


Article

Enhancing Attention's Explanation Using Interpretable Tsetlin Machine

Rohan Kumar Yadav ^{1,*} and Dragoş Constantin Nicolae ²

¹ Centre for Artificial Intelligence Research, Department of Information and Communication, University of Agder, 4879 Grimstad, Norway

² Research Institute for Artificial Intelligence "Mihai Drăgănescu", 050711 Bucharest, Romania; dragosnicolae555@gmail.com

* Correspondence: rohan.k.yadav@uia.no

Abstract: Explainability is one of the key factors in Natural Language Processing (NLP) specially for legal documents, medical diagnosis, and clinical text. Attention mechanism has been a popular choice for such explainability recently by estimating the relative importance of input units. Recent research has revealed, however, that such processes tend to misidentify irrelevant input units when explaining them. This is due to the fact that language representation layers are initialized by pre-trained word embedding that is not context-dependent. Such a lack of context-dependent knowledge in the initial layer makes it difficult for the model to concentrate on the important aspects of input. Usually, this does not impact the performance of the model, but the explainability differs from human understanding. Hence, in this paper, we propose an ensemble method to use logic-based information from the Tsetlin Machine to embed it into the initial representation layer in the neural network to enhance the model in terms of explainability. We obtain the global clause score for each word in the vocabulary and feed it into the neural network layer as context-dependent information. Our experiments show that the ensemble method enhances the explainability of the attention layer without sacrificing any performance of the model and even outperforming in some datasets.



Citation: Yadav, R.K.; Nicolae, D.C. Enhancing Attention's Explanation Using Interpretable Tsetlin Machine. *Algorithms* **2022**, *15*, 143. <https://doi.org/10.3390/a15050143>

Academic Editors: Fabio Massimo Zanzotto and Frank Werner

Received: 24 March 2022

Accepted: 20 April 2022

Published: 22 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: NLP; interpretability; explainability; Tsetlin Machine; Bi-GRUs; attention

1. Introduction

In natural language processing, text categorization is a crucial task (NLP) [1,2] and neural network models are the ones to dominate state-of-the-art approaches. However, these models are often assumed to be blackbox in nature. The models' opacity has become a serious impediment to their creation, implementation, and improvement, especially in crucial tasks like medical diagnosis [3] and legal document inspection [4]. As a result, explainable text classification has become a major topic, with the objective of providing end-users with human-readable descriptions of the classification logic [1,5–7].

The attention mechanism is a prominent technique among current explainability approaches that identify essential sections of the input for the prediction job by offering a distribution across attended-to-input units [8]. Many NLP tasks, such as text categorization, question answering, and entity identification, have shown outstanding results using attention-based models [2,8,9]. In particular, in many NLP systems, the self-attention mechanism that underpins the Transformer design has played a key role [10,11]. Despite this, recent research has revealed that learned attention weights are frequently unrelated to the relevance of input components as judged by various explainability approaches [12], and that alternative attention distributions can provide identical predictions [13,14].

Various alternative approaches could replace attention-based neural networks for explainable NLP such as decision tree and logistic regression. However, they suffer from low performance compared to neural networks. In addition to this logistic regression does not provide a logical explanation but provides mathematical weights for selected

inputs [15]. On the other hand, decision trees are only suited for a limited dataset size. It becomes extremely difficult to get the explainability once the trees get more complex. Due to these limitations, there has been a limited study in obtaining a logical explanation for NLP classification. A recent study has found that Tsetlin Machine (TM) has been a promising tool for rule-based explanation in image, text, and numerical data [16–19]. TM is an interpretable rule-based model that uses conjunctive clauses to learn basic and complicated correlations. Unlike Deep Neural Networks (DNNs) and basic rule-based systems, TM learns rules in the same way that humans do, using logical reasoning, and it does so in a visible and interpretable manner [16,20]. TM has shown that it obtains a good trade-off between accuracy and interpretability on many NLP tasks [21,22]. However, there lie some limitations such as boolean bag-of-words input and incapable of using pre-trained information.

One efficient way to deal with the above-mentioned problem is to use prerequisite knowledge to enhance the input layer for better interpretation. Integrating human rationales as supplementary supervision information for attention learning is a promising way to enhance the explainability of attention-based models. Human rationales have previously been found to be useful input for increasing model performance and discovering explainable input in model prediction [1,23]. However, obtaining such human rationales is an expensive and time-consuming process. Hence, to make it more easy and efficient, we use a logic-based model TM that mimics human-level understanding to generate prerequisite information to initialize the input layer of the neural network. Since TM can be explained by logic and rules, the information it provides can be easily explained to make the attention layer focus on important input tokens.

In this paper, we train TM on two movie review datasets and leverage the clause score of TM for each word in the vocabulary. We then use this prerequisite information of each word as initial information for the input layer in the neural network. We use Bidirectional Gated Recurrent Unit (Bi-GRU) [24] for language representation for neural networks and GloVe [25] to initialize the word embedding. In addition to this, we multiply the input embedding layer with prerequisite information of each word from TM. This makes the attention layer on top of Bi-GRU focus on important words.

