Automated Trustworthiness Testing for Machine Learning Classifiers

Steven Cho
University of Auckland
Auckland, New Zealand
scho518@aucklanduni.ac.nz

Seaton Cousins-Baxter
University of Auckland
Auckland, New Zealand
scou766@aucklanduni.ac.nz

Stefano Ruberto
JRC European Commission
Ispra, Italy
stefano.ruberto@ec.europa.eu

Valerio Terragni
University of Auckland
Auckland, New Zealand
v.terragni@auckland.ac.nz

***Abstract*—Machine Learning (ML) has become an integral part of our society, commonly used in critical domains such as finance, healthcare, and transportation. Therefore, it is crucial to evaluate not only whether ML models make correct predictions but also whether they do so for the correct reasons, ensuring our trust that will perform well on unseen data. This concept is known as trustworthiness in ML. Recently, explainable techniques (e.g., LIME, SHAP) have been developed to interpret the decision-making processes of ML models, providing explanations for their predictions (e.g., words in the input that influenced the prediction the most). Assessing the plausibility of these explanations can enhance our confidence in the models' trustworthiness. However, current approaches typically rely on human judgment to determine the plausibility of these explanations.**

This paper proposes TOWER, the first technique to automatically create trustworthiness oracles that determine whether text classifier predictions are trustworthy. It leverages word embeddings to automatically evaluate the trustworthiness of a model-agnostic text classifiers based on the outputs of explanatory techniques. Our hypothesis is that a prediction is trustworthy if the words in its explanation are semantically related to the predicted class.

We perform unsupervised learning with untrustworthy models obtained from noisy data to find the optimal configuration of TOWER. We then evaluated TOWER on a human-labeled trustworthiness dataset that we created. The results show that TOWER can detect a decrease in trustworthiness as noise increases, but is not effective when evaluated against the human-labeled dataset. Our initial experiments suggest that our hypothesis is valid and promising, but further research is needed to better understand the relationship between explanations and trustworthiness issues.

***Index Terms*—Machine Learning, Machine Learning Testing, Test Oracle Problem, Trustworthiness, Explainability, Software Testing, Text Classifications, Word Embeddings**

I. Introduction

Machine Learning (ML) has become integral to many areas of society. With its prevalence, it is crucial to determine whether we can *trust* ML models. *Trust goes beyond correctness*. An ML model can achieve high accuracy and make correct predictions yet still be untrustworthy. This occurs when the reasons behind its correct predictions are flawed, making the model unreliable for unseen data.

One way to measure trustworthiness is through understanding the decision-making process of ML models [5], [7], [15]. While there are ML classifiers that are inherently explainable (e.g., decision trees), most ML classifiers (e.g., (deep) neural networks) cannot directly explain their decisions in a way that a human would understand. However, the area of **Explainable Machine Learning (XML)** [26] studies techniques to explain the predictions of any classifier, as long as it has interpretable inputs (i.e., text, numbers, or images).

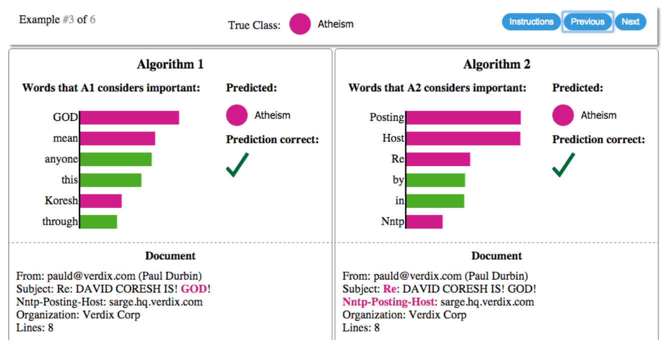


Fig. 1: Example of "untrustworthy" prediction from LIME paper [25]. Algorithm 2 based the prediction on irrelevant words: "Posting", "Host", "Re", and "Nntp"

One of the first and most representative XML technique is LIME [25]. It produces textual or visual artifacts for a qualitative understanding of the relationship between the test input instance's components (e.g., words in a text) and the model's prediction. For example, for text classification, LIME produces a list of words that most influenced the prediction. Given such explanations, a human can judge if they are coherent with the problem domain, and thus the ML model is trustworthy and will likely generalize well to unseen data.

An example discussed in the LIME paper is an ML model that classifies emails belonging to "Christianity" or "Atheism" newsgroups (see Fig. 1). Although the classifier achieves 94% accuracy, LIME's explanations show that some predictions were made for arbitrary reasons: the words "Posting", "Host", and "Re" were prevalent in only one class and have nothing to do with either "Christianity" or "Atheism" (see algorithm 2 in Fig. 1). While it is easy for a human to conclude that these predictions are not trustworthy, relying on the manual analysis of classifier explanations is very costly.

This paper presents **T**rustworthiness **O**racle through **W**ord **E**mbedding **R**elatedness (**TOWER**), **the first automated trustworthiness testing technique for ML text classifiers**. TOWER generates automated oracles that replace a human in judging if the explanations of a ML prediction are trustworthy or not. The key advantage of an automated approach is to avoid the manual cost of inspecting XML's explanations. Manual judgement (as envisioned by Ribeiro et al. when proposing LIME) is infeasible to perform for many ML predictions.

TOWER is based on the intuition that such trustworthy oracles can be automatically constructed by relying on **word embedding models**. Embedding models use vectors to represent words in a low-dimensional space to quantify semantic relatedness between words. All these models have in common that two words are semantically related if the vectors representing them are close in the vector space. The inverse of such a distance is the semantic similarity score (usually [0,1]). Using word embeddings, TOWER automatically checks if the explanations produced by XML techniques contain words that are semantically related to the class name/description. For example, consider the "Christianity" and "Atheism" classifier discussed above, the `word2vec-google-news-300` embedding model returns low semantic similarity scores between the LIME explanation "Post" and class labels "Christianity" and "Atheism" (0.054 and 0.065, respectively). In the context of TOWER, such a low similarity score could indicate trustworthiness issues in the data or model (like in the example discussed in Fig. 1).

