Teaching the Machine to Explain Itself using Domain Knowledge

Abstract

Machine Learning (ML) has been increasingly used to aid humans to make better and faster decisions. However, humans struggle to comprehend the rationale behind model predictions, hindering trust in algorithmic decision-making systems. Considerable research work on AI explainability attempts to win back trust in AI systems by developing explanation methods but there is still no major breakthrough. At the same time, popular explanation methods (*e.g.*, LIME and SHAP) produce explanations very hard to understand for the non data scientists humans-in-the loop. To address this, we present JOEL, a neural network-based framework to jointly learn a decision-making task and associated domain knowledge explanations. JOEL is tailored to human-in-the-loop domain experts that lack deep technical ML knowledge, providing high-level insights about the model's predictions that very much resemble the experts' own reasoning. Moreover, by collecting the domain feedback from a pool of certified experts (human teaching), we promote seamless and better quality explanations. Lastly, we resort to semantic mappings between legacy expert systems and domain taxonomies to automatically annotate a bootstrap training set, overcoming the absence of concept-based human annotations. We validate JOEL empirically on a real-world fraud detection dataset. We show that JOEL can generalize the explanations from the bootstrap dataset. Furthermore, obtained results indicate that human teaching is able to further improve the explanations prediction quality.

1 Introduction

Notwithstanding the benefits of Machine Learning (ML) commoditization in many areas, many state-of-the-art ML methods convey recondite information about their predictions. The inability to comprehend these outcomes hinders trust in these systems.

A new research field, dubbed Explainable Artificial Intelligence (XAI), focuses on improving the interpretability (or transparency) of these systems. Whilst the XAI literature is booming with explainer methods [1, 2, 3, 4, 5], the large majority focus on the data scientist persona, creating low-level explanations that are too difficult to grasp to other non-technical personas.[6, 7, 8]. This gap is witnessed in the recent impetus of feature-attribution based explainers [1, 2, 3]. These build explanations as lists of input features and associated feature scores[1]. These explanations are too specific and too technical. For this reason, explainer methods alike fail to address the information needs of *domain experts*, which are deeply involved in many human-AI cooperative systems [6, 8].

Instead, it is necessary to create explanations capable of conveying domain knowledge. This should reflect the semantic concepts related with the task of the the human-in-the-loop, encoding information about the domain experts underlying reasoning upon decision-making. Although recent lines of research have begun exploring "concept-based explanations" [7, 6, 8, 9] these are not applicable to decision-making tasks.

[1]The feature's contribution to a given ML prediction

Preprint. Under review.

In the face of the current research, we propose a novel XAI framework, JOEL, a self-explainable neural network that jointly learns semantic concepts and a decision associated with a predictive task [8, 10]. Jointly learning the decision task and associated explanations leads to more robust and authentic[2] explanations. We opt for encoding this ML interpretability in the architecture of a neural network (NN) model. This gives us the flexibility to incorporate domain knowledge and associate semantic concepts to the decision-making task [7, 6, 8, 9]. Moreover, these high-level concepts may encode external information that while not present in the data itself may be very useful.

We address this problem from a yet unexplored area in XAI, which is through human teaching. Our framework is particularly tailored for humans-in-the-loop that are extremely skilful in a specific domain but lack deep technical ML knowledge. The human-AI close proximity facilitates human teaching, where the model learns how to improve itself based on the teacher's (expert's) feedback.

Finally, the creation of explainers based on domain knowledge poses difficulties, especially, in the data collection [8]. This reflects in the low availability of datasets. To address the impracticalities of human-labeling[3], we use a semi-supervised approach based on semantic mappings between existing legacy rules system and domain taxonomies.

The summary of our contributions are as follows:

1. a novel **self-explainable neural network framework** to jointly learn decision (class) labels and associated domain concepts (section 2.1);
2. a **human teaching** process that uses the human experts in the loop to teach the explanation model how to improve explanations (section 2.2);
3. a **semi-supervised approach** for bootstrapping the proposed self-explainer model (section 2.3.

2 Methodology

In this section, we describe our framework for jointly learning a decision task and associated domain knowledge explanations. We assume the semantic concepts (used as explanations) will help the domain experts (end-users) reasoning throughout their decision-making process. Conversely, the framework leverages the domain experts to provide feedback about which concepts justify their decisions. Hence, it can continuously improve both its predictive accuracy and explainability. We coin this framework Jointly learned cOncept-based ExpLanations (JOEL).