2. Related Works

Machine learning explainability has lately received a lot of attention, owing to the necessity for transparency [5,26]. Existing explainability approaches may be divided into two types: post-hoc and intrinsic explainability. The goal of post-hoc explainability is to provide explanations for a model that already exists. In the feature space, a representative technique approximates decisions of the model with an explainable technique (e.g., a linear model) [5]. Generative Explanation Framework (GEF) [27] is a recent development in this field that aims to explain a generic encoder-predictor architecture by concurrently generating explanations and classification results. The goal of intrinsic explainability is to create self-explanatory models. This can be accomplished by enforcing feature sparsity [28], representation disentanglement [29], or sensitivity to input characteristics through explainability requirements in model learning. Attention mechanisms, that identify sections of the input that are considered by the model for specific output predictions, are a more prevalent technique to explain individual predictions [8,30]. These attention processes have long been essential in NLP, not just because of their explainability, but also because of the improvements they provide to model performance [10,11]. An empirical study recently questioned their effectiveness in explaining model performance, pointing out that attention distributions are contrary with the importance of input features measured by gradient-based methods, and those adversarial distributions can be found yielding similar model performance [13]. These discoveries have sparked heated debates, such as how attention mechanisms provide larger weights to key input features for a specific task even when the model for prediction changes [14].

The concept of adding human rationales for improvement of the model may be traced back to a situation in which a human teacher highlights sections of text in a document as a justification for label annotation [23]. By restricting the prediction labels, the logic is integrated into the loss function of SVM classifier. Similar concepts have been investigated for neural network models [1] and various methods of human reason integration, such as learning a mapping between human rationales and machine attention [31] or assuring variety among hidden representations learned at different time steps [32]. Even though recent studies in human computation [33] have shown that asking workers to provide human annotation rationales—by headlining the supporting text excerpts from the given context—requires no additional annotation effort, reassigning human rationales in the previous datasets requires additional time and cost. Hence, there is the need for a human explainable model that can substitute the human-in-loop system as prerequisite knowledge for the neural network model.

In this paper, we propose an alternative for human rationales by using Tsetlin Machine and its explainability. TM consists of several clauses in the form of propositional logic. Each feature in TM represents the collection of the clauses for a particular classification model. Such clause score also represents the weightage of each feature in the model. Since TM is easily explainable, it makes sense to use this explanation as ensemble information for the neural network.

3. Proposed Architecture: TM Initialized Attention Model

Here, we discuss the architecture of the model that ensemble the information from TM into neural network. First we explain the architecture of TM and then process to obtain the clause score.

3.1. Clause Score from Tsetlin Machine Architecture

A revolutionary game-theoretic strategy that organizes a collection of decentralized team of Tsetlin Automata is at the heart of the TM (TAs). Based on disjunctive normal form, the strategy directs the TAs to learn arbitrarily complicated propositional formula in the form of conjunctive form [34,35]. A TM is interpretable in a way that it decomposes issues into self-contained sub-patterns that may be interpreted separately, notwithstanding its ability to learn complicated nonlinear patterns. Each sub-pattern is represented by a conjunctive sentence, which is a series of literals, each of which represents an input bit or its negation. As a result, sub-pattern representation and evaluation are both Boolean. In comparison to other approaches, this makes the TM computationally efficient and hardware friendly [36,37].

TM is a new classification approach based on a team of Tsetlin Automata that manipulates phrases in propositional logic. TA is a deterministic automaton with a fixed structure that learns the best action from a collection of actions provided by the environment. A two-action TA with $2N$ states is shown in Figure 1. The states from 1 to N are referred to as Action 1, whereas the states from $(N + 1)$ to $2N$ are referred to as Action 2. TA conducts the action depending on the current state and interacts with the environment during each iteration. This, in turn, causes the environment to issue a random reward or penalty based on an unknown probability distribution. If TA is rewarded, it advances deeper into the state; if it is penalized, it moves closer to the center of the state, weakening the preformed action, and finally jumping to the side of the other action. Each input bit in TM is represented by two TAs, TA and TA' . The original bit of the input sample is controlled by TA , while the negation is controlled by TA' . As a result, the TM, which is made up of clauses, will eventually converge to the desired pattern. There are two sorts of feedback (reward or penalty) supplied to the TM: Type I and Type II feedback. The TA for the training samples is given rewards or penalties based on these feedback types. The both feedbacks are shown in Tables 1 and 2 respectively.

Table 1. The Type I Feedback [16].

Input	Clause Literal	1		0	
		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 2. The Type II Feedback [16].

Input	Clause Literal	1		0	
		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

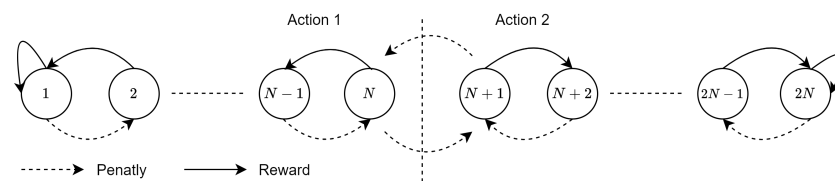


Figure 1. The two-action TA and its transition in TM.

