Robust Interpretable Text Classification against Spurious Correlations Using AND-rules with Negation

Rohan Kumar Yadav , **Lei Jiao** , **Ole-Christoffer Granmo** and **Morten Goodwin**
Centre for Artificial Intelligence Research, University of Agder, 4879, Grimstad, Norway
{rohan.k.yadav, lei.jiao, ole.granmo, morten.goodwin}@uia.no

Abstract

The state-of-the-art natural language processing models have raised the bar for excellent performance on a variety of tasks in recent years. However, concerns are rising over their primitive sensitivity to distribution biases that reside in the training and testing data. This issue hugely impacts the performance of the models when exposed to out-of-distribution and counterfactual data. The root cause seems to be that many machine learning models are prone to learn the shortcuts, modelling simple correlations rather than more fundamental and general relationships. As a result, such text classifiers tend to perform poorly when a human makes minor modifications to the data, which raises questions regarding their robustness. In this paper, we employ a rule-based architecture called Tsetlin Machine (TM) that learns both simple and complex correlations by ANDing features and their negations. As such, it generates explainable AND-rules using negated and non-negated reasoning. Here, we explore how non-negated reasoning can be more prone to distribution biases than negated reasoning. We further leverage this finding by adapting the TM architecture to mainly perform negated reasoning using the specificity parameter s. As a result, the AND-rules becomes robust to spurious correlations and can also correctly predict counterfactual data. Our empirical investigation of the model's robustness uses the specificity s to control the degree of negated reasoning. Experiments on publicly available Counterfactually-Augmented Data demonstrate that the negated clauses are robust to spurious correlations and outperform Naive Bayes, SVM, and Bi-LSTM by up to 20%, and ELMo by almost 6% on counterfactual test data.

1 Introduction

Despite impressive advances of Deep Neural Network (DNN) architectures for Natural Language Processing (NLP), their implementations still suffer from various challenges. One of the challenges is associated with DNN's capability of learning simple correlations and ignoring more complex ones [Sauer and Geiger, 2021]. This behavior of DNN becomes questionable when the simple correlation is spurious and absent from the test data, or occurs in an unfitting context. For instance, in the sentence *Nolan's films are always great mostly because of his excellent direction*, the influential word for predicting a positive sentiment should be "great" and "excellent" instead of "Nolan's" and "direction". However, due to the majority of samples consist of "Nolan having a great movie", it makes the classifier learn that "Nolan" corresponds to a positive sentiment word [Wang and Culotta, 2020]. Similarly, a toxicity classifier learns that "gay" corresponds to a toxic comments [Wulczyn *et al.*, 2017] and a medical diagnosis classification system learns the disease associated with the patient ID [Kaufman *et al.*, 2012] The issue of spurious patterns also moderately impacts the out-of-distribution (OOD) generalization of models that are trained on independent identical distribution (IID) data, resulting in performance degradation when the distribution shifts.

Researchers recently have found that the decay in model performance, as well as social bias in NLP, appear out-of-domain due to sensitivity towards spurious signals. One of the solutions to deal with such vulnerability in NLP models is data augmentation with counterfactual samples [Kaushik *et al.*, 2020], which can help the model with learning real causal correlations between input and labels. For instance, a man-made counterfactual sample of the last example could be *Nolan's films are always* ***boring*** *mostly because of his* ***poor*** *direction.* Inserting such counterfactual data into the original training sets has shown to be beneficial for learning real causal correlation thereby improving the robustness of the model [Kaushik *et al.*, 2020]. However, augmentation with counterfactual data usually relies on a human-in-the-loop system to generate sentiment-flipped samples. For this process, humans are asked to make minimal and believable edits to generate counterfactual samples. Even though such an addition of data makes the model robust against spurious correlations, completing a human-in-the-loop process is costly and time-consuming.

The main reason behind the failure of DNNs on counterfactual data during inference is still unclear because of their black-box nature [Rudin, 2018]. What they learn from the data that limits the models' robustness against different distribution samples is currently an open research question. Some researchers argue that the attention mecha-

nism provides an explanation of DNN models, which assigns soft weights to the input representations, and then extracts highly weighted tokens as rationales [Bahdanau *et al.*, 2015]. However, these attention weights do not provide faithful explanations for classification [Serrano and Smith, 2019; Brunner *et al.*, 2020]. In addition, DNNs fail to conduct logical reasoning in various tasks. Logical reasoning is one of the most important prerequisites in NLP that supports various practical applications such as legal assistants, medical decision support, and personalized recommender systems. Due to these issues, DNNs have failed to demonstrate their robustness on counterfactual data. On the other hand, a rule-based knowledge system is a powerful tool that offers logical reasoning because of its explainability. However, most rule-based systems rely on static rules that are hand-crafted. Without learning capability, the performance and generalization is limited. Keeping these two challenges in consideration, we employ a recent architecture called Tsetlin machine (TM), which is an interpretable rule-based model that learns both simple and complex correlations via conjunctive clauses [Granmo, 2018]. Unlike DNNs and simple rule-based architectures, TM learns rules with logical reasoning as a human does and it also offers a transparent and interpretable learning [Yadav *et al.*, 2021a]. Such rule-based logical reasoning is important for processing counterfactual data, and we here study the TM robustness towards counterfactual test data compared with DNNs. In addition, we demonstrate how TM supports reasoning with negation, which is completely different from attention-based DNN models. The main contributions of the paper are as follows:

- We design a TM-based approach that is robust to spurious correlations on counterfactual and out-of-domain data, without any data augmentation.
- Our in-depth analysis of TM specificity parameter s records transparent learning and explainable predictions on counterfactual data.

