Received: 8 April 2021 | Revised: 24 July 2021 | Accepted: 13 October 2021

DOI: 10.1111/exsy.12873

ORIGINAL ARTICLE

Expert Systems WILEY

Using Tsetlin Machine to discover interpretable rules in natural language processing applications

Rupsa Saha | Ole-Christoffer Granmo | Morten Goodwin

Centre for AI Research, University of Agder, Grimstad, Norway

Correspondence
Rupsa Saha, Centre for AI Research, University of Agder, Grimstad, Norway.
Email: rupsa.saha@uia.no

Abstract

Tsetlin Machines (TM) use finite state machines for learning and propositional logic to represent patterns. The resulting pattern recognition approach captures information in the form of conjunctive clauses, thus facilitating human interpretation. In this work, we propose a TM-based approach to three common natural language processing (NLP) tasks, namely, sentiment analysis, semantic relation categorization and identifying entities in multi-turn dialogues. By performing frequent itemset mining on the TM-produced patterns, we show that we can obtain a global and a local interpretation of the learning, one that mimics existing rule-sets or lexicons. Further, we also establish that our TM based approach does not compromise on accuracy in the quest for interpretability, via comparison with some widely used machine learning techniques. Finally, we introduce the idea of a relational TM, which uses a logic-based framework to further extend the interpretability.

KEYWORDS
artificial intelligence, interpretable AI, multi-turn dialogue analysis, natural language processing, rule mining, semantic analysis, sentiment analysis

1 | INTRODUCTION

The burgeoning of machine learning (ML) and artificial intelligence (AI) research and techniques have resulted in their widespread use in everyday technology. Currently, many AI-based systems lack interpretability, that is, the ability to explain their actions or decisions in a form that is understandable to humans. Without proper explanations, identifying and preventing erroneous behaviour becomes impossible, or at the very least, complicated. Sometimes the consequences of this lack of interpretability are extremely harsh, from a human-centric point of view, such as prisoners being incorrectly denied bail or questionable financial decisions being taken without human intervention. As more and more AI-based systems become part of the social fabric of our lives, building trust in such systems becomes paramount (Rudin, 2019).

A major driving force behind the push to make AI interpretable is to enable researchers and engineers to improve their models by leveraging the model itself. By accessing and analysing information on where and why a model fails, debugging (or bettering) it becomes easier. Perhaps one of the most compelling reasons for using AI as compared to human reasoning has been, historically, that machines have the potential of being unbiased. Unfortunately, that belief has been held for a very long time and only recently have studies been conducted into the veracity of the claim. It has come to light that bias is more widespread in AI systems than previously thought, and it creeps undetected into algorithms either via inherent biases that researchers hold themselves or even from the data used for developing models. Limited interpretability hampers the detection of such bias once it is encoded into a model and their subsequent rectification, thus leading to biased systems falsely being advertised otherwise (Lipton, 2018).

Despite the increasing research into the necessity and techniques of interpretable AI, still, there are not many readily available interpretable ML techniques (compared to standard non-interpretable ones), especially those that work well for the type of complex problems that AI generally deals with. Current research usually follows either of two different approaches: (a) Model-based—intrinsic interpretability, which is achieved during the running of a model and the model is simple and describable; and (b) Post-hoc—extrinsic interpretability, where explanations are extracted from a trained model after the model has been run. There are concerns that model-based approaches may not reach sufficiently high predictive accuracy. Contrarily, post-hoc processing also suffers from the handicap that the explanations acquired are, at best, an approximation of what the model has learned. It is not enough for a method to be highly accurate—the extracted explanations must also be relevant (Murdoch et al., 2019).

By dint of relying on standard ML techniques, natural language processing (NLP) is also afflicted by the conflict between interpretability and accuracy. There are several approaches to pattern recognition in NLP, some of which we list here in order of an increasing loss of interpretability: rule-based pattern-recognition system, statistical (TF-IDF) (Jones, 2004) methods, linear classifiers (Aggarwal & Zhai, 2012), and neural network models employing vector space representations of words (Bengio et al., 2003). Basic neural network architectures are currently further enhanced using multiple techniques, such as grammatical information, pooling, convolution, and so forth (Collobert et al., 2011; Socher et al., 2013), leading to gains in terms of accuracy. This work proposes the use of a recently introduced paradigm, Tsetlin Machine (TM), for NLP, and show that it can successfully address concerns related to both interpretability and accuracy.

1.1 | Tsetlin Machines

TM are a pattern recognition approach, which provides an interpretable approach to ML (Granmo, 2018). A TM stands apart from other traditional techniques in its ability to construct patterns in the form of highly discriminative conjunctive clauses in propositional logic (hence making them understandable to humans). More recently, Phoulady et al. (2019) proposed a modified learning mechanism that combats false positives and encourage true positives, resulting in a weighted scheme that further increases the discrimination power of the clauses by assigning them weights. Recent works show Tsetlin machines to have successfully addressed several machine learning tasks, including natural language understanding (Berge et al., 2019; Bhattarai et al., 2021; Saha et al., 2020; Yadav et al., 2021a, 2021b), speech understanding (Lei et al., 2021), image analysis (Granmo et al., 2019), classification (Abeyrathna et al., 2021), and regression (Darshana Abeyrathna et al., 2020). While the performance showed by Tsetlin machines in such areas has been comparable to state-of-the-art machine learning techniques, the method has also been shown to have a smaller memory footprint and faster inference in reported cases than more traditional neural network-based models (Granmo et al., 2019; Wheeldon et al., 2020). Furthermore, Shafik et al. (2020) elaborate on Tsetlin machines being able to completely mask stuck-at faults, thus having fault-tolerant properties as well. Inherent interpretability makes Tsetlin machines a promising candidate for the cause of interpretable AI. Indeed, conjunctive clauses have turned out to be well suited for human interpretation, while still allowing complex nonlinear patterns to be formed (Wang et al., 2017).