In regards to NLP, TM heavily relies on the Boolean Bag-of-words (BOW) given by $X = [x_1, x_2, x_3, \dots, x_n]$. Let l be the number of clauses that represent each class of the TM, covering q classes altogether. Then, the overall learning problem is solved using $l \times q$ clauses. Each clause $C_i^j, 1 \leq j \leq q, 1 \leq i \leq l$ of the TM is given by :

$$C_i^j = \left(\bigwedge_{k \in I_i^j} x_k \right) \wedge \left(\bigwedge_{k \in \bar{I}_i^j} \neg x_k \right), \tag{1}$$

where I_i^j and \bar{I}_i^j are non-overlapping subgroup of the input variable indices, $I_i^j, \bar{I}_i^j \subseteq \{1, \dots, m\}, I_i^j \cap \bar{I}_i^j = \emptyset$. The subgroup decide that which of the input variables to participate in the clause, and whether they are in the original form or the negated. The indices of input features in I_i^j represent the literals that are included as original form of the literals, while the indices of input features in \bar{I}_i^j correspond to the negated ones. Among the q clauses of each class, clauses that are indexed with odd number are assigned positive polarity (+) whereas those with even indexed are assigned negative polarity (-). The clauses with positive polarity vote for the true target class and those with negative polarity vote against it. A summation operator aggregates the votes by subtracting the total number of negative votes from positive votes, as shown in Equation (2).

$$f^j(X) = \sum_{i=1,3,\dots}^{l-1} C_i^j(X) - \sum_{i=2,4,\dots}^l C_i^j(X). \tag{2}$$

For q number of classes, the predicted output y is given by the argmax operator which classifies the input features based on the highest sum of votes obtained, as shown in Equation (3).

$$\hat{y} = \operatorname{argmax}_j (f^j(X)). \quad (3)$$

Once the model is trained with a particular dataset, we can explore the clauses that holds information of combination of literals in propositional form. Such information is humanly interpretable and can be used for downstream applications of NLP. Here, we explore the weightage of each word in the model. We pass each word in the vocabulary into the TM and obtain the clause score. The clause score is calculated by:

$$SC_{x_k} = |f^{k=tp}(X_{x_k=1}) - \Sigma f^{k=fp}(X_{x_k=1})| \quad (4)$$

Here tp refers to true prediction, fp refers to false prediction, $|\cdot|$ refers to the absolute value, and $k = 1, 2, \dots, n$ where n is the number of vocabulary. We then create the input map for each input sentence with the score obtained for each word which will then fed to neural network initial embedding layer.

3.2. Attention Based Neural Network

Here we explain the attention-based neural network for text classification where we use conventional Bi-GRU as the language representation layer and attention on top of it.

Because of its linked hidden layers, where the internal states are used to process data in a sequential fashion, recurrent neural networks (RNNs) [38] have lately become the standard for NLP. RNNs, on the other hand, have several drawbacks that have led to the creation of versions like LSTM and GRU. The GRU, like the LSTM unit, regulates the flow of information without using a memory unit, making it more efficient with near-lossless performance [39]. GRU also overcomes the issue of vanishing gradients and gradient explosions in vanilla RNN. Our selected model consists of a Bi-GRU layer on top of embedding layer initialized with Glove embedding. This layer consists of a attention layer on top of Bi-GRU. The overall architecture of proposed model is shown in Figure 2.

Consider a sentence "This is wonderful movie." which is fed to the embedding layer initialized by Glove embedding. On the other hand, we obtain the clause score for each word in the sentence and feed to the embedding layer to match the dimension of input sentence embedding. Then both the embedding layer is passed to multiplication layer, where both are multiplied element wise. The output of the multiplication layer is then fed to the Bi-GRU having multiple hidden layers. Let us assume that the input to Bi-GRU is given by $X = [x_1, x_2, x_3, \dots, x_k]$ where k is the padded length of the input sentence. This information is passed to Bi-GRU layer. In GRU, there are two types of gates: update gates and reset gates. The update gate determines how much previous data must be brought into the current state and how much new data must be introduced.

On the other hand, reset gate decides how much information from the previous steps is passed into the current state h_t . Here, h_t is the output from the GRU at time step t and z_t means the update gate. At a specific time step t , the new state h_t is given by:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h_t, \quad (5)$$

where \odot represents the element-wise multiplication. To update z_t , we have

$$z_t = \sigma(W_{z_t}x_t + U_{z_t}h_{t-1} + b_{z_t}). \quad (6)$$

Here, x_t is each word of the sentence at time step t that is passed into the network unit which is then multiplied with its own weight W_{z_t} . Similarly, h_{t-1} represents the information of previous unit and is multiplied with its own weight U_{z_t} and b_{z_t} is the bias associated with update state. The current state h_t is updated using reset gate r_t by

$$h_t = \tanh(W_{h_t}x_t + r_t \odot (U_{h_t}) + b_{h_t}). \tag{7}$$

At r_t , the candidate state of step t can get the information of input x_t and the status of h_{t-1} of step $t - 1$. The update function of r_t is given by

$$r_t = \sigma(W_{r_t}x_t + U_{r_t}h_{t-1} + b_{r_t}), \tag{8}$$

where W_{r_t} and U_{r_t} are the weights associated with the reset state and b_{r_t} is the bias.