We trained TOWER's configuration parameters using unsupervised learning with around 1,600 untrustworthy and trustworthy models obtained from noisy data as a proxy for trustworthiness issues. Then, we took the learned optimal configuration of TOWER and evaluated it by creating a human-labeled trustworthiness dataset of 328 instances.

On one hand, experiments with noisy data show promising results, demonstrating that TOWER can effectively detect a decrease in trustworthiness as noise increases. On the other hand, when evaluated against the human-labeled dataset, the TOWER trained configuration exhibited low performance. Our initial experiments suggest that our hypothesis is valid and promising, but further research is needed to understand the poor performance on the human-labeled dataset. This might be because the types of noise considered are not a good proxy for real-world trustworthiness issues. Nonetheless, this paper marks the first significant step towards automated trustworthiness testing, which is an exciting new research direction.

In summary, this paper makes the following contributions:

- We define the concept of automated trustworthiness testing and oracles for ML text classifiers;
- We present TOWER, the first technique to automatically determine the trustworthiness of a text classifier;
- We discuss a series of experiments that highlights the limitations of TOWER;
- We release a ground-truth dataset of human-labelled trustworthy and untrustworthy explanations[1].

[1] https://zenodo.org/records/11499368

II. Preliminaries and Problem Formulation

This section formalises the trustworthiness testing problem and gives the preliminaries of this work.

Trustworthiness as a concept is relevant across many disciplines. In ML, trustworthiness is generally defined as a combination of related concepts, such as robustness, security, transparency, fairness, and safety [19], [31]. Instead, in this paper, we rely on the following definition by Kästner et al. [18]:

Definition 1. *A system is* ***trustworthy*** *to a stakeholder S in a context C if the system works properly in C, and S is justified in their belief of that, given they believe it.*

Colloquially, *a system is trustworthy if it is 'right for the right reasons'*. For ML classifiers, 'right' means that the classification is correct and these 'right reasons' can be determined through *prediction explanations*.

Explainability. Most ML models are black-box; that is, their internal reasoning is opaque to an observer. To understand and trust these models, the fields of explainable AI (XAI) and explainable ML (XML) have emerged [26].

Various XML methods enables the interpretability of black-box ML models, including saliency maps, feature attribution methods, counterfactual and contrastive explanations, white-box models, and global surrogates [4]. We will focus on techniques that provide textual output – specifically, feature attribution methods. These techniques identify the most important features in an input for a given prediction.

LIME [25], one of the most popular of these methods, works by creating a simplified model that approximates the behavior of a more complex model for a particular input. This is a system-agnostic post-hoc explainability technique. The output of LIME is a list of words from the input that it deems important to the prediction, along with their corresponding importance scores—weights measuring the importance of the phrase. Henceforth, we will refer to this phrase-score list as an *explanation*.

Explanations have two properties: *faithfulness*, which measures how well the explanation represents the workings of the model; and *plausibility*, which measures how convincing the explanation is to a human [30]. Following previous work on trustworthiness [18], [20], [25], we use the plausibility of explanations as a proxy for measuring the trustworthiness of a prediction.

Definition 2. *Given a ML prediction and its explanation, a* ***trustworthiness oracle*** *is a function that returns* true *if the explanation is plausible and thus the prediction is trustworthy, and* false *otherwise.*

The goal of TOWER is to produce such oracles automatically by leveraging word embedding techniques to measure *plausibility*.

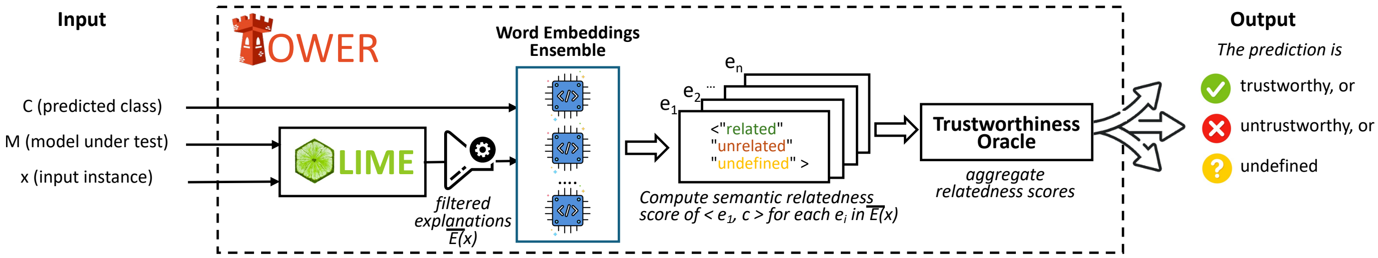


Fig. 2: Logical architecture of TOWER. Given an input instance x, a ML classifier model under test M, and the predicted class c, TOWER automatically judges if the prediction $M(x) = c$ is trustworthy, untrustworthy, or undefined (non enough information/confidence to make a decision).

III. TOWER

This paper presents TOWER, a technique for automatically testing the trustworthiness of a given text classifier using explainability techniques and word embeddings.

Fig. 2 provides an overview of TOWER's logical architecture. The inputs to TOWER include an input instance x, a machine learning classifier model M under test, and the predicted class c. TOWER implements an automated trustworthiness oracle to determine if the prediction $M(x) = c$ is trustworthy, untrustworthy, or undefined (insufficient information/confidence to make a decision). Given a large number of labelled instances, we can perform trustworthiness testing and calculate the percentage of trustworthy and untrustworthy predictions.