1.2 | Paper contributions

In this work, we aim to showcase the ability as well as the appropriateness of a TM-based classification technique to produce human-intelligible clauses when applied to NLP via experiments and analyses on sentiment analysis, semantic relation analysis and dialogue-based entity identification. We focus on the linguistic patterns obtained by means of the clauses, to identify their informativeness and interpretability. We compare these clauses with expert-crafted rules wherever available to better judge them against an established standard. We also show that we can maintain interpretability while achieving competitive accuracy, and suggest areas for further research-based upon our findings. An important aspect of the continued direction of our work is the relational TM, where we propose to enhance the framework with a logic-based structure that focuses on learning interactions between actions and entities, rather than on literals (words), from natural language text.

Paper Organization. The paper is organized as follows. In Section 2, we present a short background on interpretability, related work on interpretable NLP, and interpretable NLP with Tsetlin machines. Section 3 focuses on the working of the TM and further details on how it is suited for NLP tasks. In Sections 4 and 5, we describe how we employ TMs in three different NLP experiments, and analyse the results, as well as the by-products, in light of interpretability. Future direction of work, including a relational version of the model proposed in this work, are concisely put forward in Section 6, before finally concluding in Section 7.

2 | BACKGROUND AND RELATED WORK

As mentioned earlier in Section 1, one way of bringing interpretation to ML is via a post-hoc model that explains an initial black-box model. Unfortunately, the unknown amount of approximation involved in such a process leads to these explanations often being not reliable and can thus be

misleading (Rudin, 2019). For an explanation to be completely faithful, it needs to be an explicit description of the model itself, which is very often impossible to attain. Even so, an explanation should, at the least, correspond to how the model behaves specifically when an instance is being predicted. This brings up the two related, but separate ideas of global and local fidelity. Importantly, local fidelity does not imply global fidelity: globally important features may not be important in the local context. While global fidelity would (ideally) imply local fidelity, identifying globally faithful explanations that are interpretable remains a challenge for complex models (Ribeiro et al., 2016). Ultimately, the inclination is towards innately interpretable models, since they have a distinct correspondence to the actual computation carried out by the model (Rudin, 2019).

Existing work on interpretability for NLP has primarily focused on text classification problems, though none involves models that can organically explain the decisions. Samek et al. (2017) proposed the use of heat maps to obtain information about how much each hidden element contributes to the prediction, and use that information to build connections between the input and the output. Another approach, by Ribeiro et al. (2016) involves a model-agnostic approach: an interpretable model of the predictions of Black-box models is used, where the interpretable model describes the actions of the classifier locally (i.e., for the instance being classified). Hancock et al. (2018) developed a framework via which human annotators provided a natural language explanation for each label, and these annotations were subsequently used to create a weakly supervised larger training set. This was ultimately used to train a classifier that is capable of classifying text together with an explanation for the same. Finally, Liu et al. (2018) presented a hybrid generative-discriminative method for text classification, which creates a novel generative explanation framework that can generate reasonable explanations using information inferred from raw texts.

Since research into the application of Tsetlin machines has been relatively recent, there has been limited investigation done in terms of exploring its capabilities for NLP. Berge et al. (2019) use a TM-based approach to analyse medical data and learn human-interpretable rules from that. The approach is able to successfully categorize text, based on the presence or absence of unique medical terminology. The potential of detecting novel content in documents by using linguistic structures that are characteristic of a particular text type was shown by Bhattarai et al. (2021). Another interesting application was demonstrated by Yadav et al. (2021b) in their work dealing with word sense disambiguation depending on the surrounding context.

3 | TSETLIN MACHINE FOR NLP TASKS

3.1 | General classification and learning using Tsetlin machines

Introduced in the 1960s, a Tsetlin automaton (TA) is a deterministic automaton that learns the optimal action to be performed, among the set of actions available in an environment. It performs the action associated with its current state, prompting a reward or penalty to be activated, based on the ground truth. The state is updated correspondingly so that the TA gradually shifts closer towards the optimal action (Tsetlin, 1961).

A TM is composed of a collection of such TAs, which work together to express complex propositional formulae using conjunctive clauses. It takes a vector $X = (x_1,...,x_f)$ of Boolean features as input, to be classified into one of two classes, $y = 0$ or $y = 1$. These features, along with their negated counterparts, $\bar{x}_k = \neg x_k = 1 - x_k$, make up a set of literals $L = \{x_1,...,x_f,\bar{x}_1,...,\bar{x}_f\}$.

A TM pattern is formulated as a conjunctive clause C_j, formed by ANDing a subset $L_j \subseteq L$ of the literal set L:

$$C_j(X) = \wedge_{l_k \in L_j} l_k = \prod_{l_k \in L_j} l_k. \tag{1}$$

For example, the clause $C_j(X) = x_1 \wedge x_2 = x_1 x_2$ consists of the literals $L_j = \{x_1, x_2\}$ and outputs 1 if $x_1 = x_2 = 1$.

The number of clauses is defined by a parameter n. $n/2$ number of clauses are assigned positive polarity (henceforth termed as positive clauses), while the other half is assigned negative polarity (negative clauses). A classification decision is obtained by consolidating the clause outputs via summation and thresholding using the unit step function $u(v) = 1$ **if** $v \geq 0$ **else** 0:

$$\hat{y} = u\left(\sum_{j=1}^{n/2} C_j^+(X) - \sum_{j=1}^{n/2} C_j^-(X)\right). \tag{2}$$

That is to say, majority voting decides the classification, with positive clauses voting in support of $y = 1$ and negative clauses voting for the opposite, that is, $y = 0$. For example, the classifier $\hat{y} = u(x_1\bar{x}_2 + \bar{x}_1 x_2 - x_1 x_2 - \bar{x}_1\bar{x}_2)$, captures the XOR-relation (as shown in Figure 1) (Granmo, 2018).

Algorithm 1 encompasses the entire learning procedure. We observe that learning is performed by a team of 2f TAs per clause, one TA per literal l_k (Algorithm 1, Step 2). Each TA has two actions—Include or Exclude—and decides whether to include its designated literal l_k in its clause. Learning which literals to include is based on reinforcement: Type I feedback encourages frequent patterns, while Type II feedback increases the discriminative power of the patterns.