The Bi-GRU consists the forward GRU layer (\vec{h}_t) that models the input sentence from step 0 to t and the backward GRU (\overleftarrow{h}_t) from t to 0.

$$\vec{h}_t = \vec{GRU}(x_t), t \in [1, T], \tag{9}$$

$$\overleftarrow{h}_t = \overleftarrow{GRU}(x_t), t \in [T, 1], \tag{10}$$

$$h_t = \begin{bmatrix} \vec{h}_t \\ \overleftarrow{h}_t \end{bmatrix}. \tag{11}$$

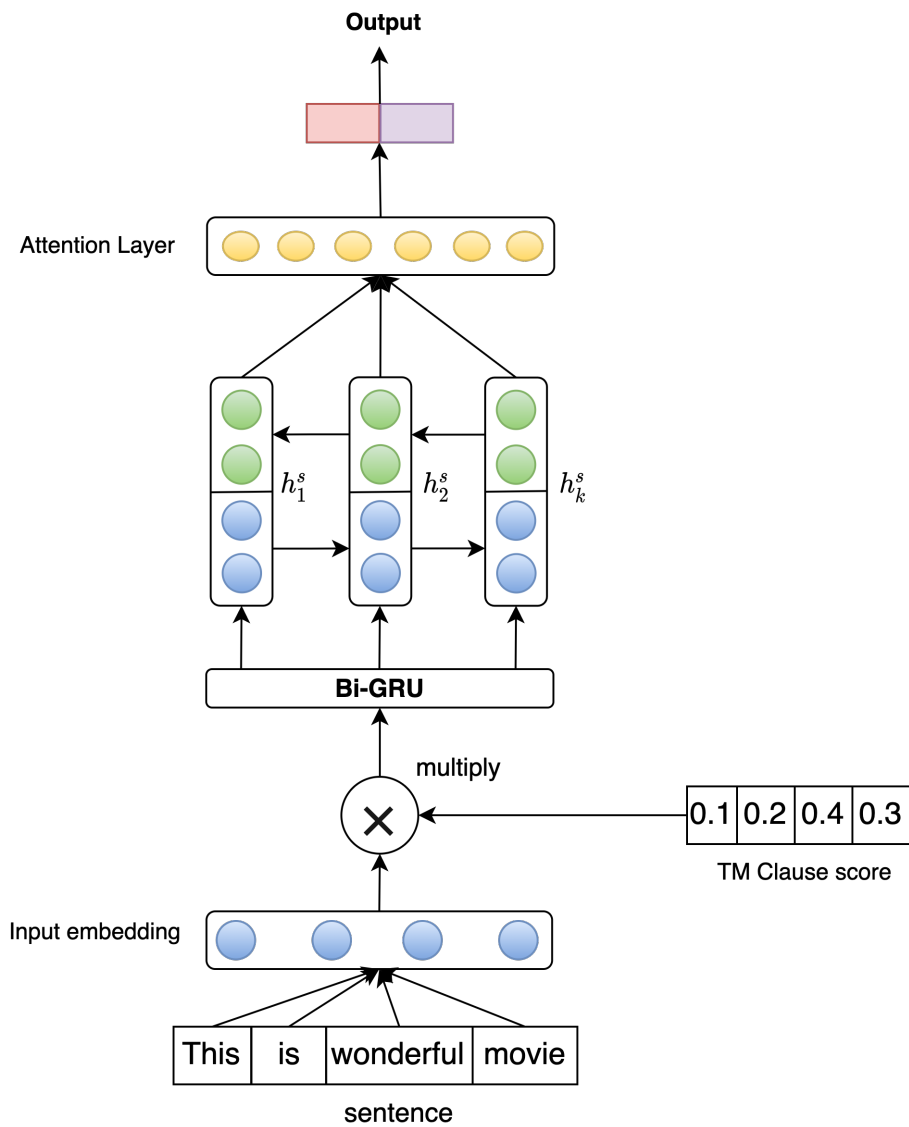


Figure 2. The two-action TA and its transition in TM.

As we all know, not all of the words in the context contribute equally to text categorization. As a result, an attention layer is allocated to the context to prioritize significant words. Attention layer is fed on top of *Bi-GRU* to learn the weight α_t for each hidden state h_t obtained at time step t . Since there are k inputs in the padded sequences, time step t will be from 1 to k . The weighting vector $\alpha = (\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k)$ is calculated based on the output sequence $H = (h_1, h_2, h_3, \dots, h_k)$. The attention vector s_1 for AL_1 is calculated based on the weighted sum of these hidden states, as:

$$s_1 = \sum_{t=1}^k (\alpha_t h_t), \quad (12)$$

where the weighted parameter α_t^1 is calculated by:

$$\alpha_t = \frac{\exp(u_t^T u_w)}{\sum_t \exp(u_t^T u_w)}, \quad (13)$$

where $u_t = \tanh(W_w h_t + b_w)$. Here W_w and h_t are the weight matrices and b_w represents the bias. The parameter u_w demonstrates context vector that is different at each time step, which is randomly initialized and learned jointly during the training process.