2.1 Architecture

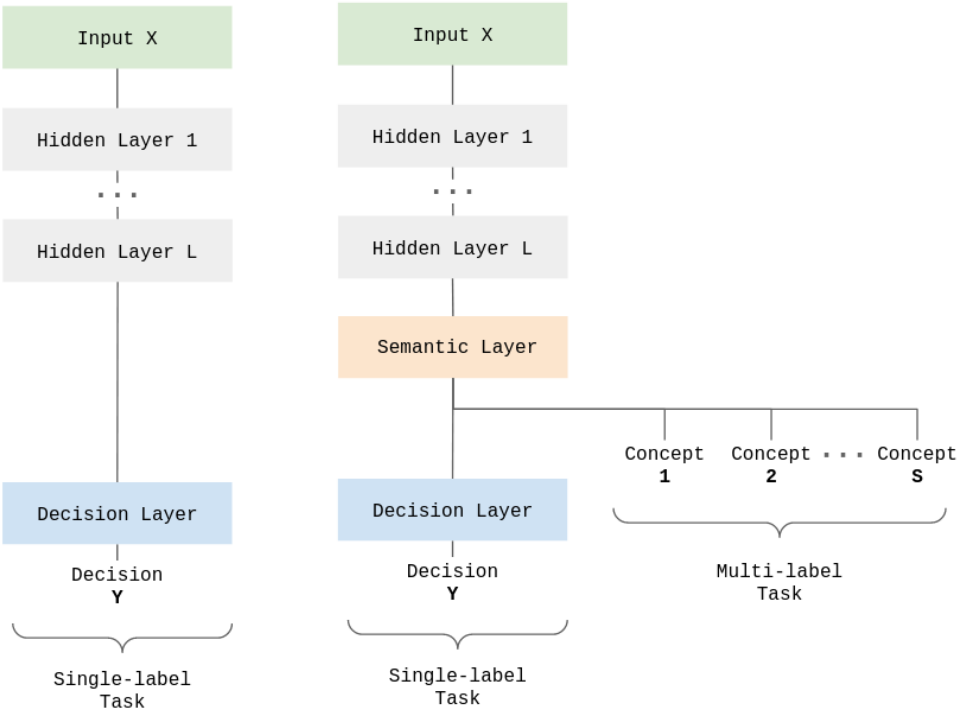
We frame the problem of associating both semantic concepts and decision labels as one of finding an *hypothesis* (learner), $h \in \mathcal{H}$, such that, for the same inputs, $x \in \mathcal{X}$, h is able to simultaneously satisfy $h : \mathcal{X} \rightarrow \mathcal{Y}$ and $h : \mathcal{X} \rightarrow \mathcal{S}$, where $\mathcal{S}$ is the set of semantic concepts, and $\mathcal{Y}$ is the set of decisions (or classes) of the predictive task.

Conceptually, JOEL comprises three building blocks: (1) a NN, (2) a semantic layer, and (3) a decision layer. NN's versatility facilitates the arrangement of these components in multiple ways. This work, however, delves into a hierarchical version (see figure 1b): whose blocks are chained sequentially, i.e., the outputs of the NN are fed as inputs to the semantic layer and its outputs are fed into the decision layer. In addition to the decision predictions given by the last layer, $\hat{Y}$, JOEL also outputs concept predictions, $\hat{S}$.

Figure 1 shows a visual comparison between vanilla NNs (figure 1a) and the proposed hierarchical method (figure 1b). Vanilla NNs seek to model single-label tasks, such as fraud detection, medical diagnosis, or predicting recividism. JOEL's hierarchical architecture can be seen as a multi-task materialization where the outputs of the $(L+1)^{th}$ layer are used both for a multi-label task (predict multiple semantic concepts) and a single-label task. By chaining the semantic and decision layers, we are able to encode external information about the domain which is not available in the feature data. This is particularly meaningful when the taxonomy of semantic concepts is intimately related to

[2] In this context, we refer to authenticity as being a true reason for a given ML outcome.

[3] Manually creating these datasets carries abnormal costs and time efforts.