2 Related Work

Many recent papers have shown that the DNN-based NLP models do not seem to learn the aspects that humans seem important for a particular classification. The state-of-the-art models have been vulnerable to fabricated transformations. Effective fabrications include distractor phrases, adversarial example generation with paraphrasing, and template-based modifications. As the result, researchers develop counterfactual data augmentation approaches for building robust classifiers [Lu *et al.*, 2020] to eliminate the effect of spurious correlations. In attempt to augment counterfactual data, [Kaushik *et al.*, 2020] develops a human-in-the-loop system using crowd-sourcing methods [Kaushik *et al.*, 2020]. It is shown that such augmented data makes the trained model robust not only to counterfactual data but also to out-of-domain datasets. However, due to the fact that crowd-sourcing is time-consuming and expensive, an automatic augmentation of counterfactual data with causal identification is proposed [Wang and Culotta, 2020]. Here, causal words are employed to generate counterfactual data using BERT sentence similarity.

Apart from data augmentation, there has been little research on studying the reason for the failure of ML models for counterfactual data. Perception and reasoning are two crucial abilities a model needs for successful problem-solving. Recent ML models such as DNNs have shown extraordinary performance in various perception tasks [Krizhevsky *et al.*, 2012]. However, such models hardly exploit refined domain knowledge in symbolic form in order to support reasoning. Despite the recent DNNs' ability to consider relational and differential knowledge representation, they still lack comprehensive logical reasoning across the dataset [Jia and Liang, 2017]. Hence, there has been an increasing interest in combining ML with logical reasoning especially in the field of NLP. For instance, Fuzzy Logic [Goguen, 1973], Statistical Relational Learning [Getoor and Taskar, 2007], and Probabilistic Logic Programming [Raedt and Kimmig, 2015] have come into the picture to enhance traditional logic-based methods. However, they often require handcrafted symbols as input from humans. There has been little research on alternative ML models that have a learning ability comparable to DNN and also possess human-like logical reasoning.

Since counterfactual inference is all about understanding the logical reasoning behind the training data, we here incorporate a recent ML model, TM [Granmo, 2018], which not only offers a transparent learning mechanism but also facilitates human-level model interpretation [Yadav *et al.*, 2021a]. Unlike DNN attention weights that arguably explain predictions after training, the TM learning process itself is fully transparent and produces logically explainable prediction [Yadav *et al.*, 2021b]. Indeed, TM has been widely accepted for its interpretability and logical reasoning [Lei *et al.*, 2021; Abeyrathna *et al.*, 2021]. Hence, in this paper, we propose and investigate how TM can deal with counterfactual test data as well as out-of-domain distributions. We further show that it produces rules as human-like logical reasoning and is more robust than DNNs. In particular, we explore its interpretability in-depth contrasting the logic-based rules with the attention weights utilized by DNNs. Thereby, we analyse the reasons that explain TM's robustness against spurious correlations. To our knowledge, this is the first time that a rule-based human-level interpretable model is used to tackle spurious correlations.

3 Detailed Implementation

3.1 Tsetlin Machine

TM is a recent ML model that learns correlations between features and labels using propositional logic [Granmo, 2018]. A propositional logic formula in TM, namely a clause, is a conjunction of negated and non-negated forms of the input features. The negated or non-negated forms of the input features are known as literals and are controlled by a set of Tsetlin Automata (TA). In a simple way, each input feature corresponds to two TAs, i.e., *TA* and *TA'*. *TA* controls the original (non-negated) form of the literal whereas *TA'* controls its negation. Each TA decides either to include or exclude the literal, and has two actions (Include/Exclude) with $2N$ states. When a TA moves from state 1 to N, action Exclude is performed. When a TA moves from state $N + 1$

to $2N$, it performs the Include action. Each move of TA is triggered by feedback in the form of Reward, Penalty, or Inaction [Granmo, 2018].

The most important component of TM is the clause, which represents a certain sub-pattern among a particular set of patterns. This sub-pattern is in propositional AND-form making it highly interpretable and amendable for logical understanding of the task. To have a clear comprehension of what a clause looks like, let us consider a bag-of-words input $X = [x_1, \cdots, x_n]$, $x_k \in \{0, 1\}$, $k \in \{1, \ldots, n\}$ where $x_k = 1$ means the presence of a word in the sentence and n is the size of the vocabulary. Let us assume there are γ classes in total. If each class needs α clauses to learn the pattern, altogether the model is represented by $\gamma \times \alpha$ clauses C_ι^κ, $1 \leq \kappa \leq \gamma, 1 \leq \iota \leq \alpha$, as:

$$C_\iota^\kappa = \left(\bigwedge_{k \in I_\iota^\kappa} x_k \right) \wedge \left(\bigwedge_{k \in \bar{I}_\iota^\kappa} \neg x_k \right), \tag{1}$$

where I_ι^κ and $\bar{I}_\iota^\kappa$ are non-overlapping subsets of the input variable indices, $I_\kappa^\iota, \bar{I}_\kappa^\iota \subseteq \{1, \cdots, n\}, I_\kappa^\iota \cap \bar{I}_\kappa^\iota = \emptyset$. I_ι^κ represents the set of indices of the features that the TAs include in original form, while the set $\bar{I}_\iota^\kappa$ contains the indices of the features that the TAs include in negated form.