4. Experiments and Results

Here, we demonstrate the experiments and the result on the proposed model for enhancing the explanation of attention layer in text classification. We use two sentiment classification datasets for evaluation. They are:

- **MR** is a movie review dataset for binary sentiment classification with just one sentence per review [40]. There are 5331 positive reviews and 5331 critical reviews in the corpus. In this study, we used a training/test split from [41] (<https://github.com/mnqu/PTE/tree/master/data/mr> (accessed on 24 February 2022)).
- **Reuters** The Reuters 21,578 dataset has two subsets: R52 and R83 (all-terms version). R8 is divided into eight categories, including 5485 training and 2189 exam papers. R52 is divided into 52 categories and 6532 training and 2568 test papers.

We employ Keras [42] to implement our model. Adam [43] is used as the models' optimization method with the learning rate of $1 \times e^{-3}$. Additionally, we adopted Dropout [44] as the regularization strategy and the probability of Dropout was kept to be 0.25. Words are initialized with Glove [25] of 300-dimension word embedding. The batch size was 128 and was run for 100 epochs in the test datasets for obtaining the best results.

Since, the main purpose of this paper is to enhance the explanation of the attention layer, we demonstrate the performance of the proposed model with the relatable models to show the impact of each model. The comparable state-of-the-arts are explained below:

- **TF-IDF+LR**: Bag-of-words model with inverse document frequency weighting for term frequency. The classifier is based on logistic regression.
- **CNN**: CNN-rand uses arbitrarily initialized word embeddings [45].
- **LSTM**: The LSTM model that we employ here is from [46], representing the entire text using the last hidden layer. We used both the model that is using pretrained embeddings and without using.
- **Bi-LSTM**: Bi-directional LSTMs are widely used for text classification that models both forward and backward information.
- **PV-DBOW**: PV-DBOW is a paragraph vector model where the word order is not considered and is trained with Logistic Regression used as a softmax classifier [47].
- **PV-DM**: PV-DM is a paragraph vector model, with word ordering taken into consideration [47].
- **fastText**: This baseline uses the average of the word embeddings provided by fastText as document embedding. The embedding is then fed to a linear classifier [48].

- **SWEM:** SWEM applies simple pooling techniques over the word embeddings to obtain a document embedding [49].
- **Graph-CNN-C:** A graph CNN model uses convolutions over a word embedding similarity graph [50], employing a Chebyshev filter.
- **Tsetlin Machine:** Simple BOW model for Tsetlin Machine without feature enhancement.
- **Bi-GRU+Attn:** Bi-directional GRUs are widely used for text classification. We compare our model with Bi-GRU fed with pre-trained word embeddings along with attention layer on top of it.
- **TM+Bi-GRU+Attn:** Proposed model with Bi-GRU model with pretrained word embedding initialized with pretrained TM score in its input layer.

4.1. Performance Comparison with State-of-the-Arts

Table 3 shows the comparison of performance for selected datasets. As we can see that traditional method such as TF-IDF with Logistic Regression (TF-IDF+LR) performs decently in MR with 74.59, R8 with 93.74, and R52 with 86.95. Some sophisticated language model such as CNN, and LSTM performs quite similarly. The only improvement seen among them is Bi-LSTM which incorporates both past and future information for better input representation thereby reaching 77.06% in MR, 96.68% in R8, and 90.54% in R52. Slightly different than language models PV-DBOW and PV-DM performs poorly in all three datasets. Similarly, Graph-based CNN and SWEM perform on par with the state-of-the-arts baselines. On the other hand, the rule-based method TM performs quite comparable to baseline by reaching 75.14% in MR, 96.16% in R8, and 84.62% in R52. The performance is slightly below Bi-LSTM/GRU-based model because of its restriction to use pre-trained word embedding. However, Yadav et al. [22] show that embedding similar words using a pre-trained word embedding significantly enhances the performance and outperforms the baselines. However, our proposed model only uses TM explainability to generate prerequisite word weightage to replace human attention input into neural network language models. Hence, this is demonstrated in the table as well. Even though the motive of this task does not necessarily impact the accuracy but there is a slight increase in performance anyway. This is due to the fact that the TM score gives additional weightage to the model's input thereby reaching 77.95% in MR, 97.53% in R8, and 95.71% in R52 for TM+Bi-GRU+Attn. This shows an increment of about 1% in average throughout the selected datasets.

Table 3. Performance of the proposed model (TM+Bi-GRU+Attn) with selected baselines.