Internally, TOWER uses LIME [25] to obtain explanations for the model M's prediction on x. It then filters out explanations with low importance scores (as measured by LIME). For each remaining explanation word $e_i \in \overline{E}(x)$, TOWER relies on an ensemble of word embeddings to compute the semantic relatedness score with the predicted (and correct) class c. TOWER aggregates the relatedness scores of all explanation words to make an informed decision about whether M's prediction on x is trustworthy. TOWER can also respond with ”undefined” if the word embeddings lack sufficient confidence about the semantic relatedness. We now discuss the internal components of TOWER in more detail.

A. Input

TOWER takes as input an ML model under test, M, an input instance, x, and the predicted class, c. Due to the nature of TOWER, it currently supports only text classification models, where the classes to be predicted are well-defined entities (e.g., “Christianity” and “Atheism”). This is because TOWER needs to measure the semantic relatedness between the explanations (which must be text-based) and the predicted class (which should be a well-defined entity). More specifically, x is a text, and c is the predicted class to which x belongs. The current version of TOWER supports classes composed of a single word. The text classifier M can be any black-box machine learning model that is supported by explanation techniques. This includes models obtained with popular machine learning techniques like neural networks.

Note that we do not consider any input instances that are not correctly predicted by the model under test, as these prediction would, by Definition 1, be untrustworthy. Thus, we assume that the class c predicted by the model is the correct class for x. This assumption can be easily met when testing the model with labeled data, as we can automatically ignore instances that are not correctly classified.

B. Explanations

The first step of TOWER uses a post-hoc black-box explanation technique to obtain the explanation of the prediction. The current implementation of TOWER uses LIME [25], but it could be replaced with any similar technique. TOWER assumes that the obtained explanation is faithful, i.e., it correctly represents the workings of the model (see Section II). This is a reasonable assumption given the good performance of LIME and similar techniques [25].

Given the model M and the input instance x, LIME perturbs x to identify the words that contributed the most to its decision-making, which constitutes the explanation. More formally, the explanation E of x is a list of words in x with associated importance scores s, ordered by s. Formally, $E(x) = \langle (e_1, s_1), (e_2, s_2), \ldots, (e_n, s_n) \rangle$, where $e_i \in x \quad \forall i = 1, \ldots, n$ and $s_i > s_{i+1} \quad \forall i = 1, \ldots, n-1$.

TOWER has two configuration parameters to filter the words in the explanations because some of them might have low importance scores. The *explanation threshold* filters out explanations with importance scores below a certain threshold, and the *explanation top-n* parameter considers only the top n words on the list. TOWER also ignores words with negative importance scores, as they represent importance towards different classes. These are not useful for measuring the relatedness of impactful words towards the predicted class. We use $\overline{E}(x)$ to denote the filtered explanation.

C. Word Embeddings Ensemble

The second step of TOWER uses word embeddings to measure the relatedness among each word in the filtered explanation $\overline{E}(x)$ and the class c. Word embeddings are vector representations of words, showing their semantic relationship, as found via their use in similar contexts. Two words are semantically related if their vectors are close in the vector space, typically measured using cosine similarity. The inverse of this distance is the semantic relatedness score. Our intuition is that untrustworthy predictions should contain unrelated words in their explanations.

Several challenges must be addressed when using word embeddings to detect untrustworthy predictions:

1) How do we deal with word embedding models bias? Word embedding models are trained on specific datasets, introducing biases. To mitigate this, TOWER uses an **ensemble approach**, combining the results of multiple word embedding techniques trained on different datasets. This strategy should also enhance the effectiveness of the semantic similarity score computation.

2) How do we deal with word embedding uncertainty? Word embeddings provide a relatedness score (usually between 0 and 1), necessitating a threshold τ to decide if the score is high enough to indicate related words. Setting a single threshold τ to differentiate trustworthy from untrustworthy explanations could result in many false positives. The semantic distances returned by word embedding models are not precise enough to ensure that a score of τ + 0.01 is trustworthy while τ - 0.01 is not. To address this, scores within a certain range of the threshold are considered 'undefined'; scores above this range are 'related' and those below are 'unrelated'. This mitigates false positives by avoiding definite decisions when the models are uncertain. The range is a configurable parameter.

3) How do we set the thresholds? Finding a single threshold that works for all embedding methods in the ensemble is challenging, as semantic relatedness scores are specific to each method and cannot be directly compared. Therefore, we propose an automated way to determine a dedicated relatedness threshold τ for each embedding method.

First, we constructed a dataset of word pairs, each containing either related or unrelated words. We identified the top 1,000 most common English words according to WORDNET [21]. Then, using the Merriam-Webster API, we generated 32,000 pairs of related words by finding synonyms for each common word. We also created 32,000 pairs of unrelated words by randomly pairing the original 1,000 words (ensuring no synonyms were paired together).

Second, we queried each word embedding model with each of the 64,000 pairs to obtain the semantic relatedness scores. Using a binary search, we identified the optimal threshold that balances precision and recall in discerning related and unrelated pairs for each embedding model. TOWER uses these thresholds to classify each word $e_i \in \overline{E}(x)$ as "related", "unrelated", or "undefined".

The final output of the ensemble is derived from the relatedness scores from each model through two methods: aggregation (taking the proportion of models returning each classification) and voting (taking the classification with the most models). A weighting can also be applied to each model in the ensemble derived from their area-under-the-curve (AUC) score based on their performance on the synonym data. These are all configurations parameters of TOWER.

D. Trustworthiness Oracle Outcome

Finally, TOWER determines the trustworthiness of the prediction by combining the relatedness scores for each word in the filtered explanation. TOWER can use several methods to combine these results: averaging (a prediction is trustworthy in proportion to the relatedness scores), plurality (a prediction is trustworthy if the majority of words in $\overline{E}(x)$ are related to c), and sufficiency (a prediction is trustworthy if at least one word in $\overline{E}(x)$ is related to c). Averaging these outcomes over many test cases provides the overall trustworthiness of the model.