(a) Vanilla Neural Network

(b) JOEL's hierarchical Network

Figure 1: Comparison between vanilla NN and JOEL's hierarchical model. Colors indicate layer type: (green) input vector; (grey) hidden layer; (orange) semantic layer; (blue) decision layer.

the decision task (*e.g.*, a fraud taxonomy of fraudulent patterns is deeply correlated with the fraud detection task). Therefore, learning to accurately predict the domain concepts is likely to also lead to better predictions and end-user decisions when it comes to the end-task.

Similarly to other NN architectures, our method learns through backward propagation of errors (*backprop*)[4] and some variant of the gradient descent method. Unlike other NNs models, the joint learning approach in JOEL's method attempts to minimize both a *decision loss*, $\mathcal{L}_D$, and a *semantic loss*, $\mathcal{L}_S$. Mathematically, given model's parameters, $\Theta = [\theta_1, \theta_2, ..., \theta_L, \theta_S, \theta_D]$, the outputs of the decision layer, $\delta_D(x, \Theta)$, and the outputs of semantic layer, $\delta_S(x, \Theta)$, the gradient with respect to the loss, $\mathcal{L}$ at the semantic layer is given by equation 1.

$$\nabla_{\delta_S} \mathcal{L}(x, y, s) = \nabla_{\delta_S} \mathcal{L}_D(\delta_D(x, \Theta), y) + \nabla_{\delta_S} \mathcal{L}_S(\delta_S(x, \Theta), s) \tag{1}$$

where:

$$\nabla_{\delta_S} \mathcal{L}_D(\delta_D(x, \Theta), y) = \frac{\partial \delta_D(x, \Theta)}{\partial \delta_S} . \nabla_{\delta_D} \mathcal{L}_D(\delta_D(x, \Theta), y)) \tag{2}$$

The decision of which loss functions to use within each task depends on the nature of the task. As the semantic layer corresponds to a multi-labeling task, we opt to use the *sigmoid* function and apply it to each individual entry of the output, before using it in the loss function. To find the mapping that simultaneously satisfies $h : \mathcal{X} \rightarrow \mathcal{Y}$ and $h : \mathcal{X} \rightarrow \mathcal{S}$ for a given input vector, $x \in \mathcal{X}$, we mutually minimize the (categorical) cross-entropy for both predictive tasks. For an input vector, $x \in \mathcal{X}$, a set of domain concepts, $s \in \mathcal{S}$, and decision labels, $y \in \mathcal{Y}$, we have that:

[4]The *backprop* algorithm makes use of local derivatives and of the *chain rule* (derivatives) to propagate the impact on the loss $\mathcal{L}$ with respect to each parameter involved in the computation.

$$\mathcal{L}_D(x, y) = -\sum_{i=1}^{|\mathcal{Y}|} y_i \log\left[\text{softmax}(\delta_D(x, \Theta)_i)\right] \tag{3}$$

$$\mathcal{L}_S(x, s) = -\sum_{i=1}^{|\mathcal{S}|} s_i \log\left[\text{sigmoid}(\delta_S(x, \Theta)_i)\right] \tag{4}$$

2.2 Human Teaching

Human teaching (or human-tuning) closes the human-AI feedback loop in algorithmic decision-making systems. It is often the case that interactions within these systems are uni-directional: a human (typically, a domain expert) interacts with an ML system to make a decision. However, while the ML system directly influences the human decision, the reverse is not verified, as most systems do not promptly adapt to the human behaviour. Instead, most systems are offline, and it is only after a certain period of time that a new model is trained and adapted to the collected knowledge.

JOEL tackles these limitations. It incorporates a human-teaching stage that continuously integrates expert feedback into the learning process. Figure 2 shows this mechanism in a illustrative example of a fraud detection task, where the main goal is to classify financial transactions as being fraudulent or not. After receiving a transaction, JOEL infers predictive scores for both the fraud label (decision) and the fraud patterns (semantic concepts). Using this information (explanation), a fraud analyst (domain expert) reviews the transaction and indicates whether she decided based on the evidence of any fraud patterns. Lastly, JOEL's learnable parameters get updated based on the expert's feedback.