Models	MR	R8	R52
TF-IDF+LR	74.59	93.74	86.95
CNN	74.98	94.02	85.37
LSTM	75.06	93.68	85.54
Bi-LSTM	77.68	96.31	90.54
PV-DBOW	61.09	85.87	78.29
PV-DM	59.47	52.07	44.92
SWEM	76.65	95.32	92.94
Graph-CNN-C	77.22	96.99	92.75
Tsetlin Machine	75.14	96.16	84.62
Bi-GRU+Attn	77.15	96.20	94.85
TM+Bi-GRU+Attn	77.95	97.53	95.71

In addition to this, we also evaluate some more metrics that supports the performance of the proposed model. Since MR is only binary classification dataset, R8 and R52 is multiclass dataset. Hence, there is need of the evaluation of the performance of each class. Usually unbalanced or multiclass datasets sometimes suffers with low F-scores because the model greedily learns the majority classes. Hence to have a clear picture of our proposed model, we evaluate precision, recall, and f-scores of main baseline TM, Bi-GRU+Attn with our proposed model TM+Bi-GRU+Attn as shown in Tables 4–6 respectively. The results clearly indicate the our proposed model also performance superior on all three selected

metrics for macro, micro, and weighted form of measurement compared to baselines TM and Bi-GRU+Attn. The performance of our proposed model is significantly higher in case of R8 and MR across all metrics. However the difference in performance for R52 is very marginal. In case of comparison with TM, our proposed outperforms all the measures for all three datasets.

Table 4. Performance of TM for various evaluation metrics.

Models	MR	R8	R52
Precision (macro)	73.22	86.12	79.18
Recall (macro)	70.42	87.44	75.44
F-Score (macro)	69.32	88.32	76.66
Precision (micro)	70.42	94.82	85.28
Recall (micro)	70.42	94.82	85.28
F-Score (micro)	70.42	94.82	85.28
Precision (weighted)	73.22	95.02	85.51
Recall (weighted)	70.42	95.12	85.12
F-Score (weighted)	69.32	95.02	85.28

Table 5. Performance of Bi-GRU+Attn for various evaluation metrics.

Models	MR	R8	R52
Precision (macro)	75.21	88.69	82.32
Recall (macro)	72.20	90.66	79.26
F-Score (macro)	71.34	89.26	79.87
Precision (micro)	72.20	95.52	95.63
Recall (micro)	72.20	95.52	95.63
F-Score (micro)	72.20	95.52	95.63
Precision (weighted)	75.21	95.60	95.33
Recall (weighted)	72.20	95.23	95.63
F-Score (weighted)	71.34	95.49	95.34

Table 6. Performance of TM+Bi-GRU+Attn for various evaluation metrics.

Models	MR	R8	R52
Precision (macro)	75.63	94.70	83.81
Recall (macro)	74.62	93.32	80.23
F-Score (macro)	74.61	93.39	80.67
Precision (micro)	74.62	96.52	96.82
Recall (micro)	74.62	96.52	96.82
F-Score (micro)	74.62	96.52	96.85
Precision (weighted)	75.63	96.58	96.51
Recall (weighted)	74.62	96.52	96.52
F-Score (weighted)	74.61	96.51	96.49

4.2. Explainability

Here, we explore the proposed model's explainability by visualizing the respective attention weight. The attention weight usually gives the impact of each individual feature for a particular prediction. However, such weight usually gives the relationship between input and the output, such method of interpreting model can be beneficial for system to understand the impact of each features. Since neural network are already an established blackbox models, one can use this interpretation to generate explainability for the understanding the context of prediction. Hence we define interpretation of the model as the weights obtained from attention layer and explainability as use-case of interpretation to design the reasoning for a particular prediction that is easily understandable to humans.

For ease of illustration, we visualize the attention weight of the Bi-GRU model and the attention weight of the Bi-GRU model initialized with TM’s word score. We use the red color gradient to demonstrate the weightage of each input word in the context. Dark color represents the higher weightage with light color representing lower weightage. As we can see from Figure 3, only using Bi-GRU, the model recognizes mostly important words for predicting correct sentiment class. However, it is not perfect as the human level. However, Figure 4 shows the visualization of attention weight using Bi-GRU and TM’s score.

Here we can see that the model focus on more significant words than the previous model. For instance, in the first example, the later model captures “look”, “away” with higher weightage which is an important context for negative sentiment than “directing” and “attempt”. This is more clearly seen in the third sample as the first model focus on “easily”, “best”, and “film” however our proposed model shifts the higher weightage to “best”, “Korean”, “film” for predicting the positive sentiment. One of the most peculiar cases where there are ambiguities in the context consisting of both positive and negative sentiment words as in the last example. Here using only Bi-GRU, the model captures “forgettable”, “rip”, and “work” as thigh-impact words. However, it does not give high weightage to the word “cheerful” which is also sentiment carrying word. However, using our proposed model, the weightage changes drastically and the model assigns higher weightage to “forgettable”, “cheerful”, “but”, and “earlier”. This makes more sense to human understanding because the context the has word “cheerful” and it is contradicted with the word “but” which eventually leads to a negative sentiment the carrying word “forgettable” thereby making the whole context negative sentiment.

is an arthritic attempt at directing by callie khouri. i had to look away - this was god awful	Negative
a visually seductive , unrepentantly trashy take on rices second installment of her vampire chronicles	Positive
could easily be called the best korean film of 2002	Positive
the best disney movie since the lion king	Positive
a cheerful enough but imminently forgettable rip-off of [bessons] earlier work	Negative

Figure 3. Visualization of attention weights with Bi-GRU only. Dark red to light red color represents the color gradients based on the attention weights in descending order.

is an arthritic attempt at directing by callie khouri. i had to look away - this was god awful	Negative
a visually seductive , unrepentantly trashy take on rices second installment of her vampire chronicles	Positive
could easily be called the best korean film of 2002	Positive
the best disney movie since the lion king	Positive
a cheerful enough but imminently forgettable rip-off of [bessons] earlier work	Negative

Figure 4. Visualization of attention weights with Bi-GRU and TM Score. Dark red to light red color represents the color gradients based on the attention weights in descending order.

