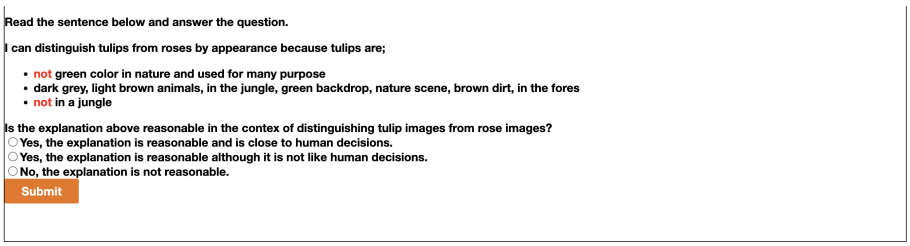

Figure 4.14: Crowdsourcing experiment for Evaluating Logic

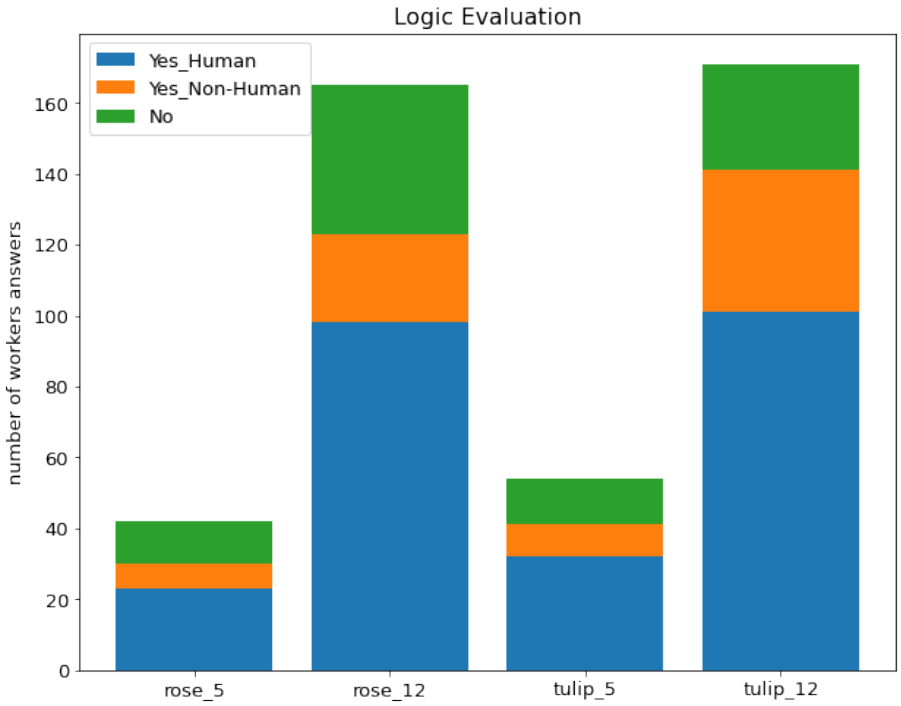

Figure 4.15: Result of Logic Evaluation

Figure 4.14 is an example of the evaluation task of logic interpretability. The generated logic is shown, and workers select the degree of plausibility of the logic from 3 options, “Yes, the explanation is reasonable and close to human decisions.”, “Yes, the explanation is reasonable although it is not like human decisions.”, “No, the explanation is not reasonable.”. Each logic is shown to 3 workers.

Figure 4.15 shows the result of the plausibility evaluation task. The ratio of the answers did not greatly differ by experimental settings, and in all experimental settings, most workers answered that the logic is reasonable and close to humans.

Chapter 5

Discussion

As results shown in Table 4.3 and Table 4.10, we succeeded in appropriately identifying components of ML models to derive predicates for symbolic rules, which answers yes to the research question **RQ1**.

The second research question was about discovering methods that generate good results in human computation. Quantitative evaluation of good results in this case is difficult, but we conclude that KMeans-based Method performs better for this task.

Figure 5.1 shows an example of sets of images made by NBDT-based Method and predicates collected by crowdsourcing. The left set of images seems to show clocks and the right seems to show plates. Since NBDT-based Method makes image sets from binary classifier results, many image sets appeared to be a mix of 2 classification classes, making it challenging for the workers to come up with abstract predicates.

