Ontology-based Interpretable Machine Learning with Learnable Anchors

Blind Review

Abstract

In this paper, we introduce a novel interpreting framework that learns an interpretable model based on an ontology-based sampling technique to explain agnostic prediction models. Different from existing approaches, our algorithm considers contextual correlation among words, described in domain knowledge ontologies, to generate semantic explanations. To narrow down the search space for explanations, which is a major problem of long and complicated text data, we design a learnable anchor algorithm, to better extract explanations locally. A set of regulations is further introduced, regarding combining learned interpretable representations with anchors to generate comprehensible semantic explanations. An extensive experiment conducted on two real-world datasets shows that our approach generates more precise and insightful explanations compared with baseline approaches.

1 Introduction

In critical scenarios, such as clinical practices, having the ability to interpret machine learning (**ML**) model outcomes is significant to reduce the error rate and improve the trustworthiness of ML-based systems [1–3]. To achieve this, typical approaches, called *Interpretable ML* (**IML**), are to train additional interpretable models to generate explanations, which usually are crucial input features (i.e., important terms, in text analysis [2, 4] or super-pixels, in image processing [5, 6]), for each predicted outcome. However, most of existing IML algorithms usually treat input features independently, without considering their semantic correlations, especially in natural language processing. As a result, generated explanations commonly are fragmented, incomplete, and difficult to understand.

Addressing this problem is a non-trivial task, since: **(1)** It is difficult to capture semantic correlations among features, which can be contextually rich and dynamic; **(2)** There is still a lack of scientific study on how to integrate semantic correlations among features into IML to generate *good* explanations, which are concise, complete, and easy to understand; and **(3)** The search space for good explanations can be large and complicated, given noisy and poor data. That results in a limited understanding of how to define good explanations, and of how to effectively and efficiently identify them.

In literature, ontology, which encodes domain knowledge, can be used to capture semantic correlations among input features, such as entities, terms, phrases, concepts, etc. [7, 8]. However, there is an unexplored gap regarding how to guide the learning process of an IML model based on ontology. Straightforwardly matching ontology and explaining data points, by randomly sampling co-occurring terms and concepts in conventional approaches, e.g., LIME [4], cannot generate good explanations, since contextual information in the data is usually very rich compared with the ontology. In addition, building an ontology that can sufficiently capture contextual information in the data is very expensive. Meanwhile, the traditional concept of anchor text [9] can be used to narrow down the search space, by pinpointing generally important text. However, the approach was not designed for each single and independent data point, i.e., at local level.

Our contributions. To synergistically overcome these challenging issues, we propose a novel *Ontology-based IML* (**OnML**), to generate good explanations. In our approach, text data is first classified by a prediction model. Then, we learn a linear interpretable model by approximating the

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

predictive model based on data sampled around the prediction outcome. In constrast to existing approaches, in our sampling, correlated words and concepts are extracted and sampled together.

Second, we introduce a new concept of *learnable anchor texts*, to narrow down the search space for explanations. A learnable anchor text essentially is a contextual phrase, which can be expanded by adding nearby terms, without affecting the impact of the 'anchor' to the model outcome. For instance, anchors can be started with a term that has negative meanings, e.g., "no," or "not," and then expanded to neighboring texts to effectively capture negative experiences and events, e.g., "not get any help."

Finally, we introduce a set of regulations to combine correlated words, terms, and concepts, learned anchor texts, and triplexes extracted from the text, to generate semantic explanations. Each generated explanation will be assigned an importance score, measuring its impact upon the model outcome. To our knowledge, our approach establishes the first connection among *domain knowledge ontology*, *IML*, and *learnable anchor texts*. Such a mechanism will greatly extend the applicability of machine learning, by fortifying the models in both interpretability and trustworthiness.

Extensive experiments conducted on two real-word datasets in critical applications, including drug abuse in the Twitter-sphere [10] and consumer complaint analysis[1], to quantitatively and qualitatively evaluate our OnML approach, show that our algorithm generates concise, complete, and easy-to-understand explanations, compared with existing mechanisms.

2 Background and Problem Definition

In this section, we revisit IML, ontology-based approaches, and information extraction algorithms, which are often used to generate explanations. We further discuss the relation to previous frameworks and introduce our problem definition.

Let D be a database that consists of N samples. A classifier outputs class scores $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ that maps inputs $x \in \mathbb{R}^d$ to a vector of scores $f(x) = \{f_1(x), f_2(x), \ldots, f_K(x)\}$ s.t. $\forall k \in [1, K]$: $f_k(x) \in [0, 1]$ and $\sum_{k=1}^{K} f_k(x) = 1$. The highest-score class is selected as the predicted label for x.

Interpretable learning. Let us briefly revisit interpretable learning techniques, starting with the definition of *interpretable model*. Given an interpretable model g, which provides insights and qualitative understanding about the model f given an input x, interpretability is closely related to the ability of humans to understand the model; therefore, g must have a low complexity to generate understandable explanations. In practice, the complexity usually is measured by a function $T(g)$, which basically is the number of important words [2, 4], based upon that users can handle to evaluate the generated explanation. Another essential criterion for interpretability is *local fidelity*, which implies the ability of g to approximate the model f in a vicinity of the input x.

Let z be a sample of x, where z is generated by randomly selecting or removing features/words in x. $\phi_x(z)$ is a similarity function to measure the proximity between x and z. Given a d'-dimensional binary vector $z' \in \{0, 1\}^{d'}$, $z'_i = 1$ indicates that the feature i-th ($\in x$) is present in z, and vice-versa.

To achieve the interpretability and local fidelity, [4] minimizes a loss function $L(f, g, \phi_x)$, with a low complexity $T(g)$, by solving the following problem:

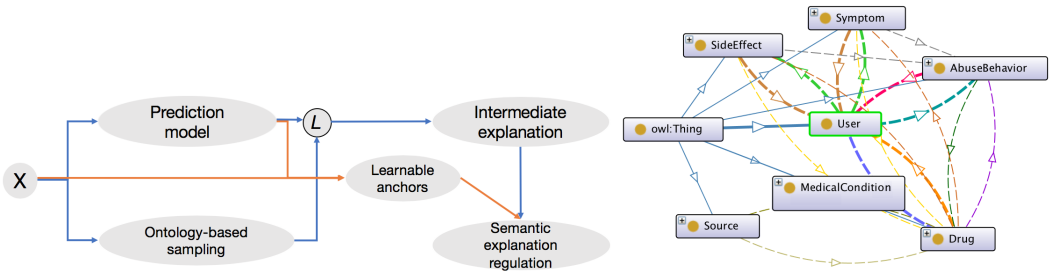
$$g^* = \arg\min_g L(f, g, \phi_x) + T(g) \tag{1}$$

where $L(f, g, \phi_x) = \sum_z \phi_x(z)(f(z) - g(z'))^2$, $\phi_x(z) = \exp(-D(x, z)^2/\sigma^2)$ is an exponential kernel with $D(x, z)$ is a distance function (e.g., cosine distance) with a width σ, and $g(z') = w_g z'$.