Here, clauses with odd indexes in each class are allocated positive polarity (+), whereas those with even indexes are assigned negative polarity (-). Positive polarity clauses vote in favor of the target class, while negative polarity clauses vote against it. As demonstrated in Eq. (2), a summation operator aggregates them by subtracting the total number of negative votes from the total number of positive votes.

$$f^\kappa(X) = \Sigma_{\iota=1,3,\ldots}^{\alpha-1} C_\iota^\kappa(X) - \Sigma_{\iota=2,4,\ldots}^{\alpha} C_\iota^\kappa(X). \tag{2}$$

For γ classes, the final output $\hat{y}$ is given by the argmax operator to classify the input based on the highest net sum of votes, as shown in Eq. (3).

$$\hat{y} = \text{argmax}_\kappa \left(f^\kappa(X) \right). \tag{3}$$

3.2 Learning Rule-based Clauses for Counterfactual Inference

The step-by-step explanation for the learning process of TM can be found in [Yadav *et al.*, 2021b]. Here we explain briefly the learning of the rule-based clauses in TM for counterfactual inference via an example. Let the sentence "*Long, boring, blasphemous. Never have I been so glad to see ending credits roll.*" be the training sample that has negative sentiment. Each of the input words in the sentence is controlled by two TAs where *TA* controls non-negated literal such as "Long", and *TA'* controls the negated form such as "¬Long". The input that represents this particular sample is a sparse Boolean bag-of-words. All the vocabulary words that are present in the given sentence get the truth value 1, while those absent get the truth value 0. By explicitly representing missing words in vector form like $[0, 0, 0, 1, 0, \cdots, 0, 1, 0, 0, 0, 1]$, the representation becomes dense. However, logically, such representation not only captures the presence of a particular word, but also equally well represents those words that are *not* present. This explicit bag-of-words representation is ideal for TMs. This is because the TM can then pick informative negated features in the very first hundred iterations of learning using the selection parameter specificity (s). We detail the role of s next.

Input	Clause	1		0	
	Literal	1	0	1	0
Include Literal	P(Reward)	$\frac{s-1}{s}$	NA	0	0
	P(Inaction)	$\frac{1}{s}$	NA	$\frac{s-1}{s}$	$\frac{s-1}{s}$
	P(Penalty)	0	NA	$\frac{1}{s}$	$\frac{1}{s}$
Exclude Literal	P(Reward)	0	$\frac{1}{s}$	$\frac{1}{s}$	$\frac{1}{s}$
	P(Inaction)	$\frac{1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$	$\frac{s-1}{s}$
	P(Penalty)	$\frac{s-1}{s}$	0	0	0

Table 1: The Type I Feedback.

Input	Clause	1		0	
	Literal	1	0	1	0
Include Literal	P(Reward)	0	NA	0	0
	P(Inaction)	1.0	NA	1.0	1.0
	P(Penalty)	0	NA	0	0
Exclude Literal	P(Reward)	0	0	0	0
	P(Inaction)	1.0	0	1.0	1.0
	P(Penalty)	0	1.0	0	0

Table 2: The Type II Feedback.

In TM, each TA that controls a literal decides the action "Include" or "Exclude" based on the feedback it receives. There are two types of feedback: Type I Feedback and Type II Feedback, shown in Tables 1 and 2 [Granmo *et al.*, 2019]. Type I Feedback is activated when a given input feature is either correctly assigned to the target label (true positive) or mistakenly ignored (false negative), while Type II Feedback is activated when an input feature is wrongly assigned to the target label (false positive). From Tables 1 and 2 we can see that parameter s, $s \geq 1$, plays a very important role in the learning process, as it controls how strongly the model favours the action "Include". It also determines how many "fine-grained" sub-patterns the clauses will acquire. The greater the value of s, the more the TAs are encouraged to include literals in their clauses. Since s decides which literals take part in the clause for classification, it is vital to fine-tune it for reducing the vulnerability against spurious correlation. For the above-mentioned training example, when s is large, the states for the corresponding TAs in a clause after training are shown in Fig. 1. As seen, the high value of s enforces TA to include many literals in the clause, such as including "ending", "boring", "credits", "¬friendly", "¬good", and "¬like". Among the included literals, spurious correlations that do not carry sentiment information, such as "ending" and "credits", indeed influence the model's prediction on counterfactual data.

When we have a small s, as shown in Fig. 2, the number of included literals is reduced and the majority, if not all, of the included literals are in the negated form. One can see from

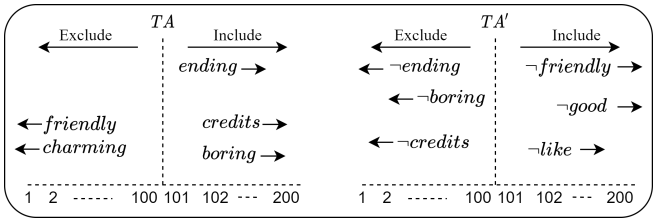


Figure 1: States of TAs when s is high for a particular clause.