Figure 5.2, shows an example of image groups made in KMeans-based Method and predicates collected by crowdsourcing. Considering "Dissimilar Group" plain cold-colored images and "Similar Group" being a mix of warm-colored images and images showing circles, the target group in the center seems to react to circles and warm colors images. Compared to the NBDT-based Method, KMeans-based Method seemed to show abstract features. Deep learning is said to capture features by learning the features step-by-step in the deep neural network layers. We assume that abstract properties such as colors and shapes are caught in the deep learning model's top to middle layers. The reason for KMeans-based Method successfully capturing abstract features is presumably because it uses all layers' outputs.

In both methods, we asked workers to provide visual descriptions such as "warm colors" and "triangle-shaped." However, most workers gave predicates based on human knowledge, such as "Things to measure time" and "Large vehicle." which is not a helpful rule in image classification. Considering the dual-process theory, we can say that answering the names of things shown in an image is a System 1 process, and seeking for properties shown in the image is a System 2 process for workers. Since System 2 process is more costly in time, it seems reasonable for workers to give names of object instead of abstract properties.

The results of crowdsourcing showed that most of the logic generated is somewhat plausible but they do not seem clear and convincing enough as human-made explanations. The predicate, "green color in nature and used for many purpose" is the root node in distinguishing class "rose" and "tulip". Figure 5.3 and 5.4 are set of images of a neuron cluster attached to the predicate, "green color in nature and used for many purpose". Referring to

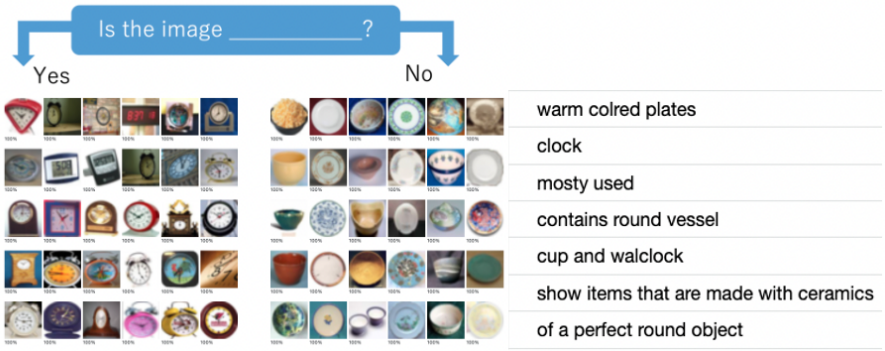

Figure 5.1: Example of Crowd sourcing results in NBDT-based Method

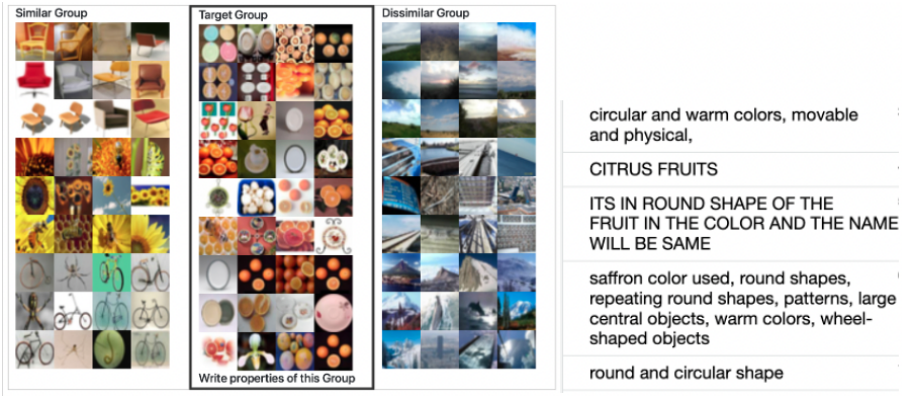

Figure 5.2: Example of Crowd sourcing results in KMeans-based Method

the images corresponding to the predicate, it seems that the predicate "green color in nature and used for many purpose" represents the green of grasses. Since there are predicates representing other greens such as snakes and pears, it can be assumed that this predicate is firing to the green leaves and stems of rose and tulip.

Although, grasping the concept that the predicate represents is quite difficult from the annotations obtained in this experiment. In the proposed method, the predicates are independent of the classification process. Therefore, the predicates seem to be missing some essential points or redundant for explanations, since they exclude the context.