Tsetlin machines learn online, processing one training example (X, y) at a time (Step 7). The TAs first produce a new configuration of clauses (Step 8), $C_1^+,...,C_{n/2}^-$, followed by calculating a voting sum v (Step 9).

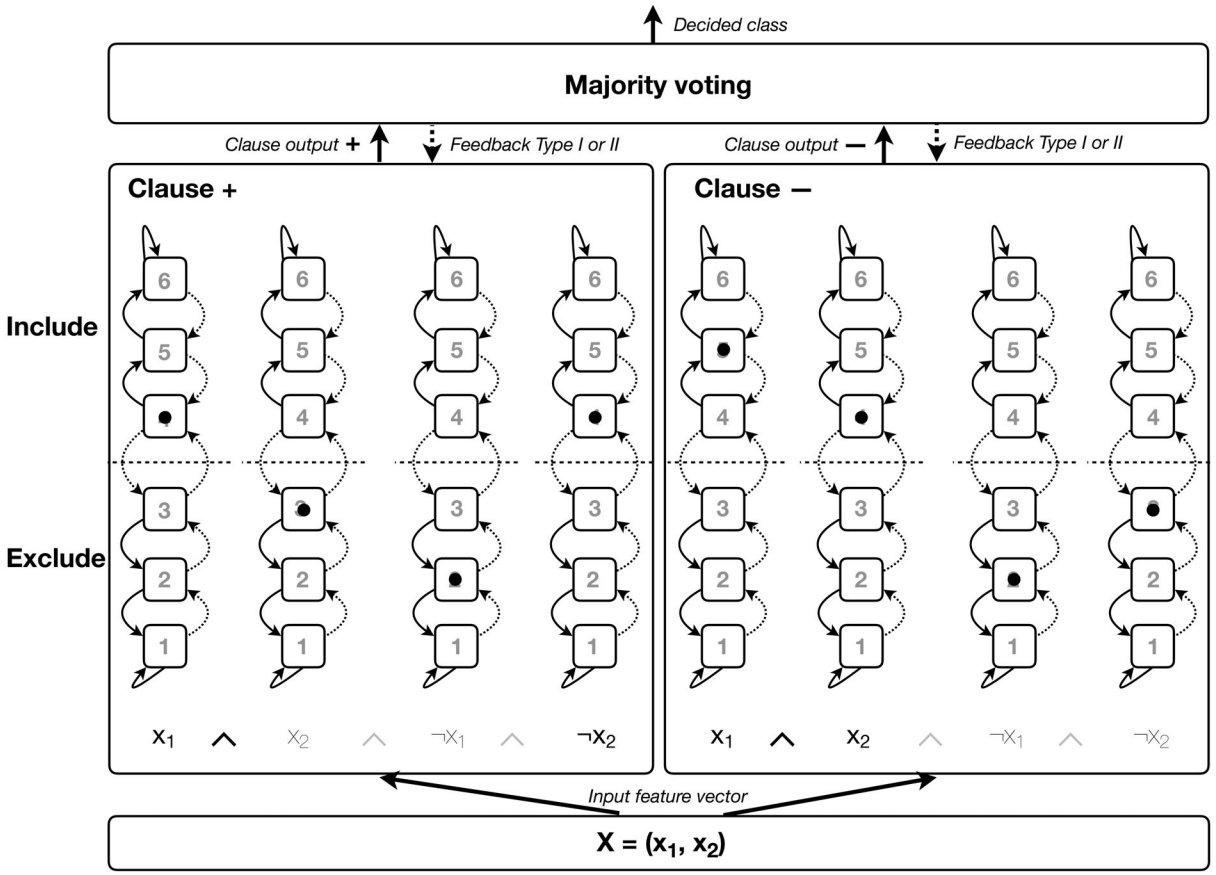


FIGURE 1 The Tsetlin Machine architecture

Feedback is then distributed stochastically to each TA team. The difference ϵ between the clipped voting sum v^c and a user-set voting target T determines the probability with which each TA team will receive feedback (Steps 12–20).

Type I feedback is given probabilistically to positive clauses when $y = 1$ and to negative clauses when $y = 0$. Each clause, in turn, reinforces its TAs based on the following three things: (1) its output $C_j(X)$; (2) the action of the TA – Include or Exclude; and (3) the value of the literal l_k assigned to the TA. Two rules are used to control Type I feedback:

- *Include* is rewarded and *Exclude* is penalized with probability $\frac{s-1}{s}$ whenever $C_j(X) = 1$ and $l_k = 1$. This reinforcement is strong (i.e., activated with high probability) and ensures the clause remembers and refines the pattern it recognizes in X.[1]
- *Include* is penalized and *Exclude* is rewarded with probability $\frac{1}{s}$ whenever $C_j(X) = 0$ or $l_k = 0$. This reinforcement is triggered with low probability and transforms infrequent patterns into more frequent ones by coarsening the infrequent ones.

Above, the user-configurable parameter s controls pattern frequency, i.e., a higher s produces less frequent patterns.

Type II feedback is similarly probabilistic but is given to positive clauses when $y = 0$ and to negative clauses when $y = 1$. It penalizes *Exclude* whenever $C_j(X) = 1$ and $l_k = 0$. Thus, this feedback produces literals for discriminating between $y = 0$ and $y = 1$, by making the clause evaluate to 0 when facing its competing class. Further details can be found in Granmo (2018).

To note here, the voting sum is clipped to normalize the feedback probability. The voting target for $y = 1$ is T and for $y = 0$ it is $-T$. For any input X, the probability of reinforcing a clause gradually drops to zero as the voting sum approaches the user-set target. This is to ensure that clauses distribute themselves across the frequent patterns, rather than ignoring some and over-concentrating on others.