Note that, unlike Definition 2, the oracle might have a third outcome, "undefined", due to the ensemble's uncertain responses. This approach is intended to avoid false positives, acknowledging that even humans can be indecisive when judging the plausibility of explanations [25].

IV. EVALUATION

We conducted a series of experiments to answer two research questions:

RQ1 Unsupervised training on noisy models: *Is it possible to train TOWER configuration parameters without labelled data but using noisy data instead?*

RQ2 Human-based evaluation: *How effective is the trained configuration of TOWER if evaluated with human labelling of trustworthiness text classifications?*

We conducted two experiments to answer these questions.

Datasets with explanations labelled as trustworthy or not are rare and expensive to create. Thus, for **RQ1**, we first create artificial datasets to train and evaluate the best configuration TOWER. To approximate this labelling, we make the assumption that models trained on noisy data will produce less trustworthy explanations, in proportion to the level of noise. Indeed, it is a common practice to use increasing artificial noise levels to produce relatively untrustworthy models [1], [17], [24], [25], [33]. For this, we artificially inject different levels of noise into datasets and train models on them, producing results with assumedly different proportions of trustworthy and untrustworthy explanations. We measure the relation between this increasing noise level and TOWER's output. Through this, we trained the optimal tool configuration for TOWER with respect to the produced untrustworthy models. Finally, we verify that we have a configuration of TOWER that can detect a decrease in trustworthiness with the increase of noise level.

For **RQ2**, we create a trustworthiness dataset – that is, a dataset of explanations labelled by humans as trustworthy or not. This dataset is more in line with what would be found in real-world data than the artificial dataset – we make no assumptions of the production of trustworthiness – and can be considered a ground truth. We then use this data to evaluate TOWER in its optimal configuration.

A. Implementation

We implemented TOWER with the following five word embedding methods: FASTTEXT, GLOVE, NEURAL-NET LANGUAGE MODEL (NNLM), SWIVEL, and UNIVERSAL

SENTENCE ENCODER (USE). These are popular methods commonly used in natural language processing. The explanation technique used is LIME at 5,000 iterations [25].

B. RQ1 - Unsupervised training on noisy models

We produced untrustworthy models as follows:

- **Model type.** Through scikit-learn, we chose the following four classifier models: multinomial naive Bayes (MNB), decision tree (DT), random forest (RF), and stochastic gradient descent (SGD). We picked these as they are some of the most popular machine learning algorithms. We use five-fold cross-validation to train these classifiers.
- **Dataset.** By popularity on HuggingFace[2], we chose the following five datasets: 20 Newsgroups, AG News, DBpedia14, Emotion, and IMDB. The 20 Newsgroups dataset was also featured in the original LIME paper [25]. Details can be found in Table I.
- **Noise type.** For each instance, we can create a noisy version via the following four chosen methods:
  - *Removal noise.* We remove 30%-70% of the words from the instance [17].
  - *Label noise.* We change the label of the instance to a different one at random [17], [33].
  - *Bias.* For each category in the dataset, we choose a unique random sentence[3]. The random sentence chosen for the instance's category is appended to the end of the instance. The sentence is different between categories, but is the same sentence for each instance within the category.
  - *Natural noise.* This is specific for the 20 Newsgroups dataset. The original dataset has mostly-irrelevant headers and footers, which were removed. For this noise method, we re-introduce these to the instance. This was done in the experiment in the original LIME paper [25].

We do this for five levels of noise: 0%, 25%, 50%, 75%, and 100%. These levels indicate what percentage of instances of the training set are injected with noise. At 25%, for example, 25% of the training instances are randomly chosen and replaced with their noisy versions.

For each noise type, the test set is the same for each noise level. It consists of all instances from the original clean test set combined with the noisy version of each instance, doubling the set's overall size. We do this as we want to see the models' explanations for both clean and noisy data. We also want all models of each noise type to be tested on the same data, regardless of their noise levels. This results in a total of 64 sets of models being trained (4 models $\times$ 5 datasets $\times$ 3 noise types + 4 models (natural noise only for 20 Newsgroups) = 64), with 25 models in each set; five-fold for each of the five noise levels (totaling around 1,600 models). The 0% noise models are the same model across each noise type.

[2]https://huggingface.co/

[3]Randomly chosen from random Wikipedia articles.

We then have another set of parameters for the tool itself that we aim to learn with unsupervised learning using the noisy models. These are:

- **Word embedding exclusion range.** $\{$*0, 0.07*$\}$ This controls the range around the thresholds of the word embedding methods for which the relation between two words is considered *undefined*.
- **Word embedding weighting.** $\{$*true, false*$\}$ This determines whether the word embeddings in the ensemble should be weighted by their AUC scores.
- **Relatedness score calculation method.** $\{$*aggregation, voting*$\}$ This determines how the word embedding outputs are combined in the ensemble. *Aggregation* is the percentage of methods that outputted each of *related*, *unrelated*, and *undefined*. *Voting* is the simple plurality of results.
- **Explanation threshold.** $\{$*0, 0.05*$\}$ Only explanatory words with importance scores above this threshold are considered.
- **Explanation top-*n*.** $\{$*5, 10*$\}$ Only the top *n* words of the explanation are considered.
- **Trustworthiness score calculation method.** $\{$*average, plurality, sufficiency*$\}$ This is how the trustworthiness of a prediction is calculated from the relatedness of words.
  - *Average* is the average of each of *related*, *unrelated*, and *undefined*, considered as *trustworthy*, *untrustworthy*, and *undefined*. For example, an explanation of 2 words with relatedness scores of $(0.0, 0.8, 0.2)$ and $(1.0, 0.0, 0.0)$ would have a trustworthiness score of $(0.5, 0.4, 0.1)$.
  - *Plurality* simply takes the largest value of the three.
  - *Sufficiency* considers the prediction trustworthy if any of the words is related. This last option is considered as research has shown that humans may only require one reason to trust something [22].