Figure 5.3: A Neuron Cluster showing images of Trees

Figure 5.4: A Neuron Cluster showing images of Lawnmowers

Chapter 6

Conclusion

This thesis explored a hybrid crowd-AI approach to develop ML models associated with human-interpretable symbolic rules. The proposed method extracts subsets of data instances that activate neurons similarly from the black-box decision process of trained neural networks to enable human abductive reasoning. Crowd workers are asked to conduct abductive reasoning to provide semantics of the extracted data instance subsets in a natural language, which serve as predicates to explain the data instance subsets.

We conducted experiments using crowdsoucing platforms on 2 methods (the NBDT-based Method and KMeans-based Method) and evaluation tasks.

This thesis provided experimental results showing that the proposed approach can obtain interpretable and meaningful symbolic rules and explanations based on them.

The answer to the first research question, "Is it possible to identify the components of ML models that corresponds to predicates for symbolic rules?" is yes considering the experimental results. The second research question was about the method generating good symbolic rules. Evaluation of symbolic rules, in this case, is difficult, but from the results showing KMeans-based Method to extract abstract features, we conclude that KMeans-based Method performs better than NBDT-based Method in this task setting.

Since workers tend to give annotations of object names in images, there were difficulties in obtaining abstract properties from crowdsourcing while properties require deliberate thinking.

In Logic Model, the results of crowdsourcing showed that most of the logic generated is somewhat plausible. Although, the predicates seemed to lack the context of explanation such as the counterpart classes of the classification.

Our future work includes generating minimal set of rules and exploration of other frameworks to obtain context-aware annotations to the neural network elements.

This research was approved by IRB of the organization the author belong to.

Acknowledgements

I would like to express my very great appreciation to Professor Wakabayashi, my research supervisor for his patient guidance, enthusiastic encouragement and useful critiques of this research work. His willingness to give his time so generously has been greatly appreciated.

I would also like to thank Professor Matsubara, Professor Ito, and Professor Morishima, for their advice and assistance in this work, offering their knowledge and exacting attention to detail have been an inspiration and kept my work on track throughout my study.

References

[1] Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205. Association for Computational Linguistics, 2020.

[2] Zach Wood-Doughty, Isabel Cachola, and Mark Dredze. Model Distillation for Faithful Explanations of Medical Code Predictions. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pp. 412–425. Association for Computational Linguistics, 2022.

[3] Sarah Wiegreffe and Yuval Pinter. Attention is not not Explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 11–20. Association for Computational Linguistics, 2019.

[4] Omar Zaidan and Jason Eisner. Modeling Annotators: A Generative Approach to Learning from Annotator Rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 31–40. Association for Computational Linguistics, 2008.

[5] Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4932–4942. Association for Computational Linguistics, 2019.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. *arXiv e-prints*, p. arXiv:1706.03762, 2017.

[7] Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556. Association for Computational Linguistics, 2019.

[8] Cynthia Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, Vol. 1, pp. 206–215, 2019.

[9] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In Jennifer G. Dy and Andreas Krause, editors, *ICML*, Vol. 80 of *Proceedings of Machine Learning Research*, pp. 2673–2682. PMLR, 2018.

[10] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E. Gonzalez. NBDT: Neural-Backed Decision Trees. *arXiv e-prints*, p. arXiv:2004.00221, 2020.

[11] Sheikh Rabiul Islam, William Eberle, Sheikh Khaled Ghafoor, and Mohiuddin Ahmed. Explainable Artificial Intelligence Approaches: A Survey. *arXiv e-prints*, p. arXiv:2101.09429, 2021.

[12] Peter Cathcart Wason and Jonathan St B. T. Evans. Dual processes in reasoning? *Cognition*, Vol. 3, No. 2, pp. 141–154, 1974.

[13] Steven Sloman. The Empirical Case For Two Systems of Reasoning. *Psychological Bulletin*, Vol. 119, pp. 3–22, 1996.

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[15] Ingo Feinerer and Kurt Hornik. *wordnet: WordNet Interface*, 2020. R package version 0.1-15.

[16] Michael Berry and Gašper Tkačik. Clustering of Neural Activity: A Design Principle for Population Codes. *Frontiers in Computational Neuroscience*, Vol. 14, p. 10.3389/fncom.2020.00020, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, p. arXiv:1301.3781, 2013.

[19] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. pp. 3973–3983, 2019.