To obtain the data z for learning g in Eq. 1, sampling approaches are employed. In LIME [4], the authors draw nonzero elements of the original data x uniformly at random. Similar to this approach, a number of works follow [11–13]. Apart from the randomization, model decomposition is another line of learning g [1, 2], in which the prediction $f(x)$ is decomposed on individual features to learn the effect of each feature on the outcome. These existing randomization and decomposition approaches treat features independently; therefore, they cannot capture correlations among features. This may not be practical in real-world scenarios, since features usually are highly and semantically correlated.

Ontology-based approaches. To capture semantic correlations among input features, ontology, which studies related concepts and their relations, can be applied. Ontology is used in [14] to filter

[1] https://www.consumerfinance.gov/data-research/consumer-complaints/



(a) A flow chart of the OnML approach.

(b) Drug abuse ontology.

Figure 1: OnML approach and Drug abuse ontology example.

and rank concepts from selected data points to conduct informative explanations. The explanations are derived in ontological forms. For example, the information, "a 30 year-old individual, with an operation occurred in 1964," can be conveyed by the representation, "*TheSilentGeneration* ⊓ *OperationIn1960s.*" (TheSilentGeneration denotes people in the age range of 30-39.)

In [8], authors use ontology to learn an understandable decision tree, which is an approximation of a neural network classifier. Explanations are in a non-syntactic form, and they are not designed for a single and independent data point. Different from [8], we aim at generating semantic explanations for each input x. In this paper, generating semantic explanations is defined as a process of mapping a text to a representation of important information in *a syntactic* and *understandable* form.

Information extraction. Apart from interpretable learning, information extraction (IE) is another direction to capture contextual information semantically. The first Open IE algorithm is TextRunner [15], which identifies arbitrary relation phrases in English sentences by automatically labeling data using heuristics for training the extractor. Following TextRunner, a number of Open IE frameworks [16–18] were introduced. Unfortunately, these approaches ignore the context. OLLIE [19] includes contextual information; and extracts relations mediated by nouns, adjectives, and verbs; and outputs triplexes (subject, predicate, and object). Compared to Open IE approaches, our algorithm mainly focuses on generating semantic explanations associated with the prediction label.

3 Ontology-based Interpretable Machine Learning with Learnable Anchors

In this section, we formally present our OnML with learnable anchors (Fig. 1a). Alg. 1 presents the main steps of our approach. Given an input x, an ontology $\mathcal{O}$, and a set of all concepts $\mathcal{C}$ in $\mathcal{O}$, we first present the notion of *ontology-based tuples* (Line 3), which will be used in an *ontology-based sampling technique* to learn the interpretable model g (Lines 4-6). Next, we learn potential anchor texts using the input x and the model $f(x)$ (Line 7). Meanwhile, OLIIE [19] is applied to extract triplexes, which have high confident scores, in x (Line 8). After learning g, learning anchor texts $\mathcal{A}$, and extracting triplexes $\mathcal{T}$, we introduce a set of regulations to combine them together to generate semantic explanations (Line 9). Let us first present the notion of ontology-based tuples as follows.

3.1 Ontology-based tuples

Given concepts A and B, $A \mapsto B$ is used to indicate that A has a directed connection to B. In considerably correlated domains, such as text data, it is observed that 1) surrounding words affect the contextual information of the certain word; and 2) different sentences usually have different contextual information. To encode the observations, we introduce a *contextual constraint*, as follows:

$$\lambda_{x_k}(x_l) \leq \gamma \tag{2}$$

where x_k and x_l are two words in x, γ is a predefined threshold, and $\lambda_{x_k}(x_l)$ measures the distance between the positions of x_k and x_l in x. In text data, if x_k and x_l belong to two sentences in x, they are considered to be violating the contextual constraint.

Definition 1. *Ontology-based tuple. Given x_k and x_l in x, (x_k, x_l) is called an ontology-based tuple, if and only if: (1) $\exists A, B \in \mathcal{C}$ s.t. $x_k \in A$ and $x_l \in B$; (2) $A \mapsto B$; and (3) $\lambda_{x_k}(x_l) \leq \gamma$.*

Algorithm 1 OnML approach

1: **Input:** Input x; ontology $\mathcal{O}$, and user-predefined anchor $\mathcal{A}_0$
2: Classify x by a prediction model $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$
3: Find ontology-based tuples (x_i, x_j) in x based on concepts and relations in $\mathcal{O}$
4: Sample x, based on ontology-based tuples found by our sampling technique to obtain sampled data $z \in \mathcal{Z}$
5: Generate vectors of predictive scores $f(z)$ with $z \in \mathcal{Z}$
6: Learn an interpretable model g based on $f(z)$ and $g(z')$ by Eq. 1
7: Learn anchor text by our anchor learning algorithm (as shown in Alg. 2)
8: Extract triplexes in x using an existing Open information technique
9: Combine correlated terms, learned anchors, and extracted triplexes by our proposed regulations
10: **Output:** Semantic explanation $\mathcal{E}$

Since ontology has directed connections among its concepts, ontology-based tuples are asymmetric, i.e., (x_k, x_l) and (x_l, x_k) are different. For the sake of clarity without affecting the generality, we use a drug abuse ontology as an example (Fig. 1b). Given the drug abuse ontology and x that is "Smoke typically causes addiction.", "Smoke," and "addiction" are in "Abuse Behavior," and "Side Effect" concepts. Following the aforementioned conditions (with $\gamma = 4$), an ontology-based tuple (Smoke, addiction) is found in x.

3.2 Ontology-based sampling technique

To integrate ontology-based tuples into learning g, we introduce a novel ontology-based sampling technique. To learn the local behavior of f in its vicinity (Eq. 1), we approximate $L(f, g, \phi_x)$ by drawing samples based on x, with the proximity indicated by $\phi(x)$. A sample z can be sampled as:

$$z = \Big(\cup_{x_i \in x, i \neq k, i \neq l} \mathcal{R}(x_i) \Big) \cup \mathcal{R}(\{x_k, x_l\}) \tag{3}$$

where $\mathcal{R}(\alpha)$ is a probability randomly drawn for each word α in x. If $\mathcal{R}(\alpha)$ is greater than a pre-defined threshold, then α will be included in z.

In our sampling process, x_k and x_l, i.e., an ontology-based tuple, are sampled together as a single element. This aims to integrate the semantic correlation between x_k and x_l, captured in an ontology-based tuple into the sampling process. In fact, we are sampling the semantic correlation, but not sampling each word/feature x_k or x_l independently. This enables us to measure the impact of this semantic correlation on $f(x)$. In addition, words, which are not in any ontology-based tuple, are sampled independently. After sampling x (as shown in Eq. 3), we obtain the dataset $\mathcal{Z}$ that consists of sampled data points z associated with its label $f(z)$. $\mathcal{Z}$ is used to learn g^* by solving Eq. 1.