This results in 96 possible tool configurations. By running TOWER using every combination of both the model and tool configurations, we produce a resultant trustworthiness score for each model. This is a 3-tuple in the form of (*trustworthiness*, *untrustworthiness*, *undefined*), summing to one.

We now wish to find the optimal configuration for TOWER; that is, the configuration with the greatest monotonic decrease in predicted trustworthiness with increasing noise level.

Here, several questions arise. Firstly, obtaining the slope of noise level against trustworthiness score requires a single value for each data point; yet, we have three values for each. Next, the fact that increasing the noise level likely decreases accuracy, which may affect results. Finally, it is possible that the noise types will not be equally good at representing untrustworthiness. Thus, we have another set of options for the analysis of these results:

- **Slope calculation method.** We can calculate the slope either by taking only the decrease in *trustworthiness*; taking only the increase in *untrustworthiness*; or taking the decrease in *trustworthiness*$/($*trustworthiness* $+$ *untrustworthiness*$)$.

TABLE I: Datasets used for evaluations.

Name	Instances	Classes	Description	RQ
20 Newsgroups	19k	20	A collection of newsgroup documents.	RQ1&2
AG News	128k	4	A collection of news articles from over 2,000 sources.	
DBpedia14	630k	14	A collection of texts from DBpedia 2014.	
Emotion	20k	6	A collection of English Twitter messages.	
IMDb	50k	2	A collection of movie reviews from IMDb.	
Amazon Review Polarity	4,000K	2	A collection of Amazon reviews across 18 years.	RQ2
Rotten Tomatoes	11k	2	A collection of movie reviews from Rotten Tomatoes.	
Yahoo Answers Topics	2,000K	10	A collection of texts from Yahoo! Answers.	

- **Adjusted slope.** Currently, for each noise level, all correctly-predicted instances are considered for the slope calculation, regardless of if they are predicted incorrectly by models at other noise levels. To reduce the impact of decreasing accuracy with increasing noise level, the calculation can be adjusted such that, for every noise level, it discards any instance that is not predicted correctly across all noise levels (following Definition 1).
- **Noise types.** What combination of noise types to consider.

This results in 48 analysis methods to determine the effectiveness of a tool configuration (or 96 for the 20 Newsgroups dataset).

We have various possible configurations for model training, tool running, and analysis. We use these now to find the optimal tool configuration.

For each tool configuration, we run TOWER on a chosen model set. This produces a set of five trustworthiness scores, one for each noise level. We wish to find the variation of scores across the noise levels. To model this, we fit a line to these points and suppose that the more negative the slope, the greater the decrease of trustworthiness. This slope is calculated based on a chosen analysis method. An example of these slopes can be seen in Fig. 3.

For each model set, the tool configurations are ranked based on their produced slope. These rankings are summed across all model sets to produce an overall score for each tool configuration. Rankings are used as the slopes cannot be directly compared between the various different options.

This is done for each analysis method. To reduce any bias between the slope calculation types, the analysis method with the least rank variation between these types is selected. This is as we consider each slope calculation type as an equally valid way to measure the slope.

Finally, the tool configuration with the highest overall score for that analysis method is chosen as the most optimal.

RQ1 Results We evaluated TOWER on its *effectiveness on noisy models* (**RQ1**) and found that, assuming increasing noise in the training data of models decreases its trustworthiness, it does detect some form of trustworthiness. Table II shows the slopes averaged across all slope calculation methods of trustworthiness scores against noise level for TOWER at its optimal configuration. Adjusted slopes are used for non-label noise. The more negative the slope, the more the trustworthiness score decreased with increasing noise percentage.

As an example, the *decision tree* models trained on the *20 Newsgroups* dataset using *bias* has an adjusted slope of -0.428 using the $trustworthiness/(trustworthiness + untrustworthiness)$ slope calculation. This indicates an overall decrease of 0.428 in trustworthiness between the models at 0% and 100% noise. In contrast, the same using *removal noise* has a positive slope of 0.220, indicating an unusual increase of trustworthiness score with increasing noise.

We see that, between all model types, this correlation occurs strongly across label and bias noise and moderately across natural noise.

We also see that TOWER did not find much difference in the use of removal noise. This indeed makes sense, as the removal of words in the training set of the models would likely decrease model accuracy, but should not overmuch alter the explanation of correct predictions – especially with the slope calculation adjusted to only consider predictions correct across all noise levels. The semantic correlation that TOWER uses should remain consistent. This is evidence towards the procedure of this evaluation not introducing unwanted biases into TOWER itself.

Of note, an error resulted in the loss of one set of data, which is absent in the Table (the *random forest* models trained on the *DBpedia14* dataset using *label noise*).

In addition, we found that the analysis methods with the least rank variation between the slope calculation types uses adjusted slope, as well as using only bias as the noise type. Thus, of the noise injection methods, bias seems to be the most representative of creating untrustworthiness.

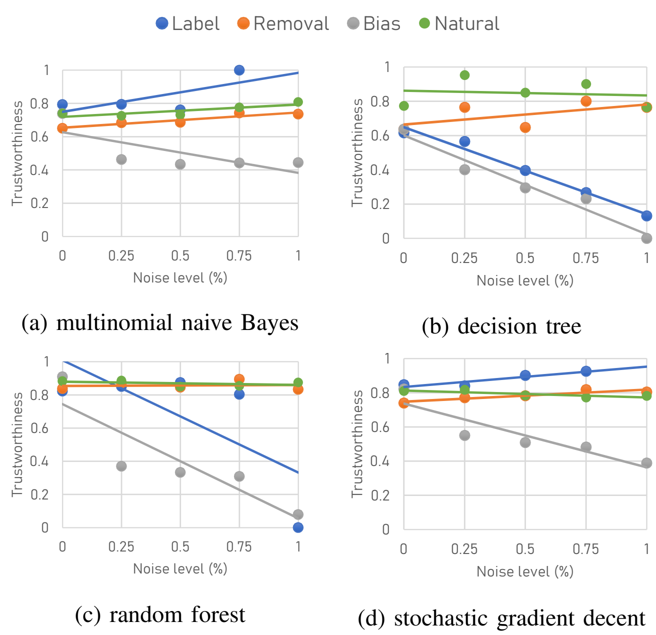
To note, we find that label noise does not evaluate correctly with the adjusted slope, as the result of having 100% label noise is 0% accuracy. This would result in zero instances considered while using the adjusted method. Thus, any analysis method including both adjusted slope and label noise is not considered.