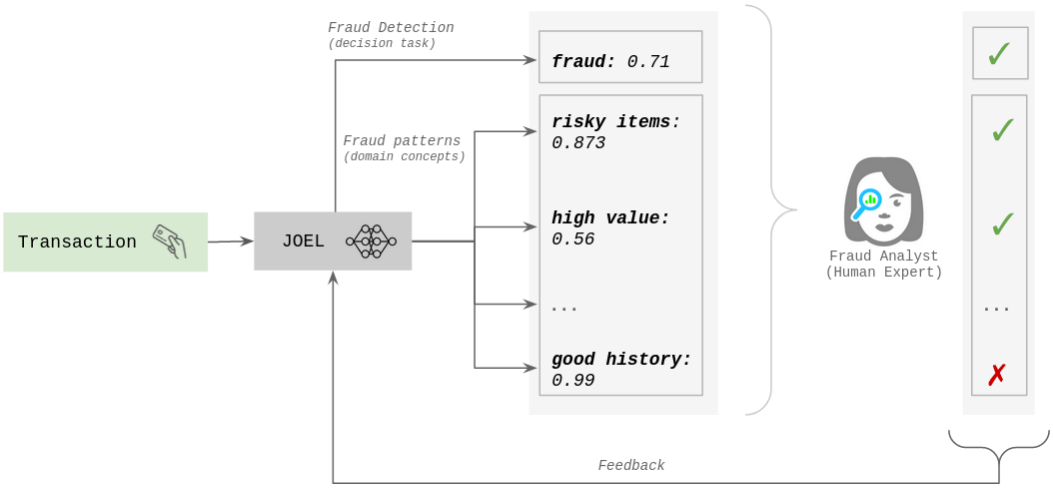


Figure 2: Example of JOEL's feedback loop (with human-teaching) in a fraud detection task.

In many real-world settings, human expertise aims to disambiguate inputs for which the model is very uncertain. In these cases, our method can exploit this short-term feedback to improve human-AI system performance. In fact, our method's hierarchical structure (with the semantic layer also acting as embeddings for the decision layer) is likely to encode additional information based on this feedback. Therefore, rapidly improving its predictive accuracy but also the quality of its explanations.

Prior to running the human teaching stage, we bootstrap JOEL using an initial concept-based annotated dataset. This may also include hyperparameter tuning. The resulting model gets deployed and JOEL enters in the human teaching stage, consistently interacting with the human-in-the-loop: providing explanations and collecting human feedback about the same explanations. Every b feedback instances, JOEL uses them to adjust its parameters through *backprop*.

In some situations, the human teaching may lead to quick model deterioration (*e.g.*lower human aptitude for the task). For this reason, we incorporate a quality control precondition for running this stage: to adhere to the pool of human teachers in the human-teaching stage, each human must meet a minimum accuracy-level (or other relevant qualitative measure).

2.3 Distant Supervision

Training neural networks requires large amounts of data in order to attain reasonably good performance [11]. Multi-label methods further exacerbate this data necessity. On top of that, difficulties associated with the collection and creation of concept-annotated datasets make NN explainers based on semantic concepts infeasible in many practical settings.

Realistically, many human-AI cooperative systems today, such as fraud detection, still depend on rule-based expert systems. Often times, these systems enclose semantic knowledge that can be associated with the concepts from the domain taxonomy. We propose to create these associations to automatically derive concepts for each datapoint without the need for human feedback. Note that this technique, known as *Distant Supervision* [12, 13, 14] that assumes a possible association between two entities. This approach still requires human effort to create the legacy systems-taxonomy mappings. However, this effort is negligible when compared with the effort of manually annotating a large dataset. As an example consider the fraud prevention domain involving a rule legacy system. Using the similarities between the domain knowledge conveyed in the rules, we create a *rule-concept mapping* to automatically associate rules to concepts in a fraud taxonomy. After proper validation of the mapping by a fraud expert, we use it to automatically label payment transactions in bulk. Specifically, consider a payment transaction X for which the legacy system triggered two rules, r_A and r_B. Suppose that according to a predefined *rule-concept mapping*, r_A maps to the "Suspicious Email" concept and r_B maps to "Suspicious IP", "Suspicious Customer", and "Suspicious Payment". Thus, by applying the Distant Supervision technique, transaction X is annotated with "Suspicious Email", "Suspicious IP", "Suspicious Customer", and "Suspicious Payment".

Although semi-supervised techniques improve scalability, it is important to still acquire a human-labeled dataset to empirically validate the results. As we will see in the ensuing section, we conduct an empirical validation of the proposed method using Distant Supervision to automatically label a dataset with 9.3 million instances, followed by a validation in a smaller human-labeled dataset with 1561 transactions.