3.3 Learnable anchor text

Before presenting our anchor text mechanism, we introduce an *importance score* notion, which will be used in choosing the best anchor and calculating the importance of generated explanations.

Importance score. To get insight into the importance of generated explanations and their impact upon the model outcome, we calculate an importance score (IC) for each explanation. Intuitively, the higher importance score, the more important the explanation is. The IC is calculated as:

$$IC(r) = \bar{c}_r \Big(f(x) - f(x/r) \Big) \tag{4}$$

where x/r is the original text x excluding words in the explanation r and $\bar{c}_r$ is average coefficients of g^* associated with all words in r.

Anchor text learning mechanism. It is challenging to work with long and poor data, e.g., large number of words, or misspelled text, since the contextual information is generally rich and complicated. Building an ontology to adequately represent such data is expensive, and insufficient in many cases. That results in a large undercovered search space for explanations. To address this problem, we introduce a learnable anchor mechanism to narrow down the search space.

The learning anchor technique is presented in Alg. 2. The anchor is initialized with an empty set (Line 2). A set of user-predefined anchors $\mathcal{A}_0$ is provided, which consists of starting-words that are further expanded by incrementally adding words to the end of the sentence. Then, the importance

Algorithm 2 Anchors learning algorithm

Input: Input x; prediction model f; number of sentences in x, denoted as M; user-predefined anchors $\mathcal{A}_0$
$\mathcal{A} \leftarrow \emptyset$ ($\mathcal{A}$: set of anchors for x)
for $i \in M$ **do**
  if any $\mathcal{A}_0$ appears in the sentence i **then**
    Denote $D_{\mathcal{A}}$ as a set of ordered words appearing after $\mathcal{A}_0$ in the sentence i in x
    $\mathcal{A}_n \leftarrow \emptyset$ ($\mathcal{A}_n$ is a set of candidate anchors)
    $\mathcal{F}_n \leftarrow \emptyset$ ($\mathcal{F}_n$ is a set of importance scores, associated with each candidate anchor)
    for $x_j \in D_{\mathcal{A}}$ **do**
      $\mathcal{A}_n \leftarrow \mathcal{A}_0 \cup x_j$; $\mathcal{A}_0 \leftarrow \mathcal{A}_n$; $\mathcal{F}_n \leftarrow \mathcal{F}_n \cup IC(\mathcal{A}_n)$
      Choose the best anchor for sentence i: $\mathcal{A}_i = \arg\max_{\mathcal{A}_n} \mathcal{F}_n$
  else
    $\mathcal{A}_i \leftarrow \emptyset$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}_i$
Output: $\mathcal{A}$

score of each candidate anchor is calculated, following Eq. 4. The top-1 anchor $\mathcal{A}$, which has the highest important score, for each sentence are then chosen from candidate anchors.

3.4 Generating semantic explanations

We further apply OLLIE to extract triplexes $\mathcal{T}$ (subject, predicate, and object) to identify the syntactic structure in a specific sentence, which can be used to shape our explanations in a readable form. To generate semantic explanations $\mathcal{E}$, we introduce a set of regulations to combine g^*, $\mathcal{A}$, and $\mathcal{T}$ together:

- 1) $\mathcal{E} \subseteq D_x$ with D_x is a set of all words in x.
- 2) If there is no ontology-based tuple found, $\mathcal{E}$ will only consist of the learned anchor texts.
- 3) In a sentence, if there are two or more ontology-based tuples, we introduce three rules to merge them together, as follows:
  - *Simplification rule:*
    - Given (x_k, x_l) and (x_k, x_m), if x_l and x_m are in the same concept, then the ontology-based explanation is $\{x_k\ x_l$ *and/or* $x_m\}$.
    - Given (x_k, x_m) and (x_l, x_m), if x_k and x_l are in the same concept, then the ontology-based explanation is $\{x_k$ *and/or* $x_l\ x_m\}$.
    - Given (x_k, x_l) and (x_l, x_m), then the ontology-based explanation is $\{x_k\ x_l\ x_m\}$.
  - *Union rule*: Given (x_k, x_l), (x_k, x_m), (x_l, x_m), and $\{x_k, x_l, x_m\}$, the ontology-based explanation is $\{x_k\ x_l\ x_m\}$.
  - *Adding Causal words rule*: Semantic explanation can be in the form of a causal relation. Thus, if a causal word, e.g., "because," "since," "therefore," "as," "so," "while," "whereas," "thus," "thereby," "meanwhile, "however," "hence," "otherwise," "consequently," "when," "whenever" appears between any words in ontology-based tuples/explanations, we add the word to the explanation, following its position in x.
  - *Combining with anchor texts $\mathcal{A}$ and triplexes $\mathcal{T}$*: After having the ontology-based explanation, we combine them with $\mathcal{A}$ and $\mathcal{T}$ based on their positions in x. Then, the *semantic explanation* is generated from the beginning towards the end of all positions of words found in the ontology-based explanations, $\mathcal{A}$, and $\mathcal{T}$. For example, in the sentences, "We were filling out all the forms in the application. However, there is a letter in saying loss mitigation application denied for not sending information to us.", after the learning process, we obtain: 1) ontology-based explanation is (loss, application); 2) anchor text is "not sending information"; and 3) triple is "a letter; denied; mitigation application". The semantic explanation $\mathcal{E}$ is "a letter in saying loss mitigation application denied for not sending information."
- 4) If different ontology-based tuples are in different sentences in x, due to the contextual constraint in Eq. 2, the explanation for each sentence follows the 3^{rd} regulation.

Figure 2: Visualization of drug abuse (*top*) and consumer complaint (*bottom*) experiments.

4 Experiments

We have conducted extensive experiments on two real-world datasets, including drug abuse (Twittersphere [10]) and consumer complaint analysis from Consumer Financial Protection Bureau[1].

4.1 Baseline Approaches

Our OnML approach is evaluated in comparison with traditional approaches: **(1)** an interpretable model-agnostic explanation, i.e., LIME [4]; and **(2)** information extraction, i.e., OLLIE [19]. LIME is one of the state-of-the-art and well-applied approaches in interpretable model-agnostic explanation, in which the predictions of any model are explained in a local region near the sample being explained. There are other algorithms sharing the same spirit as LIME, in terms of generating explanations [6, 20–25]. For the sake of clarity, we use LIME as a representative baseline regarding this line of research. Meanwhile, OLLIE is a high-precision information extraction method that learns open pattern templates over these training samples. In LIME and OLLIE, domain knowledge is not used.

4.2 Datasets and Domain Ontologies

To validate the proposed method, we have developed two different domain ontologies, which are drug abuse ontology (Fig. 1b) and consumer complaint ontology (**Appendix**).