C. RQ2 - Human-based evaluation

We wish to know whether TOWER's trustworthiness predictions align with humans' perception of trustworthiness. To do this, we obtain explanations from various models, using human judgement to determine their trustworthiness, creating a ground truth to compare to. Three authors of the paper performed the labelling.

TABLE II: Slopes for various datasets, model types, and noise types using optimal configuration and $t/(t+u)$ slope calculation. Adjusted slopes used for non-label noise.

Test models		Noise types			
Dataset	Model	Bias	Label	Removal	Natural
20 Newsgroups	MNB	-0.100	-0.176	-0.010	0.012
	DT	-0.428	-0.118	0.220	-0.129
	RF	-0.467	-0.220	-0.014	-0.101
	SGD	-0.246	-0.211	-0.044	-0.050
AG News	MNB	-0.101	-0.109	0.003	-
	DT	-0.400	-0.162	0.052	-
	RF	-0.257	-0.168	-0.025	-
	SGD	-0.315	-0.275	0.016	-
DBpedia14	MNB	-0.162	-0.390	0.023	-
	DT	-0.430	-0.340	-0.024	-
	RF	-0.567		-0.025	-
	SGD	-0.219	-0.425	0.029	-
Emotion	MNB	-0.210	-0.227	-0.009	-
	DT	-0.369	-0.211	0.078	-
	RF	-0.514	-0.320	-0.008	-
	SGD	-0.433	-0.302	-0.012	-
IMDb	MNB	-0.070	-0.030	-0.037	-
	DT	-0.334	-0.052	0.022	-
	RF	-0.233	-0.051	0.117	-
	SGD	-0.321	-0.077	0.014	-



(a) multinomial naive Bayes (b) decision tree

(c) random forest (d) stochastic gradient decent

Fig. 3: Example charts of trustworthiness against noise level for each model using optimal configuration and $t/(t+u)$ slope calculation for the 20 Newsgroup dataset. Adjusted slope used for all except label noise.

The explanations to label are generated through pretrained models fine-tuned on various datasets, chosen by popularity on HuggingFace. These are: 20 Newsgroups, AG News, Amazon Review Polarity, DBpedia14, Emotion, IMDb, Rotten Tomatoes, and Yahoo Answers Topics. Details can be found in Table I.

TABLE III: Human-labelled data.

Dataset	Untrust.	Trust.	Undefined	Total
20 Newsgroups	1	25	0	26
AG News	3	41	2	46
Amazon Reviews Polarity	7	42	2	51
DBpedia14	1	42	0	43
Emotion	0	27	0	27
IMDb	1	45	3	49
Rotten Tomatoes	6	39	6	51
Yahoo Answers Topics	2	32	1	35
Total	21	293	14	328

We chose to use pretrained models as they more closely align with real world application than models trained ourselves. This allows us to test on more realistic instances of trustworthy and untrustworthy explanations. We assume the generalisability of TOWER since it evaluates the explanations independently from the model that generates the predictions.

From each model and dataset, we randomly choose 1,000 instances from each dataset and explanations generated using LIME. We then run TOWER, using the best configuration trained in Section IV-B, on these explanations. Of the 8,000 instances, 501 are randomly chosen with a proportion of $33\% \pm 7.5\%$ between ones predicted as *trustworthy*, *untrustworthy*, and *undefined*. This is done to combat the likely trustworthy skew in the data. The error of $\pm 7.5\%$ is added to prevent potential bias from knowing the exact ratio of explanations.

The labelling procedure is thus: Each person is given an instance and its possible classes. They must predict the class. This is to ensure that they consider the text and not simply give a label based on only the explanation [28]. Next, the explanation is given – the top ten words, highlighting the respective words in the text as well as giving the importance scores. They may choose to label this explanation as trustworthy, untrustworthy or undefined. If the person does not predict the correct class, the instance is discarded; this is not shown to the human.

We have two label each instance. If two authors do not agree on the label, the third author is brought in. If none agree, the instance is discarded. These labelled instances are then compared to the predictions from TOWER to evaluate its effectiveness.

Of the 501 instances, 139 were discarded for a participant incorrectly predicting the category and 15 for a lack of agreement of trustworthiness label. Another 19 were erroneously discarded. This resulted in a final set of 328 instances.

RQ2 Results We evaluated TOWER against *human-based evaluation* (**RQ2**). To do so, we created our own dataset of human-labelled ground truths for trustworthy and untrustworthy explanations. In total, 501 instances were labelled, 328 of which were considered. As seen in Table III, 293 (89%) were labelled *trustworthy*, 21 (6%) *untrustworthy*, and 14 (4%) *undefined*.

Fig. 4 shows the confusion matrix of the optimal configuration of TOWER on the human-labelled ground truth. The precision, recall, and F1 score for labelling as trustworthy is 0.92, 0.40, and 0.56, respectively. For labelling as untrustworthy, these are 0.14, 0.48, and 0.21.