3.2 | Tsetlin Machines in NLP

A snapshot of the training and testing mechanism in a TM setup for NLP is shown in Figure 2a. It also depicts how a global and local interpretation of the task is acquired from the clauses. In the figure, the TM takes a Boolean feature vector as input $D_i = [f_1, f_2, ..., f_o] \in \{0,1\}^o$ (Figure 2a: *Training*

Algorithm 1

TM algorithm

input Tsetlin Machine (TM), example pool S, training rounds e, clauses n, features f, voting target T, specificity s

procedure train (TM, S, e, n, f, T, s)
  for $j \leftarrow 1,\ldots,n/2$ **do**
    $TA_j^+ \leftarrow$ Randomly Initialize Clause TA Team(2f)
    $TA_j^- \leftarrow$ Randomly Initialize Clause TA Team(2f)
  end for
  for $i \leftarrow 1,\ldots,e$ **do**
    $(X_i, y_i) \leftarrow$ Obtain Training Example(S)
    $C_1^+,\ldots,C_{n/2}^- \leftarrow$ Compose Clauses$\left(TA_1^+,\ldots,TA_{n/2}^-\right)$
    $v_i \leftarrow \sum_{j=1}^{n/2} C_j^+(X_i) - \sum_{j=1}^{n/2} C_j^-(X_i)$ ▷ Vote sum
    $v_i^c \leftarrow$ **clip**$(v_i, -T, T)$ ▷ Clipped vote sum
    for $j \leftarrow 1,\ldots,n/2$ **do** ▷ Update TA teams
      if $y_i = 1$ **then**
        $\epsilon \leftarrow T - v_i^c$ ▷ Voting error
        Type_I_Feedback (X_i, TA_j^+, s) **if** rand() $\leq \frac{\epsilon}{2T}$
        Type_II_Feedback (X_i, TA_j^-) **if** rand() $\leq \frac{\epsilon}{2T}$
      else
        $\int \leftarrow T + v_i^c$ ▷ Voting error
        Type_II_Feedback (X_i, TA_j^+) **if** rand() $\leq \frac{\epsilon}{2T}$
        Type_I_Feedback (X_i, TA_j^-, s) **if** rand() $\leq \frac{\epsilon}{2T}$
      end if
    end for
  end for
end procedure

Feature Vectors). In this scenario, D_i refers to a singular input text, while each feature f_k represents the presence/absence of a specific unigram or bigram in D_i, as the case may be for a particular experiment.

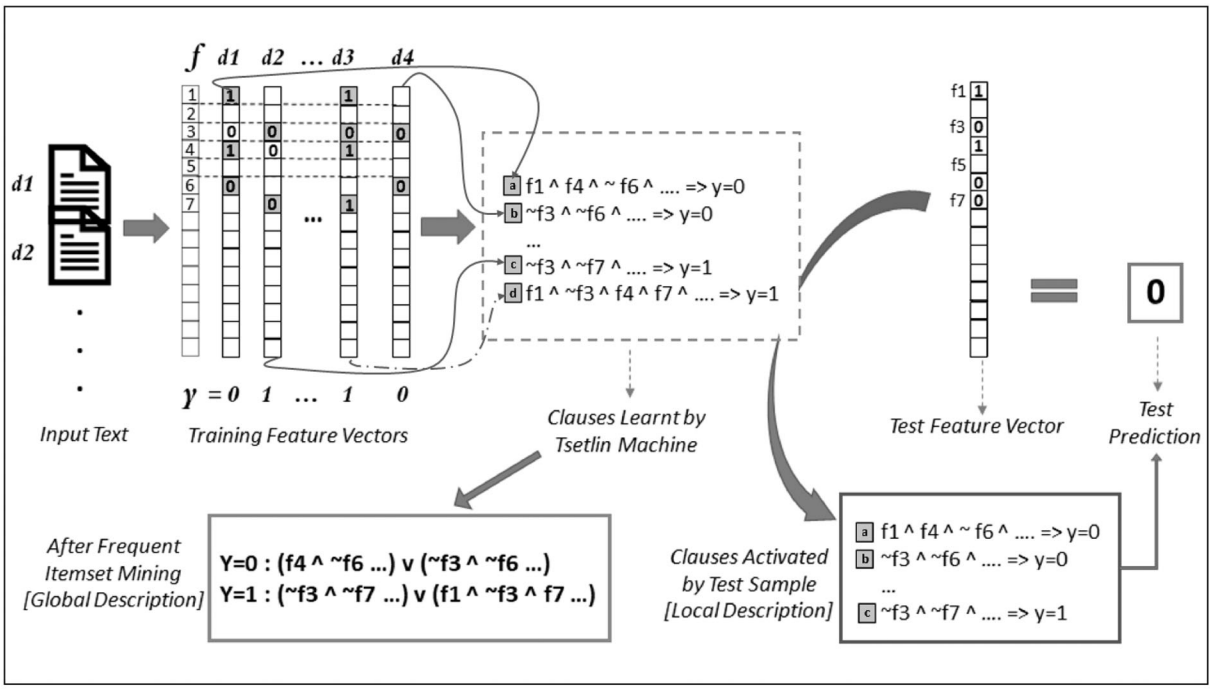
The feature vector is further processed by the clauses $C_1^+,\ldots,C_{\frac{n}{2}}^+$ and $C_1^-\ldots,C_{\frac{n}{2}}^-$. Each clause captures a particular linguistic sub-pattern as a conjunction of literals: $C_a = f_1 \wedge \ldots \wedge f_4 \wedge \neg f_6 \wedge \ldots$ (Figure 2a: *Clauses Learnt by Tsetlin Machine*).

During learning, a larger T together with an increase in the number of clauses leads to the creation of more specific clauses. That is, each clause encapsulates very particular sub-patterns seen in the text without much overlap between clauses. The evidence for a sample to belong to a class is aggregated and thresholded for the output, as discussed previously. As expected, not all the clauses learnt by the TM actively participates in the decision making process for a particular sample, only a small subset of the clauses are required. This subset leads to a local description of the said sample (In Figure 2a: *Local Description*, the prediction is made by $C_a, C_b, C_c\ldots$).