Drug abuse dataset. We will use the term “drug abuse" in the wider sense, including abuse and use of Schedule 1 drugs that are illegal and have no medical use; and misuse of Schedule 2 drugs (e.g., Oxycodone), which have medical uses, yet have a potential for severe addiction, and which can be life-threatening [26]. Main concepts of the drug abuse ontology (**DrugAO**) (Fig. 1b) capture correlation among key concepts, including abuse behaviors, drug types, drug sources, drug users, symptoms, side effects, and medical condition when using drug. In total, we have 506 drug-abuse related terms (including slang terms and street names), and 18 relations covered in our DrugAO.

The drug abuse dataset consists of $9,700$ tweets labelled by [10] with a high agreement score. Among them, $3,043$ tweets are drug abuse tweets, labeled *positive* and the rest are non drug abuse tweets, labeled *negative*. The statistical analysis is in Table 1.

Consumer complaint dataset. A consumer complaint is defined, here, as a complaint about a range of consumer financial products and services, sent to companies for response. Main concepts of the consumer complaint ontology (**ConsO**) encode the relation among different entities related to consumer complaint: for instance, who is complaining; what happened to make consumers unhappy and then complaint; etc. These ontologies were semi-manually generated, in which concepts were grouped and collected from the dataset by the K-means clustering algorithm [27], and then judged by humans to reduce redundant or inappropriate concepts. In total, we have 572 finance and product-related terms and 9 relations covered in our ontology. The consumer complaint dataset consists of $13,965$ mortgage-related complaints, labeled with 16 categories. These complaints were used for learning a model to predict the issue regarding each complaint.

4.3 Experimental Settings

Our experiment focuses on validating whether: **(1)** Our OnML approach can be applied on different agnostic predictive models; and **(2)** Our approach can generate better explanations, compared with baseline approaches, in both quantitative and qualitative measures.

To achieve our goal, we carry out our evaluation through three approaches. First, by employing SVM and LSTM, we aim to illustrate that OnML works well with different agnostic predictive models. Second, we leverage the word deleting approach [28] as an quantitative evaluation. Third, we apply qualitative evaluation with Amazon Mechanical Turk (**AMT**).

SVM and LSTM models. In the drug abuse dataset, tweets were vectorized by TF-IDF [29] and then classified by SVM. We achieved 83.6% accuracy. Tweets are short (Table 1); i.e., the average and maximum numbers of words in a tweet are 12 and 37. Therefore, it is not necessary to apply the anchor learning algorithm, which is designed to tighten down the search space for long text data.

In the consumer complaint dataset, Word2vec [30] is applied for feature vectorization. Then, a Long short-term memory (LSTM) [31] is trained as a prediction model. In LSTM, we used an embedding input layer with $d = 300$, one hidden layer of 64 numbers of hidden neurons, and a softmax output layer with 16 outputs. An efficient ADAM [32] optimization algorithm with learning rate 0.01 was employed to train LSTM. For the prediction model, we achieved 53% accuracy. We registered that this is a reliable performance, since the 16 categories are densely correlated resulting in a lower prediction accuracy [33]. Another reason for the low accuracy is the limited number of samples. We will collect more data in the future.

For sufficiently learning anchors in consumer complaints, we have chosen a set of negative terms as user-predefined anchors $\mathcal{A}_0 = \{\text{not, no, illegal, against, without}\}$. Importance scores in LIME are weights of the linear interpretable model. With OLLIE, importance scores of extracted triplexes are calculated in the same way as in our method (as shown in Eq. 4). The contextual constraint γ in Eq. 2 is 3 for drug abuse or 10 for consumer complaint dataset. The pre-defined threshold in Eq. 3 is 0.5. LIME and OLLIE settings are used as default. We only show OLLIE rules which have the confidence score greater than 0.7 and top-5 words from LIME.

It is important to note that, to be fair, we also combined the learned anchors to the results of OLLIE. In addition, another variation of our algorithm is to combine ontology-based terms and anchors, called **Ontology** algorithm. This is further used to comprehensively evaluate our proposed approach.

Quantitative evaluation. We use the word deleting approach [28], which deletes a sequence of words from the text and then re-classifies the text with missing words. By the difference between the original text and the missing text, we can examine the importance of the explanation to the prediction. Accuracy changes (AC) and prediction score changes (SC) are as follows:

$$\text{AC} = \text{Original accuracy} - \frac{\sum_{i=1}^{|test|} \text{Updating accuracy}}{|test|}$$

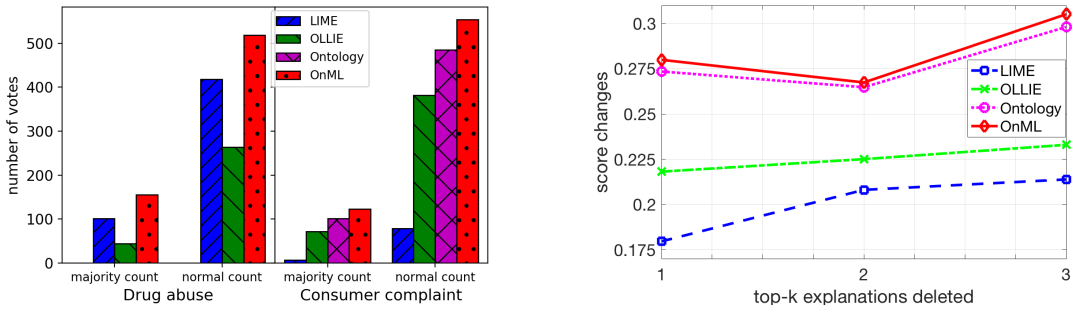
$$\text{SC} = \frac{\sum_{i=1}^{|test|} IC(\text{top-}k \text{ explanations of } i\text{-th sample})}{|test|}$$

where the higher values of AC and SC indicate the more important explanations derived.

In our experiment, we deleted the top-k highest importance score explanations in OnML and OLLIE approaches and the top-m highest weighted words in LIME. To be fair, m is the number of words in

Table 1: Data statistical analysis.

Statistics \ Dataset	Drug abuse	Consumer complaint
# of samples	9,700	13,965
# of categories	2	16
Max # of words/sentence	37	4,893
Mean # of words/sentence	12	285



(a) AMT experiment results.

(b) Average score changes in consumer complaint.

Figure 3: Qualitative and quantitative experiment results.

the k-deleted explanations in OnML. In drug abuse, $k = 1$ since the tweet is typically short, and so there are not many explanations generated. In consumer complaint classifying, $k \in \{1, 2, 3\}$.

Qualitative evaluation. We recruit human subjects on Amazon Mechanical Turk (AMT). This is a common means of evaluation for the needs of qualitative investigation by humans [6, 34]. Detailed guidance for each experiment is provided to users before they conduct the task.