3 Experimental Setup

We evaluate JOEL in a two-step approach. We first run a grid search over a "noisy dataset" to determine the best architectural and learning decisions, such as number of hidden layers, number of neurons at each layer, batch size, among others. In total, we train 70 models. Then, we fix the decisions for the best model and we run another grid search over the "ground truth dataset" (based on human labels) to determine the best configuration possible for a human teaching setup, that is, when the model is taught by a human expert in a given domain. Finally, throughout these experiments, we examine the different trade-offs emerging from jointly learning two distinctive tasks: fraud detection (single-label task) and creation of explanations based on fraud patterns (multi-label task). We coin the first stage *Distant Supervision* and the second *Human Teaching*.

Dataset: The dataset consists of approximately 9.3 millions of payment transactions of which only 2% represent fraudulent behaviors. For our experiment, we sequentially subdivide four subsets: train, validation, test, and production. These represent 50%, 16%, 7%, 28% of the dataset and exhibit 2.5%, 1.7%, 1.5%, 1.5%, respectively. While training JOEL, we use the validation set to apply early stopping and set the outputs' thresholds. Then, we perform model selection in the holdout test set, after which only the best model is evaluated in a production set. The production set is apart from the test set by two months, thus mimicking practical settings where there is a large time gap between model selection and deployment.

Domain Knowledge (or taxonomy): With the help of skilful fraud analysts, we created a fraud taxonomy of suspicious patterns. This type of domain knowledge guarantees the provided explanations will closely resemble their reasoning process throughout transactions reviewing. Semantically, the concepts refer to patterns involving specific information about the transaction. As an example consider the concept "suspicious billing shipping" which aims to guide analysts' attention to the information related with shipping or/and billing information and look for dubious aspects (*e.g.*mismatch between addresses, malformed address...).

Distant Supervision Dataset (or machine-labeled dataset): Departing from the initial single-label dataset (it only had fraud annotations) and from the fraud taxonomy, we used the Distant Supervision

approach to automatically annotate the dataset. We extracted the already existing information of a legacy rule system (that encoded high-level domain information) and mapped it into the fradulent patterns. Domain specialists supervisioned this mapping, which totals 300 associations. Table 1 summarizes how many rules were mapped for each concept. The result is a multi-label dataset: where each row is jointly associated with the fraud label (decision task) and fraud patterns (semantic concepts). Henceforth, given that these annotations are a proxies of the true associated concepts, we dub them "noisy labels". Indeed, a careful inspection of table 1 shows this technique to poorly cover non-fraudulent patterns (*e.g.*"Nothing Suspicious", "Good Customer History", "All Details Match"). This reveals its sensitivity to the contextual information, *i.e.*, to the availability of existing rules (the majority focusing in negative behaviors) and of the concepts in the taxonomy.

Concept	#Rules	Concept	#Rules
Suspicious Billing Shipping	86	Suspicious IP	24
Suspicious Customer	58	Suspicious Device	22
Suspicious Payment	53	Nothing Suspicious	19
Suspicious Items	48	Good Customer History	11
High Speed Ordering	31	Suspicious Delivery	5
Suspicious Email	27	All Details Match	3

Table 1: Number of mapped rules for each concept.

Human Labeled Dataset: A robust validation of our method should not depend exclusively on machine labeled datasets ("noisy datasets"), for they are not guaranteed to reflect the real data. Thus, we ran a human-labeling campaign involving three fraud experts, where we mimic their original reviewing process. After each transaction review, the system prompts the analysts about the fraud patterns (or legitimate patterns) perceived during the transaction appraisal. Analysts are able to select the concepts from the pool of concepts determined in the fraud taxonomy. This human effort yielded 1561 manually labeled transactions. Henceforth, we dub these labels as "ground truth" labels.

Evaluation Metrics: JOEL comprises two distinct tasks: a single-label task focused on discerning fraudulent from non-fraudulent transactions, and a multi-label task focused on producing a complete and informative explanation for the former. Following existing business constraints, we set out to evaluate the first task in terms of Recall at 3% False Positive Rate (FPR). The evaluation results (after training and validation) require an aggregate measure of JOEL's generalization at the multi-label task in holdout datasets. We use the mean Area Under the Curve (AUC) over all labels (predicted concepts) because AUC gives a general idea about the overall performance for a single label.