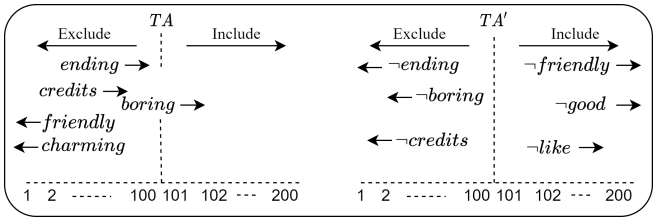


Figure 2: States of TAs when s is low for a particular clause.

the figure that the non-negated literals are now not enforced to be included in the clause. The states in *TA* for "ending", "boring", and "credits" have not reached to action "Include". Nevertheless, TM still learns negated features easily in contrast to non-negated features due to sparse input representation thereby not affecting the states of "¬friendly", "¬good", and "¬like".

3.3 Robustness against Counterfactual Sample

In this subsection, we will detail the reason why a trained TM model is robust and insusceptible to spurious correlations. Let us consider a model trained with a low value of $s = 2$ and two sentences with different sentiment labels: S_1 with positive and S_2 with negative sentiment. From Fig. 3, we can see the behavior of trained clauses for the negative class and the positive class for the original samples. The rule-based logic that is formulated by TM is in propositional form, ANDing several literals. The clause associated with propositional logic becomes 1 if an input satisfies the conjunction.

When context S_1 is received by the model, it correctly predicts negative sentiment because it triggers all the five clauses in the negative class, whereas only one clause for the positive class. Similarly, when S_2 is given, it predicts positive sentiment because the input triggers all five clauses in the positive class compared to only one clauses in the negative class.

Now consider two human generated counterfactual samples S_1^{cf} for S_1 and S_2^{cf} for S_2 as shown in Fig. 4. For S_1, the word "boring" is replaced by "fascinating"; "blasphemous" is replaced by "soulful"; and "glad" is replaced by "sad". Similarly, for S_2, the word "friendly" is replaced by "depressing"; "charming" is replaced by "charmless"; and "unpretentious" is replaced by "pretentious". This means that the labels for the corresponding counterfactual samples are now flipped. When S_1^{cf} is sent to the trained TM model with $s = 2$, it only triggers two clauses from the negative class and three clauses in the positive class. Similarly, when S_2^{cf} is given to the model, it triggers four clauses in the negative class but only one clause in the positive class. Even though the probability of being in a class decreases due to the reduction in clause score, it still manages to predict such counterfactual samples correctly.

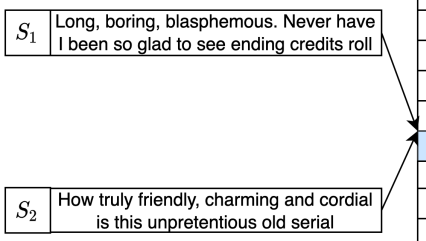
Since most of the entries in the sparse bag-of-word representation are zeros, the majority of literals presented in the clause will be in the negated form after a few iterations. With a comparatively small number of included literals due to the small s, the majority of clauses most likely becomes monotone in the negated form. Negated literals provide a more general form of the features that are not presented in a particular input sample thereby being less sensitive to spurious correlations as compared with the non-negated literals. We can clearly observe from Fig. 4 that the non-monotone clauses that have non-negated features are the ones that fail to capture counterfactual reasoning. This means monotonous clauses that have only negated features are more insusceptible to such modified data.

4 Experiments and Results

In this section, we present experimental results for analyzing the performance of TM on counterfactual data. As we have already discussed the significance of s for inheriting robustness in the model, we experiment with different values of s on the dataset designed by [Kaushik *et al.*, 2020] [1]. This dataset has been developed using IMDB reviews that consist of $50k$ samples divided equally across train/test splits after removing 20% of reviews. Among them, $2.5k$ reviews have been split into training, validation, and testing of 1707, 245, and 488 respectively. These reviews are modified using Amazon's Mechanical Turk crowdsourcing so that the labels are flipped to generate counterfactual samples. In addition, to evaluate the out-of-domain performance of the proposed model, we used Amazon reviews [Ni *et al.*, 2019] on data aggregated over six domains, i.e., *beauty, fashion, appliances, gift cards, magazines, and software*, SemEval Twitter sentiment analysis [Rosenthal *et al.*, 2017], and Yelp challenge dataset.

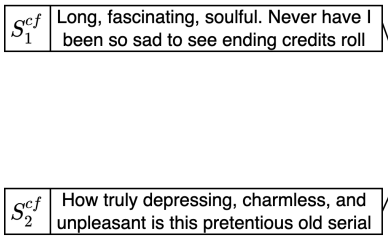
We used the original $1.7k$ samples as the training dataset to evaluate the robustness of the model on human-generated counterfactual test data of size 488. We also train the model using counterfactual data of size $1.7k$ and evaluate it on the original test samples of size 488. The performance of the model for various values of s is shown in Table 3. Other parameters of TM are the same for all the training datasets selected in the paper, with 3000 clauses per class and the threshold (T) value of 80×16. These parameters are selected by trial and error. For evaluating the behavior of s, we only validate on the test samples that are not from the same training data, and the complete performance evaluation is detailed later in the paper. Here, we use the features extension technique as the preprocessing as in [Yadav *et al.*, 2021a]. As seen in Table 3, the accuracy of the model trained on original training samples achieves 72.1% on counterfactual test data when $s = 2$, and it decreases as s increases. Similarly, the accuracy of the model trained on counterfactual training samples achieves 65.20% when $s = 2$ and decreases as s increases. This indicates that lowering the value of s fine grains