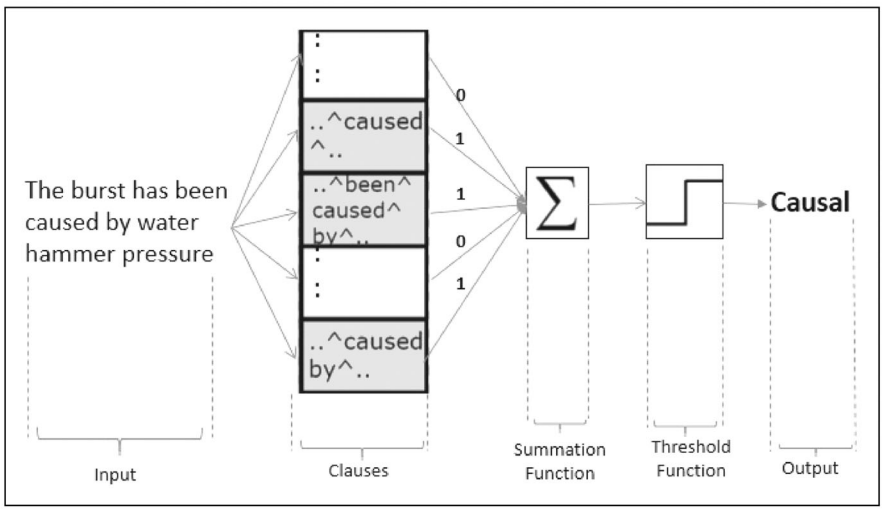
In the process of forming clauses that best describe the training data, the TM chooses features that are judged most important for decision-making. In our experiments, features are unigrams and bigrams for the first two datasets, and only unigrams for the third one. Since the clauses are initialized with a random set of literals at the beginning, in a single epoch of training, there are some inconsequential literals selected as part of one or more clauses, along with the important ones. To preserve our original objective, i.e. interpretability of rules for text classification, we take advantage of the Tsetlin machine's proclivity towards frequently occurring sub-patterns. We perform a frequent itemset mining (Agrawal & Srikant, 1994) of clauses over multiple epochs, and arrive at a set of representative sub-clauses (Figure 2a: Global Description). As discussed subsequently in Section 5, these representatives hold up well to linguistic scrutiny. The same approach can also be used for analysing local clauses, i.e. clauses activated for making a classification in a single sample only.

4 | EXPERIMENTAL SETUP

In this work, we have used three datasets: (a) SemEval 2010 Semantic Relations dataset (Hendrickx et al., 2009), (b) Sentiment140 Twitter dataset (Go et al., 2009), and (c) Multi-Turn, Multi-Domain, Task-Oriented Dialogue Dataset (Eric & Manning, 2017).



(a)



(b)

FIGURE 2 Tsetlin Machine (TM) for natural language processing tasks. (a) Mechanism of interpretable rule mining using TM. (b) Example of classification using TM

The SEMEVAL-2010 Task 8 focused on identifying semantic relations between pairs of nominals in text. The dataset consists of 10,717 annotated samples, each of which belongs to one of the following 10 relation classes: Cause-Effect (S:CE), Product-Producer (S:PP), Content-Container (S:CC), Member-Collection (S:MC), Entity-Origin (S:EO), Entity-Destination (S:ED), Instrument-Agency (S:IA), Component-Whole (S:CW), Message-Topic (S:MT) and Other.

The Sentiment140 dataset is primarily for sentiment classification on short texts. It contains 1.6 million English language tweets, classified as positive or negative. Preprocessing involves stripping the data of emoticons to force classifiers to learn from other features, i.e. the words only.

While the above two datasets have become fairly standard with respect to semantic relation analysis and sentiment analysis (in short texts) respectively, the third dataset we have used is targeted towards building task-oriented dialogue systems. It consists of 3030 multi-turn dialogues, each of which has an average of 5.25 utterances, belonging to one of three predefined tasks (Calendar Scheduling, Point of Interest Navigation or Weather Information Retrieval). Out of these, we use Weather Information Retrieval dialogues (which consist of 997 out of the total 3030

dialogues) for the purpose of this paper. Each such dialogue contains one or more of the following slots that form the basis of that dialogue: location, weekly time, and weather attribute. We task the classifier with determining whether or not each of these given slots is present at each turn of a dialogue.

To better, highlight the Tsetlin machine's capability to discern between the salient and non-important parts of the text, we purposefully do not perform stopword or punctuation removal in any of the datasets.

Since the SemEval dataset contains multiple possible classes, we first do a set of experiments in a binary classification setup (i.e., one-vs.-all), followed by a final experiment with all the classes in a multi-class setup. The Dialogue dataset was subjected only to binary one-vs.-all classifications. We use a standard TM model for all experiments,[2] as described by Berge (Berge et al., 2019). For experiments involving binary classification on the SemEval dataset, the TM was configured hyperparameters were 40 clauses, a threshold value (T) of 15 and s-value of 3.9. For the multi-class classification, we used 1500 clauses, since there was a lot more information to be learnt, with a T of 800. In case of the Sentiment140 dataset, we use 4000 clauses, with the same T and s as before. Classification on the Dialogue dataset required hyperparameters of 40 clauses, a T of 60 and s-value of 6.

5 | ANALYSIS OF INTERPRETABILITY PROVIDED BY TSETLIN MACHINES

The interpretability of the TM approach is expressed in the clauses produced in the learning process. Inspecting the clauses learnt (for training) and activated (for testing) establishes why each text was classified the way it was. In this section, we explore the clauses obtained in the experiments and showcase their relevancy with respect to the aim of providing a linguistic explanation. During training, the TM arrives at a set of clauses using the features provided, which together gives a description of the task in general. During testing, each sample can match only a subset of all clauses, and these clauses define the classification problem with respect to that particular sample only.