Hyperparameter Optimization best model criteria: From all the evaluated hyperparameter configurations, we select the one with highest performance at the fraud detection task. This decision guarantees business constraints are met regardless of the explanations. In case of ties, we discern the best model based on the performance at the second task - explanations accuracy.

Human Teaching (or human tuning): We conducted a second experiment using the human-teaching stage, the human-labeled dataset, and the best architectural decisions as determined in the first model selection stage. We carried out a small experiment to obtain proxy measures of the fraud analysts' accuracy (prior to being shown explanations). All three analysts satisfied the minimum accuracy level set for this experiment (they obtained 0.636, 0.612, and 0.597 of accuracy), and were, therefore, integrated in the teachers pool of the human-teaching stage. Afterwards, we split the human-labeled dataset into training, validation, and test sets. The first 800 transactions are used for training and the ensuing 200 are used for early stopping, threshold calibration, and model selection. Then, we re-train the best performing model using both the training and validation sets. We estimate its generalization performance in an holdout test set composed of 561 transactions.

4 Results and Discussion

We tested 70 differently configured JOEL models. Given the business constraints, we select the best model in terms of fraud recall at 3% FPR, achieving 0.2281 Recall on the test set. Table 2 summarizes the experiment results. The first column refers to results of the selected model on production dataset. The second column ("AUC No tuning") refers to the results of pre-trained model

with Distant Supervision on manually labeled dataset (the labels are provided by human experts), without tweaking any hyperparameter or its structure. Finally, the last column shows the results of applying the human tuning, *i.e.*, update the pre-trained model's hyperparameters with human experts' feedback.

Concept	AUC Distant Supervision	AUC No tuning	AUC With tuning
Suspicious Items	0.8499	0.2911	0.7119
Suspicious IP	0.9190	0.4760	0.7215
Suspicious billing shipping	0.8921	0.5943	0.7531
Suspicious Delivery	0.6467	0.5962	0.6368
Suspicious Email	0.9376	0.6231	0.7222
Suspicious Payment	0.9156	0.6971	0.7170
Suspicious Device	0.8862	0.7348	0.7799
High speed ordering	0.9468	0.7365	0.6874
Suspicious Customer	0.9401	0.4621	0.3640
All details match	0.6470	0.3496	0.3976
Nothing suspicious	0.7169	0.4151	0.3425
Good customer history	0.8370	0.4250	0.3266
Mean	**0.8446**	**0.5393**	**0.6125**

Table 2: Model performance on (1) production dataset, trained with Distant Supervision; (2) on human-labeled test set, without human tuning; (3) on human-labeled test set, with human tuning.

Distant Supervision results: Since we are using Distant Supervision to create "noisy labels", the quality of prediction depends on the mapped rules for each concept. For instance, the concept "All details match" has only three mapped rules, meaning that the model will struggle to catch these label patterns. In fact, this lack of representation (reduced number of mapped rules) is reflected in the results. Analysing table 2, we observe that the selected model only achieves 0.6470 AUC in the concept "All details match". We hypothesize this is due this concept only has three mapped rules. For the fraud concepts, in general, the model shows high AUC values. However, the model performs poorly for the "Suspicious Delivery" for the same reason as the ones previously discussed. This concept has only 5 mapped rules, meaning that the model's representation for that label is not enough to achieve high performance. Finally, we check the selected model performance on the manual labeled dataset, which labels were collected in annotation campaign. Until this stage, we use a "noise labels" crafted with Distant Supervision approach. But in real world setting, the self-explainable model will consume the human expert feedback. Thus, we evaluate the selected model performance on dataset with labels provided by human experts (which are fraud analysts in our use-case). A second column on table 2 shows the resulted AUCs for each label. Analysing this results we can conclude that the main limitation of the Distant Supervision approach is the quality of the mapping. This is, if semantic concepts do not have enough mapped rules, it could result in poor performance (as we observe the concepts "Suspicious Delivery" and "All details match", for which the model presents poor performance). The number of satisfactory mapped rules is task dependent, *e.g.*, in our use-case, semantic concepts with around 30 mapped rules gives satisfactory results.