[1] https://github.com/acmi-lab/counterfactually-augmented-data



Negative Class Clauses	S_1	S_2
$C_1 = ending \wedge boring \wedge \neg good \wedge \neg interesting \wedge \neg like, \cdots$	$C_1 = 1$	$C_1 = 0$
$C_3 = \neg fascinating \wedge \neg interesting \wedge \neg like, \cdots$	$C_3 = 1$	$C_3 = 1$
$C_5 = \neg good \wedge \neg friendly \wedge \neg like, \cdots$	$C_5 = 1$	$C_5 = 0$
$C_7 = \neg friendly \wedge \neg charming \wedge \neg fascinating, \cdots$	$C_7 = 1$	$C_7 = 0$
$C_9 = \neg excellent \wedge \neg good \wedge \neg like, \cdots$	$C_9 = 1$	$C_9 = 1$
Positive Class Clauses	S_2	S_1
$C_1 = truly \wedge \neg depressing \wedge \neg long \wedge \neg worst, \cdots$	$C_1 = 1$	$C_1 = 0$
$C_3 = \neg bad \wedge \neg uninteresting \wedge \neg boring, \cdots$	$C_3 = 1$	$C_3 = 0$
$C_5 = \neg boring \wedge \neg worst \wedge \neg pretentious, \cdots$	$C_5 = 1$	$C_5 = 0$
$C_7 = \neg bad \wedge \neg charmeless \wedge \neg regret, \cdots$	$C_7 = 1$	$C_7 = 1$
$C_9 = serial \wedge \neg worse \wedge \neg unpleasent \wedge \neg like, \cdots$	$C_9 = 1$	$C_9 = 0$

Figure 3: Clause triggered by original samples S_1 and S_2 on both classes when $s = 2$.



Negative Class Clauses	S_1^{cf}	S_2^{cf}
$C_1 = ending \wedge boring \wedge \neg good \wedge \neg interesting \wedge \neg like, \cdots$	$C_1 = 0$	$C_1 = 0$
$C_3 = \neg fascinating \wedge \neg interesting \wedge \neg like, \cdots$	$C_3 = 0$	$C_3 = 1$
$C_5 = \neg good \wedge \neg friendly \wedge \neg like, \cdots$	$C_5 = 1$	$C_5 = 1$
$C_7 = \neg friendly \wedge \neg charming \wedge \neg fascinating, \cdots$	$C_7 = 0$	$C_7 = 1$
$C_9 = \neg excellent \wedge \neg good \wedge \neg like, \cdots$	$C_9 = 1$	$C_9 = 1$
Positive Class Clauses	S_2^{cf}	S_1^{cf}
$C_1 = truly \wedge \neg depressing \wedge \neg long \wedge \neg worst, \cdots$	$C_1 = 0$	$C_1 = 0$
$C_3 = \neg bad \wedge \neg uninteresting \wedge \neg boring, \cdots$	$C_3 = 1$	$C_3 = 1$
$C_5 = \neg boring \wedge \neg worst \wedge \neg pretentious, \cdots$	$C_5 = 0$	$C_5 = 1$
$C_7 = \neg bad \wedge \neg charmeless \wedge \neg regret, \cdots$	$C_7 = 0$	$C_7 = 1$
$C_9 = serial \wedge \neg worse \wedge \neg unpleasent \wedge \neg like, \cdots$	$C_9 = 0$	$C_9 = 0$

Figure 4: Clauses triggered by counterfactual samples S_1^{cf} and S_2^{cf} on both classes when $s = 2$.

the pattern in the clause with negated literals, which confirms the robustness against counterfactual data as discussed earlier.

To compare the performance of our model with the state of the art, extensive experiments have been carried out. Since $s = 2$ performs the best against counterfactual samples, we utilize this value for performance comparison. In addition to DNN based models, we also include typical interpretable linear models in our comparison. The models are mainly taken from [Kaushik *et al.*, 2020], as: •**Standard Methods:** We train linear standard model such as SVM and Naive Bayes (NB) for sentiment classification using "scikit-learn" [Kim *et al.*, 2016]. •**Bi-LSTM:** For training Bi-LSTM, Kaushik et al. [Kaushik *et al.*, 2020] restricted vocabulary of $20k$, replacing out-of-vocabulary as *UNK* tokens. The model consists of bidirectional LSTM with hidden dimension 50, recurrent dropout 0.5, and global max pooling following the embedding layer. •**ELMo:** Kaushik et al. [Kaushik *et al.*, 2020] computed contextualized word representation (ELMo) using character based word representation and bidirection LSTM [Peters *et al.*, 2018] using weighted sum of representation of 1024 dimensions. •**BERT:** Kaushik et al. [Kaushik *et al.*, 2020] used an off-the-shelf uncased BERT Base model to fine tune each task. In order to consider the BERT's sub tokenization, token length is set at 350 and trained for 20 epochs.