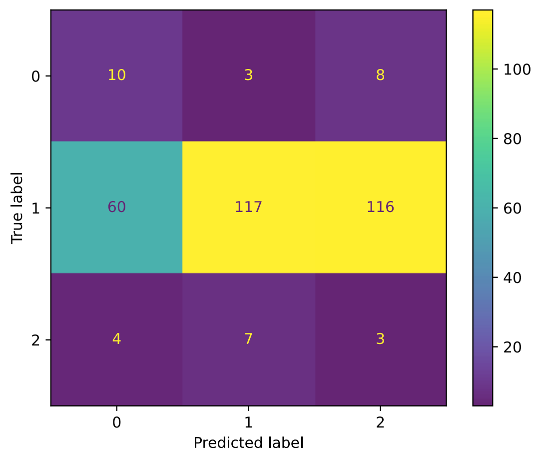


Fig. 4: Confusion matrix. Labels 0, 1, and 2 are trustworthy, untrustworthy, and undefined, respectively.

D. Discussion

From our noise-based evaluation (**RQ1**), we find that there is generally a correlation between TOWER's output and the trustworthiness of a model, supposing that artificially induced noise into the training data of a model is a good approximation of untrustworthiness. We find that bias is better at this approximation than label or removal noise, assuming that TOWER's predictions are accurate. We also see from Table II that TOWER is not able to detect models trained on removal noise very well. This may be because of our specific implementation of the noise, or because machine learning models are often robust to this type of noise [1].

From our human-based evaluation (**RQ2**), however, we find that the introduction of noise may not be a good approximation for untrustworthiness after all. TOWER, using the optimal configuration found from the noise-based evaluation, performed poorly in against human-labelled ground truths of trustworthy explanations. We find that it has a bias against predicting as *trustworthy*. Noise can introduce spurious correlations that can be detected via a semantic measure like TOWER. It is also possible that trustworthy explanations found in our simpler models may be different to that of pretrained models.

We can gain insight by looking at the incorrectly-labelled instances. Of the 328 ground-truth instances, TOWER labels 198 incorrectly. We are more interested in the *trustworthy* and *untrustworthy* instances, so we will disregard for now the 135 instances labelled as *undefined* by either TOWER or humans. Of the 63 remaining, 60 are falsely labelled *trustworthy*. Of the 63, 32 are from the Amazon Polarity, Rotten Tomatoes, and IMDb datasets; these all have the categories 'positive' and 'negative'. For the rest, the most common categories are 'joy', 'sports', 'science', 'sadness', and 'health'. From this, we can intuit that the effectiveness of TOWER is somewhat proportional to how specific the class name is. The more general the class, the harder it is to relate words to it.

This seems to be, in part, due to our limited method of measuring trustworthiness. TOWER only measures the relatedness between the explanation words and the singular class name. To alleviate this, a possible solution is to have the category instead be a collection of words describing the concept in more detail, such as a definition.

Other possible areas of improvement include taking into account words in context (e.g. "not good"), importance scores, unusually impactful singular words, other class names, incorrect model predictions, and the relationship between words in the explanation.

Finally, a major way forward would be to curate a larger and less-skewed dataset so to more effectively evaluate the tool.

V. Threats to Validity

Generalisability of results. One threat of external validity is that the evaluation results dot not generalise for other models and dataset. We mitigate this threat by using various different datasets and models in the evaluation. The noise-based evaluation uses 5 datasets and 4 models, while the human-based evaluation used 8 datasets and models. This reduces the chance of individual datasets or models biasing the results.

Bias of word embedding methods. Word embeddings may have their own bias, errors, and trustworthiness issues. Are we not simply moving trust from the machine learning model to these embedding methods? To alleviate this, an ensemble of methods is used to reduce the impact of errors in any single embedding method.

Human collection of ground truth. Humans may have biases and failings when labelling explanations as trustworthy or not. To alleviate this, two authors of this paper did the labelling independently, and final labels were picked only with majority vote involving a third independent author. The participants were also made to choose a class for the input first, incentivising them to consider the input in its entirety when labelling. In addition, the proportion of potentially trustworthy and untrustworthy instances was somewhat randomised, preventing bias towards labelling for an equal ratio. As the human labellers knew how TOWER worked, they were instructed to label the data using only their human intuition, disregarding any knowledge of the tool.

VI. Related Work

Trustworthiness AI and ML is an important research topic that has gained a lot of attention [11]. To the best of our knowledge, there are no prior attempts to create an automated trustworthiness oracle. TOWER is the first automated trustworthiness technique.

There have been metrics proposed for measuring trustworthiness in machine learning models. Schmidt and Biessmann [27] proposes a metric based on the increase in information transfer rate for humans doing a task – that is, the change in speed and accuracy of human interpreters when assisted by a machine explanation or not. The Trustworthy Explainability Acceptance metric requires industry experts to evaluate the system in its calculations [15], [29]. There is the building of trust probabilities based on subjective logic via uncertainty given by a human observer [7]. RITUAL is a collection of metrics that require humans to manually evaluate [5]. All of these require humans in the loop, however; TOWER aims to evaluate trustworthiness automatically.

The use of explainability in ML testing has also been explored. There have been testing on the effectiveness of individual explanatory techniques [32] and testing of the evaluation of explanations [12], [23]. There has been research to mitigate potential unreliability in the explanations themselves [10], [16]. More relevant to our goal, there have been research on evaluating models based on explanations [27], as mentioned previously, including the original LIME paper itself [25]. These works also require humans in the loop.

In addition, the current work towards the measurement of plausibility of explanations seems to lie mainly within the natural language processing field. They involve procedures such as comparing against human-created ground truths [3], [6], [8], [9], or asking people directly [13]. Again, we have found no fully automated way to do this.