Human teaching results: We started with the evaluation of the selected model on manually labeled test set, containing 561 instances, without tweaking any hyperparameter or its structure (second column in table 2). Then, we use the same model but apply human tuning, *i.e.*, we use the real human experts feedback to update the model's parameters (last column in table 2). For these results, we use a learning rate of 0.05 (the model was originally trained with 0.0001 of learning rate), meaning that the model's weights will be updated more aggressively. Analysing the third column in table 2), we see, for instance, an improvement for the "Suspicious Delivery" concept, for which the Distant Supervision approach had poor performance due the lack of mapped rules. The same applies to "All details match", but the observed improvement is small. However we can observe that human tuning process reduces the performance for "High speed ordering". One possible cause of this result may be the inconsistency of the human experts' mental representation for each concept. This is, if each decision maker have a different notion of the domain concepts, the collected feedback will incorporate this inconsistency, worsening the quality of labels.

Overall, we can observe that (1) the pre-trained model with Distant Supervision approach is able to generalize the explanations, and (2) the human teaching improves the model's capability producing explanation, this is, improving the prediction quality on the concepts that the pre-trained model struggle to predict.

5 Related Work

The use of human-friendly semantic concepts is becoming more popular, specifically, in the computer vision domain. For instance, Kim *et al.*[7] proposed "Testing with Concept Activation Vectors" (TCAV), a concept-based explainer, which weighs the importance of a user-defined concept to a classification result (*e.g.*, how sensitive to the prediction of "bird" is to the presence of concepts "feathers"). Recently, this method has also been applied to the natural language domain [15]. Also in the computer vision domain, Ghorbani *et al.*[6] propose "Automated Concept-Based Explanation" (ACE), that extracts concepts directly from the input data without any external source. It first obtains different embeddings for different image segments and then applies a clustering algorithm in the embeddings space to identify the concepts.

In another perspective, Melis and Jaakkola [8] include the interpretability straight into the ML model architecture (self-explainer concept-based model). They use autoencoders to generate the basis concepts rather than using low-level features. In another vein, Panigutti *et al.*[9] propose the use of domain knowledge to produce more comprehensive explanations in medical diagnosis tasks. This domain information is used to guide the explanation generation process of a low-level explainer, LIME [1].

All in all, most concept-based methods in the literature are specific to computer vision. Indeed, there is currently no research work in what it comes to high-level concept-based explanations for decision-making tasks. While *DoctorXAI* [9] uses concepts to create explanations for a decision-making setting, the explanation itself does not convey any domain knowledge about the decision but rather about the feature space. Consequently, it fails to address information needs of non data scientist persona.

6 Conclusion and Future Work

In this work, we propose JOEL, an NN-based framework for jointly learning a decision task and associated domain knowledge explanations. Simultaneously, our framework incorporates the decision-makers feedback regarding the concepts that justify their decisions. Initial experiments show that using human tuning can lead to significant gains in predictive accuracy and explainability. Moreover, we show that JOEL is able to generalize from a dataset automatically populated (with Distant Supervision). Future directions include robustly assessing the framework's true impact and utility for the end-users, exploring different NN variants (*e.g.*, architectural, loss functions), and exploring incremental learning to continuously adapt to new knowledge.

References

[1] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-August-2016:1135–1144, 2016.

[2] Scott M. Lundberg and Su In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017-Decem(Section 2):4766–4775, 2017.

[3] Gregory Plumb. Model Agnostic Supervised Local Explanations. (NeurIPS):1–10, 2018.

[4] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1527–1535, 2018.

[5] Muhammad Rehman Zafar. DLIME : A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. 2019.

[6] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards Automatic Concept-based Explanations. 2019.

[7] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *35th International Conference on Machine Learning, ICML 2018*, 6:4186–4195, 2018.

[8] David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pages 7775–7784, 2018.

[9] Cecilia Panigutti, Alan Perotti, and Dino Pedreschi. Doctor xai: an ontology-based approach to black-box sequential data classification explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 629–639, 2020.

[10] Daniel C Elton. Self-explaining ai as an alternative to interpretable ai. *arXiv preprint arXiv:2002.05149*, 2020.

[11] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[12] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, 2009.

[13] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[14] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762, 2015.

[15] Ben Hutchinson, K J Pittl, and Margaret Mitchell. Interpreting Social Respect : A Normative Lens for ML Models. 2019.