5. Conclusions

Recently, attention weights have been a great tool for visualization of the weightage of input rationales in the model. However, their weightage sometimes gives higher weightage

to unwanted tokens that did not make sense to humans. This led to the requirement of human-annotated rationales that are embedded into the models. Even, such human annotators are not a very extensive task to obtain while annotating new datasets, the problems come annotating human rationales to existing datasets. It takes high time and cost to re-annotate human rationales for explainability. Hence, in this paper, we propose an alternative approach to get human explainable rationales using interpretable Tsetlin Machine (TM). Since TM can be explained using logical rules, it provides human-level interpretation and is used as a prerequisite annotation of input rationales. The proposed model shows that embedding such information in attention-based models not only increases the accuracy but also enhances the weightage of attention layer for each input rationales thereby making the explanation more sensible to humans. The visualization also shows that the proposed model is capable of capturing the ambiguity of the context much better than traditional models.

However, the concern with current study of explainability in AI is the subjectivity of explainability. Even though the mode of interpreting a model has been very sophisticated with proof of concept. It still fails to align with human understanding because of the subjectivity of opinion. Hence, as a future work, one can collect the human rationales annotation while manually labelling the particular datasets. This can be used as an evaluation criteria on how explainability of ML models align with various human understanding.

Author Contributions: Conceptualization, R.K.Y. and D.C.N.; methodology, R.K.Y.; software, R.K.Y.; validation, R.K.Y. and D.C.N.; formal analysis, R.K.Y. and D.C.N.; investigation, R.K.Y.; resources, R.K.Y.; writing—original draft preparation, R.K.Y.; writing—review and editing, R.K.Y. and D.C.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: <https://github.com/mnqu/PTE/tree/master/data/mr> accessed on 24 March 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, Y.; Marshall, I.J.; Wallace, B.C. Rationale-Augmented Convolutional Neural Networks for Text Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Volume 2016, pp. 795–804.
2. Wang, W.; Yang, N.; Wei, F.; Chang, B.; Zhou, M. Gated Self-Matching Networks for Reading Comprehension and Question Answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; Volume 1, pp. 189–198.
3. Lakkaraju, H.; Bach, S.H.; Leskovec, J. Interpretable Decision Sets: A Joint Framework for Description and Prediction. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1675–1684.
4. Mahoney, C.J.; Zhang, J.; Huber-Fliflet, N.; Gronvall, P.; Zhao, H. A Framework for Explainable Text Classification in Legal Document Review. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 1858–1867.
5. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2016.
6. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning JMLR (ICML'17), Sydney, Australia, 6–11 August 2019; Volume 70, pp. 3319–3328.
7. Camburu, O.M.; Rocktäschel, T.; Lukasiewicz, T.; Blunsom, P. e-SNLI: Natural Language Inference with Natural Language Explanations. In Proceedings of the NeurIPS, Montréal, QC, Canada, 3–8 December 2018.
8. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv* **2015**, arXiv:1409.0473.
9. Parikh, A.; Täckström, O.; Das, D.; Uszkoreit, J. A Decomposable Attention Model for Natural Language Inference. In Proceedings of the Conference on Empirical Methods in Natural Language Processing; Austin, TX, USA, 1–5 November 2016; pp. 2249–2255.

10. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 4171–4186.
12. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv* **2014**, arXiv:1312.6034.
13. Jain, S.; Wallace, B.C. Attention is not Explanation. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1, pp. 3543–3556.
14. Wiegrefe, S.; Pinter, Y. Attention is not not Explanation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 11–20.
15. Lipton, Z.C. The Mythos of Model Interpretability. *Queue* **2018**, *16*, 31–57. [[CrossRef](#)]
16. Granmo, O.C. The Tsetlin Machine—A Game Theoretic Bandit Driven Approach to Optimal Pattern Recognition with Propositional Logic. *arXiv* **2018**, arXiv:1804.01508.
17. Granmo, O.C.; Glimsdal, S.; Jiao, L.; Goodwin, M.; Omlin, C.W.; Berge, G.T. The Convolutional Tsetlin Machine. *arXiv* **2019**, arXiv:1905.09688.
18. Yadav, R.K.; Jiao, L.; Granmo, O.C.; Goodwin, M. Human-Level Interpretable Learning for Aspect-Based Sentiment Analysis. In Proceedings of the AAI, Vancouver, BC, Canada, 2–9 February 2021.
19. Bhattarai, B.; Granmo, O.C.; Jiao, L. Explainable Tsetlin Machine framework for fake news detection with credibility score assessment. *arXiv* **2021**, arXiv:2105.09114.
20. Abeyrathna, K.D.; Bhattarai, B.; Goodwin, M.; Gorji, S.R.; Granmo, O.C.; Jiao, L.; Saha, R.; Yadav, R.K. Massively Parallel and Asynchronous Tsetlin Machine Architecture Supporting Almost Constant-Time Scaling. In Proceedings of the ICML, PMLR, Online, 2021; pp. 10–20.
21. Yadav, R.K.; Jiao, L.; Granmo, O.C.; Goodwin, M. Interpretability in Word Sense Disambiguation using Tsetlin Machine. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART), Vienna, Austria, 4–6 February 2021.
22. Yadav, R.K.; Jiao, L.; Granmo, O.C.; Goodwin, M. Enhancing Interpretable Clauses Semantically using Pretrained Word Representation. In Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, Punta Cana, Dominican Republic, 11 November 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 265–274.
23. Zaidan, O.; Eisner, J.; Piatko, C. Using Annotator Rationales to Improve Machine Learning for Text Categorization. In Proceedings of the Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, USA, 22–27 April 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 260–267.
24. Cho, K.; van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014.
25. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the EMNLP, Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
26. Doshi-Velez, F.; Kim, B. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv* **2017**, arXiv:1702.08608.
27. Liu, H.; Yin, Q.; Wang, W.Y. Towards Explainable NLP: A Generative Explanation Framework for Text Classification. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5570–5581.
28. Freitas, A.A. Comprehensible classification models: A position paper. *SIGKDD Explor.* **2014**, *15*, 1–10. [[CrossRef](#)]
29. Zhang, Q.; Wu, Y.N.; Zhu, S.C. Interpretable Convolutional Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018; pp. 8827–8836.
30. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.C.; Salakhutdinov, R.; Zemel, R.S.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the ICML, Lille, France, 6–1 July 2015.
31. Bao, Y.; Chang, S.; Yu, M.; Barzilay, R. Deriving Machine Attention from Human Rationales. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 1903–1913.
32. Mohankumar, A.K.; Nema, P.; Narasimhan, S.; Khapra, M.M.; Srinivasan, B.V.; Ravindran, B. Towards Transparent and Explainable Attention Models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, WA, USA, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 4206–4216.
33. McDonnell, T.; Lease, M.; Kutlu, M.; Elsayed, T. Why Is That Relevant? Collecting Annotator Rationales for Relevance Judgments. In Proceedings of the HCOMP, Austin, TX, USA, 30 October–3 November 2016.
34. Zhang, X.; Jiao, L.; Granmo, O.C.; Goodwin, M. On the Convergence of Tsetlin Machines for the IDENTITY- and NOT Operators. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**. [[CrossRef](#)] [[PubMed](#)]

35. Sharma, J.; Yadav, R.; Granmo, O.C.; Jiao, L. Human Interpretable AI: Enhancing Tsetlin Machine Stochasticity with Drop Clause. *arXiv* **2021**, arXiv:2105.14506.
36. Lei, J.; Wheeldon, A.; Shafik, R.; Yakovlev, A.; Granmo, O.C. From Arithmetic to Logic Based AI: A Comparative Analysis of Neural Networks and Tsetlin Machine. In Proceedings of the 27th IEEE International Conference on Electronics Circuits and Systems (ICECS2020), Online, 2020.
37. Lei, J.; Rahman, T.; Shafik, R.; Wheeldon, A.; Yakovlev, A.; Granmo, O.C.; Kawsar, F.; Mathur, A. Low-Power Audio Keyword Spotting Using Tsetlin Machines. *J. Low Power Electron. Appl.* **2021**, *11*, 18. [[CrossRef](#)]
38. Mikolov, T.; Karafi, M.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Interspeech, Makuhari, Japan, 6–30 September 2010.
39. Chung, J.; Gülçehre, Ç.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
40. Pang, B.; Lee, L. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Proceedings of the Association for Computational Linguistics, Ann Arbor, MI, USA, 26–31 July 2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 115–124.
41. Tang, J.; Qu, M.; Mei, Q. PTE: Predictive Text Embedding through Large-Scale Heterogeneous Text Networks. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; Association for Computing Machinery: New York, NY, USA, 2015; pp. 1165–1174.
42. Chollet, François and Others: Keras. 2015. Available online: <https://keras.io> (accessed on 1 March 2022).
43. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
44. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
45. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1746–1751.
46. Liu, P.; Qiu, X.; Huang, X. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proceedings of the IJCAI, Manhattan, CA, USA, 9–16 July 2016; pp. 2873–2879.
47. Le, Q.; Mikolov, T. Distributed Representations of Sentences and Documents. In Proceedings of the 31st International Conference on Machine Learning, PMLR, Beijing, China, 21–26 June 2014; Volume 32, pp. 1188–1196.
48. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. In Proceedings of the EACL, Valencia, Spain, 3–7 April 2017; ACL: Stroudsburg, PA, USA, 2017; Volume 2, pp. 427–431.
49. Shen, D.; Wang, G.; Wang, W.; Min, M.R.; Su, Q.; Zhang, Y.; Li, C.; Henao, R.; Carin, L. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms. In Proceedings of the ACL, Melbourne, Australia, 26–28 March 2018; ACL: Stroudsburg, PA, USA, 2018; Volume 1, pp. 440–450.
50. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016, Volume 29.