Other related works are techniques on automated fairness testing. They do not rely on word embeddings but use LIME explanations to check whether the predictions were influenced by the presence of sensitive or protected attributes (e.g., gender, ethnicity, age). For example, Aggarwal et al. [2] proposes an automated method for the evaluation of fairness, via the use of the local linear model created by LIME. However, fairness by itself does not constitute a full evaluation of trustworthiness; our technique focuses on evaluating the justification behind the explanations.

The most related work of TOWER is that of Jiang et al. [14], which defines a 'trust score'; a measure of the level of agreement between a classifier and a modified nearest-neighbor classifier for a single response. This is a rather different approach to our proposed technique as it only rely on the agreement between multiple classifiers to measure trustworthiness. Instead, TOWER analyses the decision-making process of ML classifiers to measure trustworthiness.

VII. Conclusion

This paper presented TOWER, one of the first attempts to automatically perform trustworthiness testing. The experiments with noisy data suggests that the idea behind TOWER is indeed relevant in identifying trustworthiness issues. However, the poor results on the human labelled data hint that semantic unrelatedness between explanations and class names might not characterise all types of untrustworthy predictions.

This paper sparks exciting future work in this new research area of automated trustworthiness testing. More research is needed to understand the relationship between explanations and trustworthiness issues.

References

[1] Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. How much noise is too much: A study in automatic text classification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 3–12, 2007.

[2] Aniya Aggarwal, Pranay Kr. Lohia, Seema Nagar, Kuntal Dey, and Diptikalyan Saha. Black box fairness testing of machine learning models. *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019.

[3] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. Interpretable neural predictions with differentiable binary variables. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy, July 2019. Association for Computational Linguistics.

[4] Federico Cabitza, Andrea Campagner, Gianclaudio Malgieri, Chiara Natali, David Schneeberger, Karl Stoeger, and Andreas Holzinger. Quod erat demonstrandum? - towards a typology of the concept of explanation for the design of explainable ai. *Expert Syst. Appl.*, 213(PA), mar 2023.

[5] Alberto Huertas Celdrán, Jan Bauer, Melike Demirci, Joel Leupp, Muriel Figueredo Franco, Pedro M. Sánchez Sánchez, Gérôme Bovet, Gregorio Martínez Pérez, and Burkhard Stiller. Ritual: a platform quantifying the trustworthiness of supervised machine learning. In *2022 18th International Conference on Network and Service Management (CNSM)*, pages 364–366, 2022.

[6] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1448–1458. PMLR, 13–18 Jul 2020.

[7] Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. There is hope after all: Quantifying opinion and trustworthiness in neural networks. *Frontiers in artificial intelligence*, 3:54, 2020.

[8] Miruna-Adriana Clinciu, Arash Eshghi, and Helen Hastie. A study of automatic metrics for the evaluation of natural language explanations. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2376–2387, Online, April 2021. Association for Computational Linguistics.

[9] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.

[10] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. *Explanations can be manipulated and geometry is to blame*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[11] Luciano Floridi. Establishing the rules for building trustworthy ai. *Ethics, Governance, and Policies in Artificial Intelligence*, pages 41–45, 2021.

[12] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, 2018.

[13] Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. Learning to faithfully rationalize by construction. In *Annual Meeting of the Association for Computational Linguistics*, 2020.

[14] Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5546–5557, Red Hook, NY, USA, 2018. Curran Associates Inc.

[15] Davinder Kaur, Suleyman Uslu, Arjan Durresi, Sunil V. Badve, and Murat Dundar. Trustworthy explainability acceptance: A new metric to measure the trustworthiness of interpretable ai medical diagnostic systems. In *Computational Intelligence in Security for Information Systems*, 2021.

[16] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T. Schütt, Sven Dähne, D. Erhan, and Been Kim. The (un)reliability of saliency methods. In *Explainable AI*, 2017.

[17] R. Andrew Kreek and Emilia Apostolova. Training and prediction data discrepancies: Challenges of text classification with noisy, historical data. In Wei Xu, Alan Ritter, Tim Baldwin, and Afshin Rahimi, editors, *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 104–109, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[18] Lena Kästner, Markus Langer, Veronika Lazar, Astrid Schomäcker, Timo Speith, and Sarah Sterz. On the relation of trust and explainability: Why to engineer for trustworthiness. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 169–175, 2021.

[19] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy ai: From principles to practices. *ACM Comput. Surv.*, 55(9), jan 2023.

[20] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, jun 2018.

[21] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41, 1995.

[22] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[23] Sina Mohseni, Jeremy E Block, and Eric Ragan. Quantitative evaluation of machine learning explanations: A human-grounded benchmark. In *Proceedings of the 26th International Conference on Intelligent User Interfaces*, IUI '21, page 22–31, New York, NY, USA, 2021. Association for Computing Machinery.

[24] Dong Nguyen. Comparing automatic and human evaluation of local explanations for text classification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1069–1078, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[25] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938, 2016.

[26] Waddah Saeed and Christian Omlin. Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems*, 263:110273, 2023.

[27] Philipp Schmidt and Felix Biessmann. Quantifying interpretability and trust in machine learning systems. *ArXiv*, abs/1901.08558, 2019.

[28] Terne Sasha Thorn Jakobsen, Laura Cabello, and Anders Søgaard. Being right for whose right reasons? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1033–1054, Toronto, Canada, July 2023. Association for Computational Linguistics.

[29] Suleyman Uslu, Davinder Kaur, Samuel J. Rivera, Arjan Durresi, Mimoza Durresi, and Meghna Babbar-Sebens. Trustworthy acceptance: A new metric for trustworthy artificial intelligence used in decision making in food-energy-water sectors. In *International Conference on Advanced Information Networking and Applications*, 2021.

[30] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

[31] Jeannette M. Wing. Trustworthy ai. *Commun. ACM*, 64(10):64–71, sep 2021.

[32] Jianlong Zhou, Amir Hossein Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 2021.

[33] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: a quantitative study of their impacts. *Artif. Intell. Rev.*, 22(3):177–210, nov 2004.