As we can see from Table 4, when the original data is used as the training samples, SVM's accuracy on CF test data drops to 51.0% compared with that of the original test data, i.e., 80%. A similar trend is observed for NB, Bi-LSTM, and ELMo. Interestingly, the performance of BERT suffers less perhaps due to the benefit of large pretrained information. However, disregarding the pre-trained language model of BERT, our proposed TM reaches 73.56% and outperforms all of the remaining models including 66.7% of ELMo. In the case of CF data as the training samples, the accuracy on original test samples by previous best model ELMo is 63.8% except BERT. Again, our proposed TM model outperforms all of them except BERT, achieving 65.98%. Although the main aim of the paper is to evaluate TM on different/counterfactual distribution and it is not necessary to augment both original and CF data, we still show the performance using augmented data as well as the remaining IMDB data of size $19k$ as training samples, and it can be seen that the performance of TM is on par with the other models.

Here, we demonstrate the performance of various models trained using original and counterfactual data on out-of-domain balanced test data, as shown in Table 5. For a comparison, we again use preprocessing for feature extension from Glove embedding as in [Yadav *et al.*, 2021a]. The results of other models such as SVM, NB, Bi-LSTM, ELMo and BERT have been taken from [Kaushik *et al.*, 2020]. Here when original data is used as the training sample, understandably, BERT outperforms the other models in all the cases because

Training Data	s = 2		s = 3		s = 5		s = 10		s = 15		s = 20		s = 30		s = 50	
	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF
Orig ($1.7k$)	-	**72.1**	-	71.1	-	68.87	-	65.53	-	64.73	-	60.64	-	58.63	-	54.31
CF ($1.7k$)	**65.20**	-	63.92	-	62.45	-	62.92	-	61.01	-	59.01	-	57.70	-	54.27	-

Table 3: Accuracy of TM on Counterfactual (CF) test data using Original (Orig) training samples and vice-versa for various values of s.

Training Data	SVM		NB		ELMo		Bi-LSTM		BERT		TM	
	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF	Orig	CF
Orig ($1.7k$)	80.0	51.0	74.9	47.3	81.9	66.7	79.3	55.7	87.4	82.2	85.65 (84.30 ± 0.78)	73.56 (72.1 ± 0.40)
CF ($1.7k$)	58.3	91.2	50.9	88.7	63.8	82.0	62.5	89.1	80.4	90.8	65.98 (65.20 ± 0.80)	92.20 (91.09 ± 0.55)
Orig ($19k$)	87.8	60.9	84.3	42.8	86.5	64.3	86.3	68.0	93.2	88.3	88.14 (87.94 ± 0.16)	73.77 (72.46 ± 0.70)
Orig + CF ($3.4k$)	83.7	87.3	86.1	91.2	85.0	92.0	81.5	92.0	88.5	95.1	84.22 (83.45 ± 0.42)	91.2 (89.95 ± 0.75)

Table 4: Experiment results of various models trained using Original and Counterfactual training dataset on their respective opposite test data. The upper results show the best reproducible accuracy and lower ones represent the mean and standard deviation of the last 50 epochs when running the model for five times.

Training Data	SVM	NB	ELMo	Bi-LSTM	BERT	TM
Accuracy on Amazon Reviews						
Orig ($1.7k$)	74.7	66.9	79.1	65.9	80.0	76.2
Orig + CF ($3.4k$)	77.1	82.6	78.4	82.7	85.1	78.5
Accuracy on Semeval 2017 (Twitter)						
Orig ($1.7k$)	61.2	64.6	69.5	55.3	79.3	65.2
Orig + CF ($3.4k$)	66.5	73.9	70.0	68.7	82.9	66.2
Accuracy on Yelp Reviews						
Orig ($1.7k$)	81.8	77.5	82.0	78.0	85.3	82.5
Orig + CF ($3.4k$)	87.6	89.6	87.2	86.2	89.4	85.7

Table 5: Results on out-of-domain balanced test data.

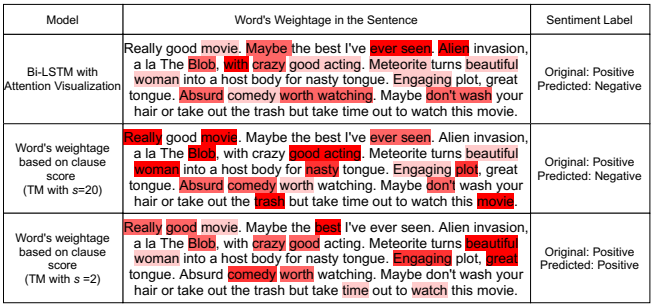


Figure 5: Visualization of words' weightages of attention based model vs TM on a counterfactual sample.

of its access to huge data and better language understanding than traditional models. Disregarding BERT and ELmo as these have huge pretrained information, TM outperforms all the cases of out-of-domain datasets when trained on only original $1.7k$ samples as intended. However, when the CF data is added to the original training sample, the performance of all the models increases by a big margin but the change for TM is not very significant compared with other models. This results in some lower accuracy compared to SVM and NB in Semeval and Yelp reviews. This is because TM has been initialized with a low value of $s = 2$ and most of the features in the clauses are generally in the negated form. For this reason, TM is already less sensitive to spurious correlations and the additional CF training data does not impact much. Hence, only with original training sample of $1.7k$, TM outperforms all the previous model combating spurious correlations.