We begin with instances that contain a Cause-Effect relationship in the SemEval data. While it is possible to have multiple linguistic ways of expressing causation in text, the presence of one or more specific literals (termed as unambiguous causal connectives) is enough to classify a sentence as being a causal sentence (Xuelan & Kennedy, 1992). Hence, we can hypothesize that a major discriminator to classify sentences as causal or non-causal will be the presence or absence of such words in a sentence.

An overview of the task Cause-Effect versus all classification becomes clearer in the form of propositional logic, which we get from a frequent itemset from the clauses from over 300 runs of the proposed classifier:

$$\text{Sentence\_Contains}(\ (\textit{caused\_by}) \vee (\textit{causes}) \vee (\textit{triggered\_by}) \vee (\textit{resulted\_in}) \vee (\text{been\_caused\_by}) \vee (\text{radiating\_from}) \vee (\text{lead\_to}) \vee \ldots) \Rightarrow \text{Class}(\text{Cause-Effect}).$$

The literals present in the above description match those termed as 'causal connectives', by researchers in linguistic causality (Girju, 2003; Xuelan & Kennedy, 1992).

While the above summarizes the global description of the classification, it is not applicable in its entirety in the local context of a single sample. When classifying a single instance, like 'The burst has been caused by water hammer pressure' shown in Figure 2b, the clauses activated (greyed) provide a local explanation of the classification.

While there are no similar exhaustive studies available for the other relations as there is for causality, visual inspection of the descriptions obtained for the one-vs.-all classification for such relations indicate similar levels of success. For example, the description for the class Entity-Destination relationship is as follows:

$$\text{Sentence\_Contains}((\text{into\_the}) \vee (\textit{sent\_to}) \vee (\text{delivered\_to}) \vee (\textit{donated\_to}) \vee (\text{put\_inside}) \vee (\text{shipped\_to}) \vee (\text{added\_to}) \vee \ldots) \Rightarrow \text{Class}(\text{Entity-Destination}).$$

For the Sentiment140 Dataset, similarity with expert-verified lexicons is judged by comparing frequent subclauses with a range of existing standard Twitter Sentiment Analysis lexicons. The various manual and semi-automatically created sources used are: AFINN-111 (Nielsen, 2011), Affect Intensity Lexicon (Mohammad, 2018), EmoLex (Mohammad & Turney, 2013), NRC Hashtag Lexicon (Kiritchenko et al., 2014), LIWC (Tausczik & Pennebaker, 2010), Bing Liu's opinion lexicon (Hu & Liu, 2004), MPQA subjective lexicon (Wilson et al., 2005), Sentiment Composition Lexicon (Kiritchenko & Mohammad, 2017).

Table 1 shows the coverage obtained by frequent TM sub-patterns (created by training the TM) over the available lexicons. High overlap is obtained with almost all lexicons, further confirming that the TM-based approach can indeed automatically arrive at a linguistic description of the classification at hand.

These two experiments give a fair idea of how the TM-based classifier allows us to have a global interpretation of the classifier's work. The clauses obtained, taken together, piece together an understanding of the objective of the task at hand, giving us an approximate description of the individual classes. With the experiments on the Dialogue dataset, we focus on the local interpretability. In a multi-turn dialogue context, information that has been seen previously may not have an effect currently (at the current turn), but it still needs to be remembered for further

TABLE 1 Coverage of various lexicons by Tsetlin Machine clauses

Lexicon	Source: Twitter	Mode of creation	Size of L_D*	L_D coverage
AFINN-111	Yes	Manual	1363	87.97%
Affect Intensity	Yes	Manual	18,719	83.55%
EmoLex	No	Manual	6272	85.30%
NRC Hashtag	Yes	Automatic	29,249	85.68%
LIWC	No	Automatic	1771	92.20%
Bing Liu	No	Manual	1048	83.21%
MPQA	No	Manual	473	87.53%
SCL-NMA	No	Manual	1339	84.17%

Note: *L_D = Lexicon ∩ Dataset.

TABLE 2 Local snapshot of clauses for example dialogues

Dialogue	Turn	Speaker	Utterance	Clauses	Predicted class
1	1	Driver	In the next 48 h will it rain in Alhambra?	(S1_Loc and S1_in) or (S1_Loc and S1_Driver)	Present
	2	Assistant	Just a little: the weather in Alhambra will be drizzle today and overcast tomorrow.	(S1_Loc and S2_Loc) or NOT(#S1_LOC and S1_Driver)	Present
	3	Driver	Thank you.	(S1_Loc and S2_Loc and S3_Driver)	Absent
	4	Assistant	Sure!	(End_Dialogue and S4_Assistant and S1_Loc and #S3_Loc)	Absent
2	1	Driver	What is the weather forecast for the upcoming week?	(#S1_Loc and #S1_in and S1_Driver) or (#S1_Loc and #S1_at and S1_Driver)	Absent
	2	Assistant	What city would you like to know the weather about?	(#S1_Loc and #S1_in and S1_Driver and S2_what and S2_Assistant) or (S2_like and S2_what and S2_know)	Absent
	3	Driver	Will it be overcast in Durham this week?	(S3_Loc and S3_Driver and S3_in)	Present
	4	Assistant	It will not be overcast in Durham this week.	(S3_Loc and S3_Driver and #End_Dialogue and S4_Assistant)	Present
	5	Driver	Thanks.	(S3_Loc and S4_Loc and S4_thanks)	Absent
	6	Assistant	You are welcome.	(End_Dialogue and S6_Assistant and S3_Loc and S4_Loc and S4_thanks)	Absent

processing. Primarily, we wish to show that, due to the nature of clauses obtained by the TM, such behaviour is not only obtainable, but also track-able, on a per dialogue basis.

More importantly, this behaviour is available without compromising on the initial goal of classifying whether or not certain key information is present in the dialogue, as shown in Table 3 (rows Dialogue:Location, Dialogue:Weekly Time and Dialogue:Weather Attr.).