We asked AMT workers to choose the best explanation by seeing side-by-side explanation algorithms. On top of that, we provided the original tweet/ complaint associated with their labels and prediction results. The visualization showing explanation results of the approaches is presented in Fig. 2. **It is important to note that**, in our real experiment, to avoid bias, name of each algorithm is hidden, and their positions in the visualization are randomized.

We were recruiting 4 users/tweets in the drug abuse and 5 users/complaints in the consumer complaint experiment. There are two ways to quantify the voting results from AMT users: **(1)** Count the total number of votes, called *normal count*, i.e., the best algorithm is chosen over all $1,500$ votes (5 users/complaint $\times$ 300 complaints); and **(2)** Count the majority number of votes, called *majority count*, i.e., the best algorithm for each complaint is the algorithm of the largest number over 5 votes.

4.4 Experimental results and analysis

300 positive tweets and 300 complaints, randomly selected, were used to evaluate the interpretability of each approach.

Drug abuse explanation. As in Table 2, the accuracy is deducted significantly, and the predictive score changes the most in OnML. In fact, the values of AC and SC are 25.52% and 33.48% given OnML, compared with 15.47% and 23.52% given OLIIE, and 15.04% and 26.98% given LIME. This demonstrates that the explanations generated by our algorithm are more significant, compared with the ones generated by baseline approaches. In the evaluation by humans using AMT (Fig. 3a), OnML clearly outperforms LIME and OLLIE. Text in the tweet data is generally short and can be represented by several key words in the tweet. Therefore, individual words learned by LIME can be sufficient to generate more insightful explanations, compared with OLLIE. Meanwhile, OLLIE tends to extract all possible triplexes in the text, which can be redundant and wordy explanations.

Consumer complaint explanation. The results on the consumer complain dataset further strengthen our results. Fig. 3b shows SC after deleting top-1, top-2, and top-3 explanations from OnML, Ontology, and OLLIE, as well as after deleting the most important words in LIME. In all three cases,

Table 2: AC and SC in drug abuse.

	Accuracy changes (%)	Score changes (%)
LIME	15.04	26.98
OLLIE	15.47	23.52
OnML	**25.52**	**33.48**

score changes in OnML have the highest values, indicating that the explanations generated by OnML are the most significant to the prediction. In the evaluation by humans using AMT (Fig. 3a), our OnML algorithm outperforms baseline approaches. Ontology approach achieves higher results than LIME and OLLIE. This shows the effectiveness of the ontology-based approach. LIME algorithm does not consider semantic correlations among words/features, resulting in a poor outcome.

Completeness and concision. In Fig. 2 (*top*), OnML generates "i smoking weed," which provides concise and complete information about why it is predicted as a drug abuse tweet (smoking weed) and who was doing it (i) in a syntactic form (S-V-O). Meanwhile, 1) LIME derives relevant words to drug abuse (i.e., weed, smoking) without considering the correlation among these words; and 2) OLLIE generates lengthy and somewhat irrelevant explanations, e.g., "chinese food; be eating on; a roof." In Fig. 2 (*bottom*), OnML derived semantic explanations for consumer complaints, which tell us that consumers were facing issues in loan refinance, e.g., "called fha and they clain that fha dows not review loans." Compared to OnML, Ontology generates laconic explanations, e.g., "fha loan" that give no sense of what is going on with the loan and why consumer complaints. Meanwhile, LIME provides a set of fragmented words and OLLIE generates wordy explanations, which are difficult to follow. More examples of drug abuse and consumer complaint explanation are in the Appendix.

Our key observations are: **(1)** Combining correlated terms, anchor texts, and information extraction can generate complete, concise, and insightful explanations to interpret the prediction model f; and **(2)** Our OnML model outperforms other baseline approaches in both the quantitative and qualitative experiments, showing a promising result.

5 Conclusion

In this paper, we proposed a novel ontology-based IML to generate semantic explanations, by integrating interpretable models, ontologies, and information extraction techniques. A new ontology-based sampling technique was introduced, to encode semantic correlations among features/terms in learning interpretable representations. An anchor learning algorithm was designed to limit the search space of good explanations. Then, a set of regulations for connecting learned correlated terms, anchor texts, and extracted triplexes is introduced, to produce semantic explanations. Our approach achieves a better performance, in terms of semantic explanations, compared with baseline approaches, illustrating a better interpretability into ML models and data. Our approach paves an early brick on a new road towards gaining insights into machine learning using domain knowledge.

References

[1] S. M. Robnik and I. Kononenko, "Explaining classifications for individual instances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 589–600, 2008.

[2] D. Martens and F. Provost, "Explaining data-driven document classifications," 2013.

[3] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, "Making the V in VQA matter: Elevating the role of image understanding in visual question answering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6904–6913.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1135–1144.

[5] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[7] H. Phan, D. Dou, H. Wang, D. Kil, and B. Piniewski, "Ontology-based deep learning for human behavior prediction with explanations in health social networks," *Information Sciences*, vol. 384, pp. 298–313, 2017.

[8] R. Confalonieri, F. M. delPrado, S. Agramunt, D. Malagarriga, D. Faggion, T. Weyde, and T. R. Besold, "An ontology-based approach to explaining artificial neural networks," *arXiv preprint arXiv:1906.08362*, 2019.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] Han Hu, NhatHai Phan, James Geller, Stephen Iezzi, Huy Vo, Dejing Dou, and Soon Ae Chun, "An ensemble deep learning model for drug abuse detection in sparse twitter-sphere," in *MEDINFO'19)*, 2019.

[11] S. Nagrecha, J. Z. Dillon, and N. V. Chawla, "Mooc dropout prediction: lessons learned from making pipelines interpretable," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 351–359.

[12] A. Adhikari, D. M. Tax, R. Satta, and Ma. Fath, "Example and feature importance-based explanations for black-box machine learning models," *arXiv preprint arXiv:1812.09044*, 2018.

[13] Y. Jia, J. Bailey, K. Ramamohanarao, C. Leckie, and M. E. Houle, "Improving the quality of explanations with local embedding perturbations," 2019.

[14] L. Freddy and W. Jiewen, "Semantic explanations of predictions," vol. arXiv:1805.10587v1, 2018.

[15] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web.," in *International Joint Conference on Artificial Intelligence*, 2007, vol. 7, pp. 2670–2676.

[16] F. Wu and D. S. Weld, "Open information extraction using wikipedia," in *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, 2010, pp. 118–127.

[17] S. Soderland, B. Roof, B. Qin, S. Xu, O. Etzioni, et al., "Adapting open information extraction to domain-specific relations," *AI magazine*, vol. 31, no. 3, pp. 93–102, 2010.

[18] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1535–1545.

[19] M. Schmitz, R. Bart, S. Soderland, O. Etzioni, et al., "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 523–534.

[20] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 2017, pp. 4765–4774.

[21] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3145–3153.