5 Global Interpretation of TM on Spurious Correlation

The fundamental of TM is the clauses in propositional logic that learn the sub-pattern for a particular task. Hence, TM can be accessed to have multiple forms of interpretation. Generally, two forms of interpretation are highly accepted, namely Global Interpretation and Local Interpretation.

- **Global Interpretation:** From Eq. (2), we can obtain the clause score for each feature of a trained model. This clause score provides the weightage of each input on the model. It has been employed in many applications such as word scoring mechanism [Bhattarai *et al.*, 2022] and novelty detection [Bhattarai *et al.*, 2020].
- **Local Interpretation:** Local interpretation is achieved by analyzing the form of propositional logic. It can be observed by visualizing each clause that votes for the particular input. This interpretation is an important feature of TM since it offers rule-based logic to explain the prediction, such as for sentiment analysis [Yadav *et al.*, 2021b]. This is explained in detail in Section 3.

Since we have already shown the local interpretation previously in Figs. 3 and 4 when learning is introduced, here we demonstrate the benefit of our approach against the spurious correlation using the global interpretation of the model. We examine how the weightage of each feature changes as a function of s. In more detail, we calculate the clause score of each word in the selected vocabulary via Eq. (2). Since the vocabulary is huge, for ease of illustration, we select the highest weighted 20 words for illustration. When we train the model using $s = 20$, words such as *bad, worst, horror, 1, and 2* are the most important features that represent the negative class as shown in Table 6. However, there are certain words such as *minutes, money, instead, reason, and plot* that

Negative Class		Positive Class	
Words	Clause Score	Words	Clause Score
bad	1218	great	2483
worst	560	wonderful	2052
horror	344	romantic	2038
terrible	229	excellent	1951
boring	233	perfect	1913
awful	281	love	1910
waste	310	family	1883
poor	494	young	1863
worse	609	best	1834
stupid	644	beautiful	1801
horrible	660	especially	1788
1	705	enjoyed	1768
minutes	769	loved	1748
money	769	lives	1736
instead	821	life	1734
reason	831	performances	1710
poorly	844	highly	1710
plot	864	performance	1708
dull	876	amazing	1705
2	881	gives	1697

Table 6: Clause scores (weightages) of top 20 words for each sentiment class when the model is trained using $s = 20$.

Negative Class		Positive Class	
Words	Clause Score	Words	Clause Score
worst	2057	romantic	1875
horror	2027	perfect	1418
terrible	1518	wonderful	1335
waste	1515	excellent	1057
awful	1443	enjoyed	948
bad	1254	romance	874
boring	1254	great	871
worse	1131	loved	856
poor	1077	favorite	743
poorly	834	heart	683
horrible	802	8	671
1	796	lives	660
stupid	791	beautiful	657
pointless	685	recommended	657
pathetic	580	wonderfully	645
effort	566	highly	642
dull	562	feelings	641
badly	522	drama	617
lacks	515	amazing	611
money	487	performances	586

Table 7: Clause scores (weightages) of top 20 words for each sentiment class when the model is trained using $s = 2$.

do not necessarily carry the sentiment of the context but are still highly correlated to negative sentiment. Similarly, words such as *family, young, performances, especially, lives, life, and gives* that do not carry positive sentiment are present as highest weighted words in positive class. On the other hand, when the model is trained using $s = 2$, we can see that such words are removed from the given list as shown in Table 7. Table 8 shows the detail of genuine and spurious correlation for two different values of s for the top 50 words. It clearly shows that the number of spurious correlations reduces significantly, making the model robust.

6 A Case Study of TM vs Bi-LSTM

In this section, we will compare the weightage of each word in the sentence responsible for a particular prediction. We here visualize the attention weight to explain the model's prediction. For TM, we use the clause score for each word in the sentence and visualize it in a similar way to the attention model. To have a clear interpretation of how s impacts the counterfactual data, we represent two scenarios where the model is trained with $s = 2$ and $s = 20$. From Fig. 5, we can see that a particular sample has been predicted incorrectly by the Bi-LSTM model. The scoring of the word shows that Bi-LSTM assigns the highest weightage to spurious correlations such as *ever, seen, Alien, Blob, with, and wash*. Although it gives attention to some genuine correlations such as *crazy, worth, and beautiful*, the weightage is low compared with spurious correlations thereby making a wrong prediction. For TM with $s = 20$, it has high clause scores on spurious correlations such as *really, movie, acting, plot, and woman*. Although the TM assigns weightage to words such as *beautiful, good, and engaging*, the weightage for negative sentiment words such as *don't, absurd, and nasty* are much higher thereby predicting it incorrectly to negative sentiment. On the other hand, for TM with $s = 2$, it assigns high scores to genuine correlations such as *beautiful, engaging, best, good, comedy and great* as compared with spurious correlations such as *movie, woman, Blob, time, and watch* thereby correctly predicting a positive sentiment.

	s=20		s=2	
Sentiment Label	Genuine Correlation	Spurious Correlation	Genuine Correlation	Spurious Correlation
Negative	30	20	39	11
Positive	22	18	41	9

Table 8: Statistics of genuine and spurious correlation into 50 words for different s values for each sentiment label.