An interesting experimental observation was made at this stage with regards to the location information presence/absence classification. Taking inspiration from the original work, all the named location entities were changed to a generalized token 'LOC'. This one change led to the accuracy of classification going from 91.49% to 95.8%. Consequently, the learnt clauses are shown in Table 2 and the relevant discussions henceforth use the modified feature (i.e. using 'LOC' instead of location name).

In Table 2, we choose two typical examples from the dataset and show the local clauses that are triggered for each of the utterances in order to make the decision about the predicted class (location information present or location information absent). The feature-set consists of a bag of words for each utterance. Since an utterance is not a stand-alone entity, each feature is further enhanced by also including its turn number (designated as St_FeatureWord, where t is the turn number). Also, the feature vector for an utterance includes the feature vectors created for the previous utterances in the dialogue as well, if any, in order to ensure preceding context is not lost. Two special features are also included, one of which indicates the speaker of a given utterance (driver or assistant), and the other indicates whether or not an utterance is the end of a dialogue.

The first example is short, consisting of just four turns, all of which are classified correctly. The clauses for the first two utterances capture well the conditions in the text that lead to a 'present' classification. The clauses referred to for the last two utterances are more interesting. Not only do they capture the fact that a particular location was referred to earlier in the dialogue, they also recognize that that reference

TABLE 3 Comparing results of Tsetlin Machine-based classifier versus baseline approaches on SemEval-2010 task 8 data, Sentiment140 data and multi-turn dialogue data

Data	Tsetlin Machine	SVM	RF	NB	CNN-LSTM
SemEval:CE	**93.5**	92.3	86.3	86.9	87.11
SemEval:IA	**95.1**	94.0	91.9	89.9	93.2
SemEval:PP	**92.5**	90.7	89.8	86.3	91.13
SemEval:CC	91.6	**96.23**	90.4	90.7	92.89
SemEval:EO	**92.75**	91.6	87.4	86.5	90.45
SemEval:ED	92.5	**94.5**	89.3	88.8	88.4
SemEval:CW	89.75	87.8	86.3	85.9	**90.37**
SemEval:MC	**92.75**	91.1	88.4	86.8	91.02
SemEval:MT	**93.3**	92.2	89.1	86.4	92.02
SemEval:All	**50.20**	44.9	29.05	39.9	46.7
Sentiment140	69.4	**69.7**	62.52	60.5	49.9
Dialogue:Location	91.49	**94.9**	61.85	62.38	60.89
Dialogue:Weekly Time	**91.07**	90.2	86.40	69.21	82.70
Dialogue:Weather Attr.	**91.39**	90.48	64.43	64.50	66.82

Note: Best results for each dataset appear in bold.

does not affect the lack of location in the current utterance under scrutiny. In turn 4, the clause seems to encapsulate the ideas that (a) this is the end of the dialogue, (b) it is spoken by the assistant and (c) Driver provided location information in the first utterance but not in the subsequent ones.

The second example deals with a slightly more complex scenario than the first one. It consists of six turns (compared to just four in the previous one). The clauses for turn 1 indicate that (a) utterance is made by the driver, (b) but no mention of any location is made, leading to an 'absent' classification. Furthermore, clauses for turn 2 not only capture the previous information but also include the assistant asking for clarification. However, once the driver provides a location in turn 3, the previous information does not remain relevant. The last two turns are similar in terms of clauses to the previous example.

These examples reinforce the claim made previously, the flow of important information in a dialogue can be identified and tracked using a local sample-specific view of clauses in our proposed TM based approach.

In summary, the frequent itemset of clauses contribute to global interpretability, while sub-clauses that contribute to the final decision of a particular example relates to the local interpretability of the task. Moreover, the obtained clauses mimic expert-crafted rules, especially with less complexity than similar approaches (such as decision trees or random forest).

While we have discussed the interpretability of the clauses obtained by the proposed TM-based approach, we know discuss how the method does not sacrifice performance for the sake of interpretability. In order to compare with existing methods, we use SVM (RBF kernel), Random Forest (RF), Gaussian Naive Bayes (NB), and CNN-LSTM. While the first three are vanilla methods, we include CNN-LSTM to compare our work with an ensemble neural network model as well. Standard Python implementations with Scikit-Learn and Keras were employed. For the CNN-LSTM model, we used a one-dimensional Convolutional layer of filter size 64 with MaxPooling, and a unidirectional LSTM layer of output size 100, along with a dense layer with sigmoid activation. In the case of the random forest classifier, we used 100 trees with a maximum depth of two. Table 3 records the accuracy obtained by the proposed TM-based classifier on both datasets, in comparison to those by other algorithms as the baseline. The algorithms chosen for comparison are generally regarded as sufficient for the kinds of NLP tasks that we have shown in our experiments. Complex applications, for example, domain-specific tasks, may often need specialized approaches, including, but not limited to, pre-training, multiple models targeting specific aspects of the task, domain-dependent language models, and so forth. In this work, we wish to highlight the interpretability of the TA/TM paradigm, so we have chosen generalized methods rather than specialized ones. However, to ensure we do not end up with only a simplistic comparison in terms of accuracy, we also use the CNN-LSTM ensemble, to show that the TM performs at par with methods that are more complex as well.

Table 4 reports the average time to train taken by the TM versus CNN-LSTM and SVM on the three datasets, with the TM and CNN-LSTM each being trained over 100 epochs, in a multi-threaded DGX2 setup. The TM based architecture consistently trains faster compared to other methods.

In conclusion, our findings suggest that the TM-based approach produces results comparable to other standard text classification methodologies, and has comparable training times to neural networks.

TABLE 4 Comparison of average training times (in thousand seconds) for individual datasets using Tsetlin machines, CNN-LSTM and SVM

	Method		
Dataset	**Tsetlin Machine**	**CNN-LSTM**	**SVM**
SemEval	2.41	3.85	8.62
Sentiment140	4.06	8.27	19.30
Dialogue	0.007	0.024	0.016

6 | FUTURE WORK

We aim to continue this work, focusing on exploring further, how a feature set enriched by grammatical and semantic information (as in (Hendrickx et al., 2009)) can affect the performance of our currently proposed approach. We also plan to experimentally investigate the suitability of the convolutional TM architecture (Granmo et al., 2019) specifically for NLP applications, by making good on its promise of producing clauses that are even more informative. Further discussion and research are also required for the obtained clauses themselves. Making them more and more human-interpretable, while still being informative and concise, is a viable (and unexplored) avenue of research.