[22] M. Sundararajan, A. Taly, and Q. Yan, "Gradients of counterfactuals," *arXiv preprint arXiv:1611.02639*, 2016.

[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[24] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, pp. e0130140, 2015.

[25] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[26] S. Barlas, "Prescription drug abuse hits hospitals hard: Tighter federal steps aim to deflate crisis," *Pharmacy and Therapeutics*, vol. 38, no. 9, pp. 531, 2013.

[27] E. W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768–769, 1965.

[28] L. Arras, F. Horn, G. Montavon, K. R. Müller, and W. Samek, ""What is relevant in a text document?": An interpretable machine learning approach," *PloS one*, vol. 12, no. 8, pp. e0181142, 2017.

[29] J. Ramos et al., "Using TF-IDF to determine word relevance in document queries," in *Proceedings of the first Instructional Conference on Machine Learning*, 2003, vol. 242, pp. 133–142.

[30] T. Mikolov, K. Chen, G. S. Corrado, and J. A. Dean, "Computing numeric representations of words in a high-dimensional space," 2015, US Patent 9,037,464.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[32] D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?," in *European Conference on Computer Vision*, 2010, pp. 71–84.

[34] D. Martens, B. Baesens, T. Van G., and J. Vanthienen, "Comprehensible credit scoring models using rule extraction from support vector machines," *European Journal of Operational Research*, vol. 183, no. 3, pp. 1466–1476, 2007.

Appendix

Domain Ontologies

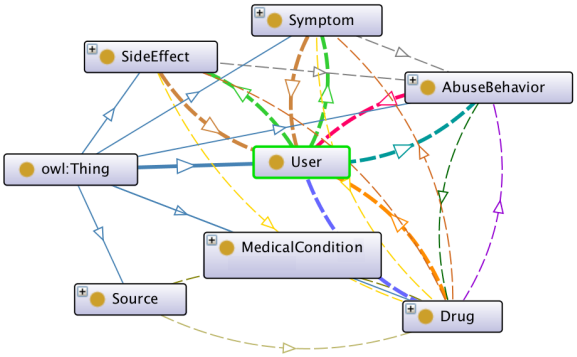


Figure 4: Main concepts of a drug abuse ontology generated by our team experts.

Drug abuse behavior is defined as the use of a medication, legally (e.g. legal painkiller and weed) or illegally (e.g. getting drugs without prescription or even from blackmarket), without a prescription, in a way other than as prescribed, but for personal experience of getting “high" or feeling “numb." The drug abuse ontology captures different concepts collected from drug abuse tweets, grouped by K-means clustering algorithm, and then finalized by our team experts. Main concepts of the drug abuse ontology (**DrugAO**) (Fig. 4) capture correlation among them and are related to drug abuse. There are seven major concepts in DrugAO, which are *AbuseBehaviror*, *Drug*, *Source*, *User*, *Symptom*, *SideEffect*, and *MedicalCondition*. AbuseBehavior is about behaviors of abusers, such as abuse, addict, blunt, etc. Drug consists of different types of legal and illegal drugs, e.g., narcotics, cocaine, and weed. Source is where User, who are the main objects of the ontology, gets drugs from. Symptom and SideEfffect are about different negative short-term and long-term effects of drugs on users. MedicalCondition contains terms about expression of disease and illness caused by using drugs. In total, we have 506 drug-abuse related terms, 18 relations covered in our DrugAO.

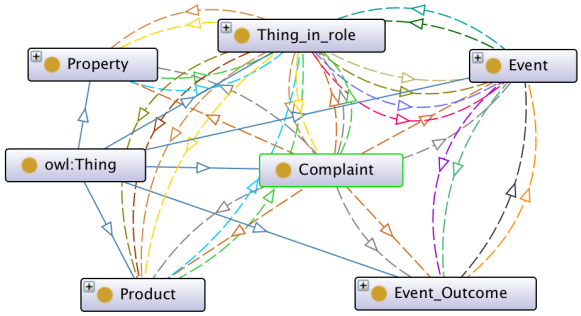


Figure 5: Main concepts of a consumer complaint ontology generated by our team experts.

A consumer complaint is defined as a complaint about a range of consumer financial products and services, sent to companies for response. In complaints, consumers typically talk about their mortgage-related issues, such as: (1) Applying for a mortgage or refinancing an existing mortgage (application, credit decision, underwriting); (2) Closing on a mortgage (closing process, confusing or missing disclosures, cost); (3) Trouble during payment process (loan servicing, payment processing, escrow accounts); (4) Struggling to pay mortgage (loan modification, behind on payments, foreclosure); (5) Problem with credit report or credit score; (6) Problem with fraud alerts or security freezes, credit monitoring or identity theft protection services; and (7) Incorrect information on consumer’s report or improper use of consumer’s report. The consumer complaint ontology (**ConsO**) (Fig. 5) encodes the relation among different entities related to consumer complaints. There are six major concepts in ConsO, which are Thing in role, Complaint, Event, Event outcome, Property, and Product. Thing in role is people and organizations related to Complaint, such as buyers, investors, dealers, et,. Event and Event outcome are about negative events happened that cause consumer complaints. Property is

things belonging to consumers and Product is substances of some parties (e.g., banks) offering to consumers. We have 572 finance and product-related terms and 9 relations covered in our ConsO.

Additional Experimental Results

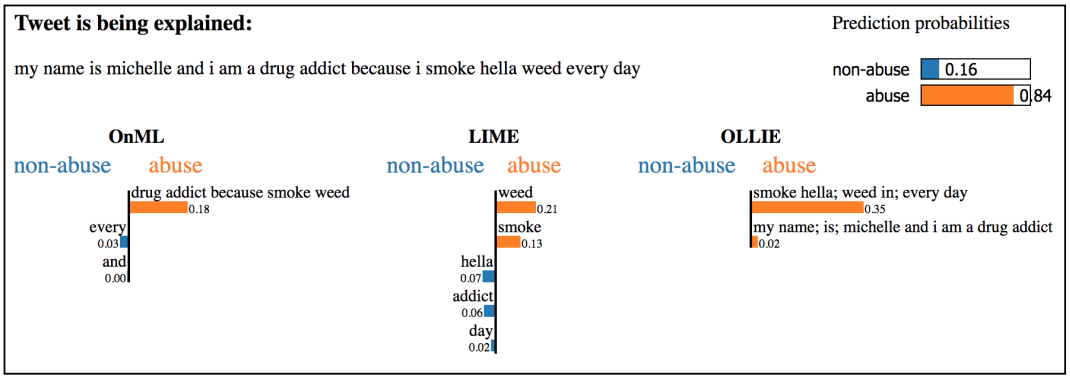


Figure 6: Visualization of a drug abuse experiment.