7 Conclusion

In this paper, we employ TM to design a robust text classification against spurious correlations. TM learns the pattern using a set of clauses that are in the form of propositional logic. Such propositional logic is a combination of features in either non-negated or negated form. Since the propositional logic is human interpretable, it is easy to extract rule-based reasoning from TM. Our methods demonstrate that such a rule can be controlled or fine-tuned by modifying the parameter specificity s. We show that by keeping the value of s small, we can filter the clause from non-monotone to monotone where a majority of features are in the negated form thereby removing spurious correlations and forcing the model to rely on genuine correlations. Experiments results have shown that the proposed s-controlled TM outperforms various existing models on counterfactual test data. In addition, unlike DNNs, the human-level interpretation obtained from the rule-based reasoning of TM gives a complete understanding of how the model achieves its robustness.

References

[Abeyrathna *et al.*, 2021] Kuruge Darshana Abeyrathna, Bimal Bhattarai, Morten Goodwin, Saeed Rahimi Gorji, Ole-Christoffer Granmo, Lei Jiao, Rupsa Saha, and Rohan K. Yadav. Massively parallel and asynchronous tsetlin ma-

chine architecture supporting almost constant-time scaling. In *ICML*, pages 10–20. PMLR, 2021.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, California, USA, 2015.

[Bhattarai *et al.*, 2020] Bimal Bhattarai, Ole-Christoffer Granmo, and Lei Jiao. Measuring the novelty of natural language text using the conjunctive clauses of a tsetlin machine text classifier, 2020.

[Bhattarai *et al.*, 2022] Bimal Bhattarai, Ole-Christoffer Granmo, and Lei Jiao. Word-level human interpretable scoring mechanism for novel text detection using tsetlin machines. *Applied Intelligence*, 2022.

[Brunner *et al.*, 2020] Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. On identifiability in transformers. In *ICLR*, Addis Ababa, Ethiopia, 2020.

[Getoor and Taskar, 2007] L. Getoor and B. Taskar. Introduction to statistical relational learning. In *MITPress*, 2007.

[Goguen, 1973] J. Goguen. Zadeh l. a.. fuzzy sets. information and control , vol. 8 (1965), pp. 338–353. zadeh l. a.. similarity relations and fuzzy orderings. information sciences , vol. 3 (1971), pp. 177–200. *Journal of Symbolic Logic*, 38:656–657, 1973.

[Granmo *et al.*, 2019] Ole-Christoffer Granmo, Sondre Glimsdal, Lei Jiao, Morten Goodwin, Christian W. Omlin, and Geir Thore Berge. The Convolutional Tsetlin Machine, 2019.

[Granmo, 2018] Ole-Christoffer Granmo. The tsetlin machine - a game theoretic bandit driven approach to optimal pattern recognition with propositional logic, 2018.

[Jia and Liang, 2017] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *EMNLP*, pages 2021–2031, Copenhagen, Denmark, 2017. ACL.

[Kaufman *et al.*, 2012] Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Trans. Knowl. Discov. Data*, 6, 2012.

[Kaushik *et al.*, 2020] Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *ICLR*, Online, 2020.

[Kim *et al.*, 2016] Byeongchang Kim, Seonghan Ryu, and G. Lee. Two-stage multi-intent detection for spoken language understanding. *Multimedia Tools and Applications*, 76:11377–11390, 2016.

[Krizhevsky *et al.*, 2012] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012.

[Lei *et al.*, 2021] Jie Lei, Tousif Rahman, Rishad Shafik, Adrian Wheeldon, Alex Yakovlev, Ole-Christoffer Granmo, Fahim Kawsar, and Akhil Mathur. Low-Power Audio Keyword Spotting using Tsetlin Machines, 2021.

[Lu *et al.*, 2020] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. *Gender Bias in Neural Natural Language Processing*, pages 189–202. Springer International Publishing, 2020.

[Ni *et al.*, 2019] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*, pages 188–197, Hong Kong, China, 2019. ACL.

[Peters *et al.*, 2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL: Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana, 2018. ACL.

[Raedt and Kimmig, 2015] L. D. Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100:5–47, 2015.

[Rosenthal *et al.*, 2017] Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th SemEval-2017*, pages 502–518, Vancouver, Canada, 2017. ACL.

[Rudin, 2018] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2018.

[Sauer and Geiger, 2021] Axel Sauer and Andreas Geiger. Counterfactual generative networks. In *ICLR*, Online, 2021.

[Serrano and Smith, 2019] Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of ACL*, pages 2931–2951, Florence, Italy, 2019. ACL.

[Wang and Culotta, 2020] Zhao Wang and Aron Culotta. Identifying spurious correlations for robust text classification. In *Findings of the EMNLP 2020*, pages 3431–3440, Online, 2020. ACL.

[Wulczyn *et al.*, 2017] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *International Conference on World Wide Web*, page 1391–1399, Perth, Australia, 2017. WWW.

[Yadav *et al.*, 2021a] Rohan Kumar Yadav, Lei Jiao, Ole-Christoffer Granmo, and Morten Goodwin. Enhancing interpretable clauses semantically using pretrained word representation. In *BlackboxNLP*, pages 265–274, Punta Cana, Dominican Republic, 2021. ACL.

[Yadav *et al.*, 2021b] Rohan Kumar Yadav, Lei Jiao, Ole-Christoffer Granmo, and Morten Goodwin. Human-Level Interpretable Learning for Aspect-Based Sentiment Analysis. In *Proceedings of AAAI, Vancouver, Canada*. AAAI, 2021.