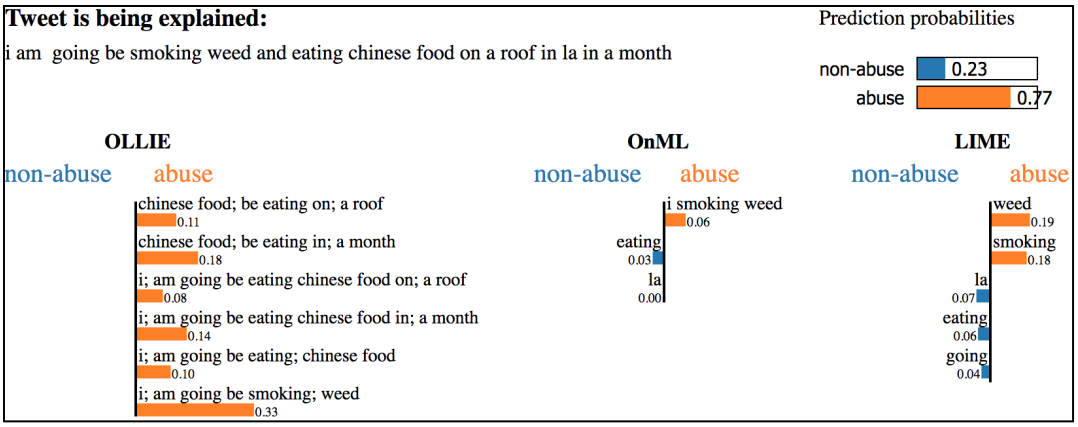


Figure 7: Visualization of a drug abuse experiment.

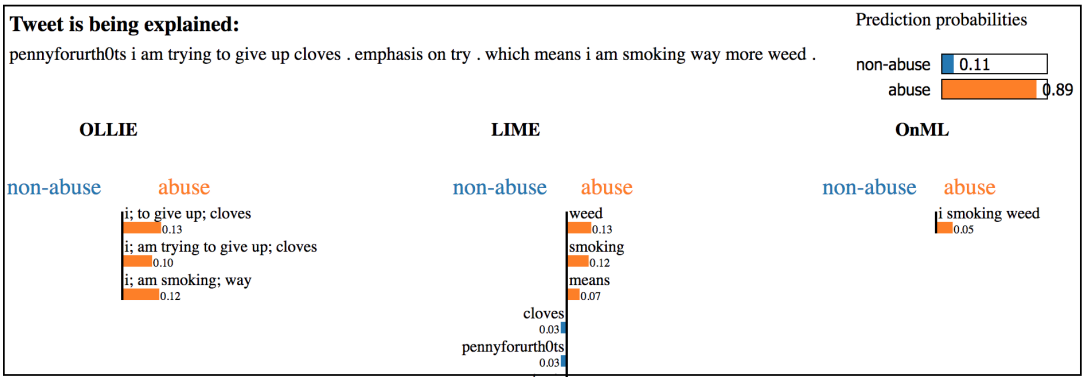


Figure 8: Visualization of a drug abuse experiment.

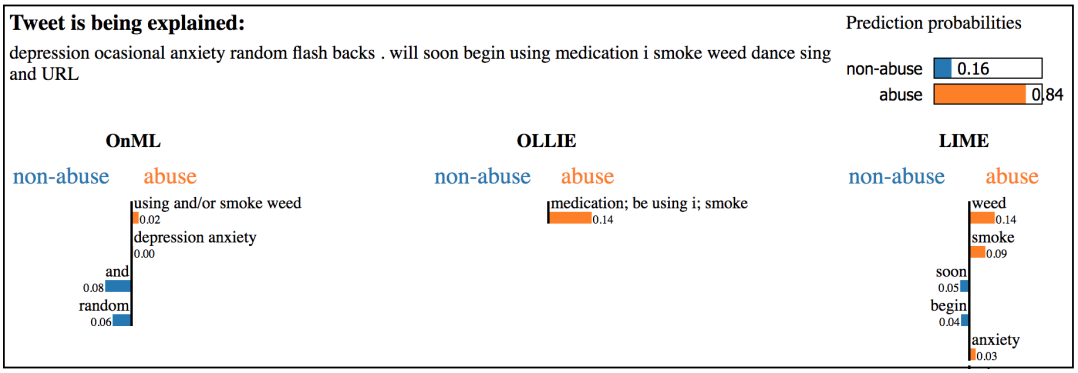


Figure 9: Visualization of a drug abuse experiment.

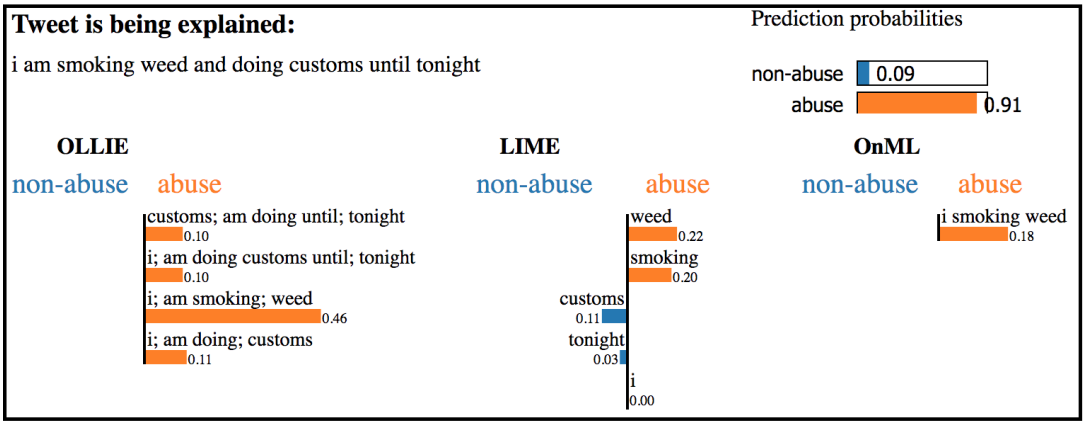


Figure 10: Visualization of a drug abuse experiment.

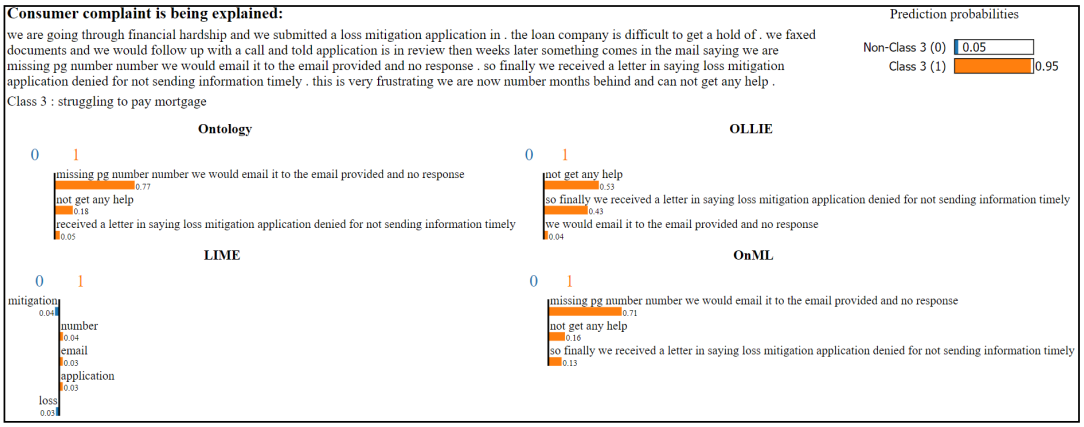


Figure 11: Visualization of a consumer complaint experiment.

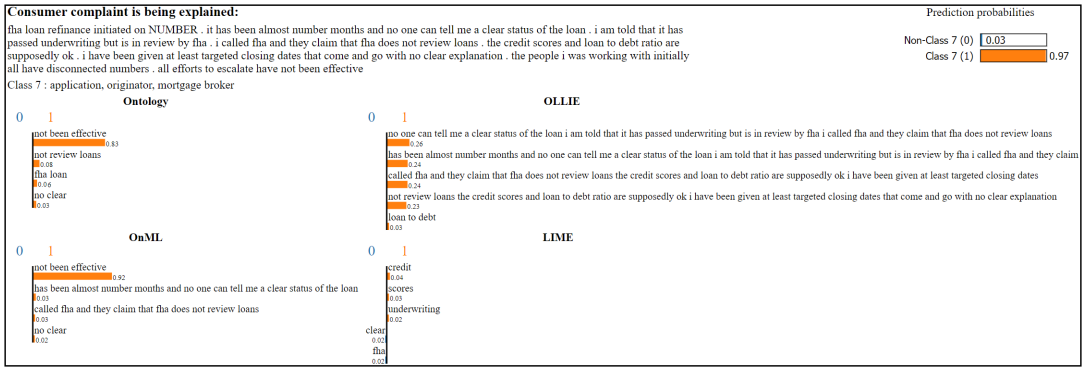


Figure 12: Visualization of a consumer complaint experiment.

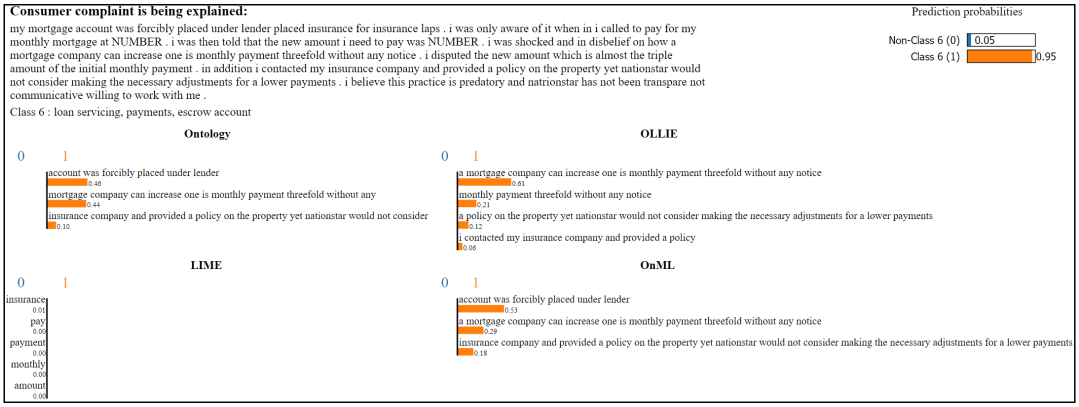


Figure 13: Visualization of a consumer complaint experiment.

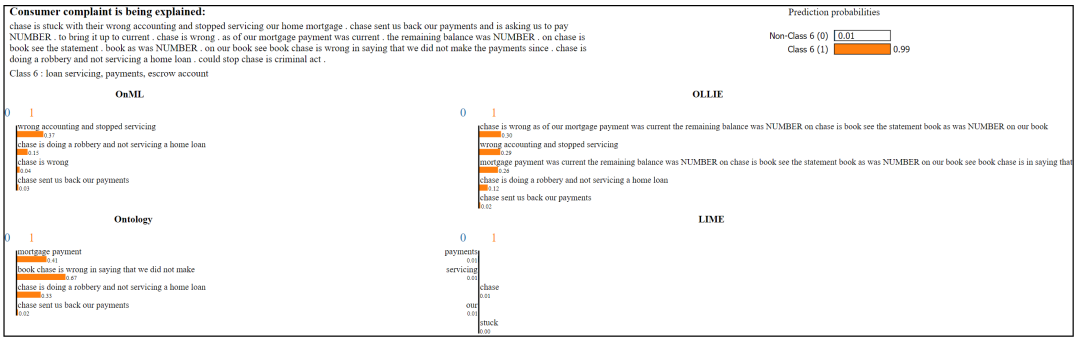


Figure 14: Visualization of a consumer complaint experiment.

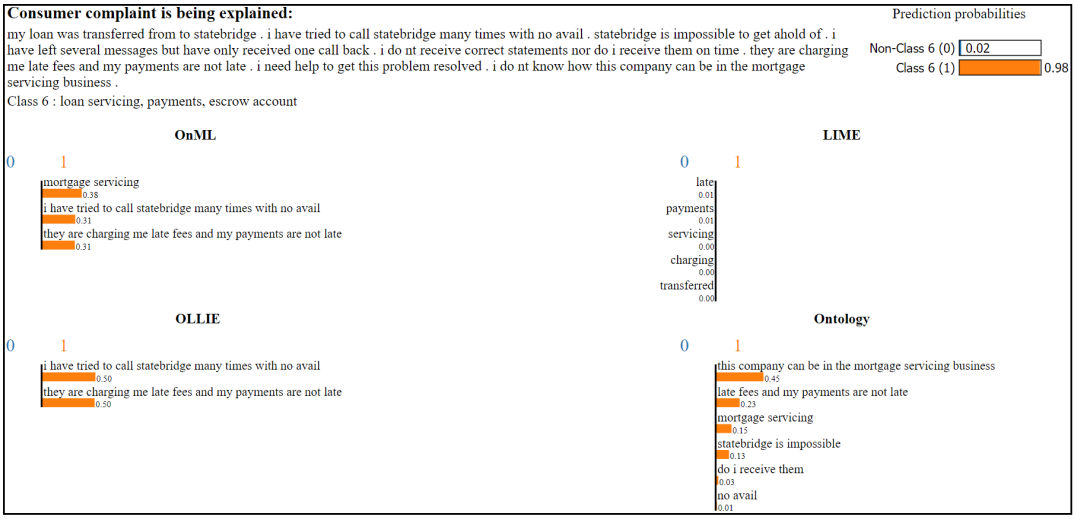


Figure 15: Visualization of a consumer complaint experiment.