The propositional clauses constructed by a TM contribute both to a global and local understanding of the job and any propositional formula can be expressed by using the disjunctive normal form. Further work on Tsetlin machines aims to increase its computing power even further by employing first-order logic. The success achieved in correctly identifying various relations present in the text, as elaborated in this work, lays the ground work for the proposed expansion. Text in natural language is reduced to its constituent relations. This allows the TM to focus on understanding and learning in terms of consequences of interactions between the relations (rather than interactions between individual words), resulting in learning that closely mimic real-world occurrences. Furthermore, we also plan to take advantage of the observation noted in Section 5, with respect to the increase in accuracy when using a generalized token, rather than individual named entities. Initial experiments, which are beyond the scope of discussion of the current work, indicate that the usage of such generalized entities hand-in-hand with a relational approach lends itself extremely well to the TM paradigm, and increases its expressiveness. The resulting *Relational* TM (Saha et al., 2021) is expected to have a wider range of applications on NLP tasks, by taking advantage of logical structures that occur in natural language in order to encode rules representing actions and effects in the form of Horn clauses.

7 | CONCLUSION

In this work, we propose the usage of an interpretable TM-based approach to text categorization. Interpretability is provided by the clauses that the TM identifies during learning and uses for making the categorization decision. The linguistic structures evident in the clauses are sufficiently similar to those arrived at by human experts, thus ensuring that the success of the model is not merely based on statistical findings that are unrelated to the language itself. We conclude that we can get an all-round view of the learning and decision process due to the clauses: analysing all the clauses together as a whole gives us a global description of the task during training, while analysis of clauses that are activated for a single test sample gives a local description of that particular sample. Because of the multi-faceted view afforded by the TM-based approach, we believe that researchers may be able to use such methods to achieve an in-depth understanding of the data and the task, especially in situations where subject experts may be unavailable. Not only do Tsetlin machines allow for a degree of transparency in the decision making process, which is difficult to obtain in more conventional methods, they are also shown to perform at par with baseline approaches, both in terms of accuracy and training time. Moreover, the interpretability of the learning process enables effective triage for future betterment, even in a scenario where the current iteration of the model makes a mistake. Further enhancements to the approach, including provisions for a relational framework, show promising results in moving closer to real-world understanding from natural language texts.

In conclusion, our novel approach to various text categorization tasks using a TM based system shows promising interpretable results. We believe that further studies into more varied NLP tasks (sequence labelling, entity resolution, question answering) can highlight the power of the approach, and pave the way for more interpretable NLP by artificially intelligent systems.

ENDNOTES

[1] Note that the probability $\frac{s-1}{s}$ is replaced by 1 when boosting true positives.

[2] Python implementation of TM based classifier: code retrieved from https://github.com/cair/pyTsetlinMachine.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Rupsa Saha https://orcid.org/0000-0002-3006-5249

REFERENCES

Abeyrathna, K. D., Granmo, O.-C., & Goodwin, M. (2021). Extending the tsetlin machine with integer-weighted clauses for increased interpretability. *IEEE Access*, *9*, 8233–8248.

Aggarwal, C. C., & Zhai, C. (2012). *Mining text data*. Springer Science & Business Media.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. Paper presented at: Proc. 20th Int. Conf. Very large data bases, VLDB (Vol. 1215, pp. 487–499).

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Berge, G. T., Granmo, O.-C., Tveit, T. O., Goodwin, M., Jiao, L., & Matheussen, B. V. (2019). Using the tsetlin machine to learn human-interpretable rules for high-accuracy text categorization with medical applications. *IEEE Access*, *7*, 115134–115146.

Bhattarai, B., Jiao, L., & Granmo, O.-C. (2021). Measuring the novelty of natural language text using the conjunctive clauses of a tsetlin machine text classifier. Paper presented at: 13th International Conference on Agents and Artificial Intelligence (ICAART 2021).

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Darshana Abeyrathna, K., Granmo, O.-C., Zhang, X., Jiao, L., & Goodwin, M. (2020). The regression tsetlin machine: A novel approach to interpretable nonlinear regression. *Philosophical Transactions of the Royal Society A*, *378*(2164), 20190165.

Eric, M., & Manning, C. D. (2017). Key-value retrieval networks for task-oriented dialogue. arXiv preprint arXiv:1705.05414.

Girju, R. (2003). Automatic detection of causal relations for question answering. Paper presented at: Proceedings of the acl 2003 workshop on multilingual summarization and question answering-volume 12 (pp. 76–83).

Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12), 2009.

Granmo, O.-C. (2018). The Tsetlin machine–a game theoretic bandit driven approach to optimal pattern recognition with propositional logic. arXiv preprint arXiv:1804.01508.

Granmo, O.-C., Glimsdal, S., Jiao, L., Goodwin, M., Omlin, C. W., & Berge, G. T. (2019). The convolutional tsetlin machine. arXiv preprint arXiv:1905.09688.

Hancock, B., Bringmann, M., Varma, P., Liang, P., Wang, S., & Ré, C. (2018). Training classifiers with natural language explanations. Paper presented at: Proceedings of the Conference. Association for Computational Linguistics. Meeting (Vol. 2018, p. 1884).

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., & Szpakowicz, S. (2009). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. Paper presented at: Proceedings of the workshop on semantic evaluations: Recent achievements and future directions (pp. 94–99).

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. Paper presented at: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 168–177).

Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1).

Kiritchenko, S., & Mohammad, S. M. (2017). The efwfect of negators, modals, and degree adverbs on sentiment composition. arXiv preprint arXiv:1712.01794.

Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, *50*, 723–762.

Lei, J., Rahman, T., Shafik, R., Wheeldon, A., Yakovlev, A., Granmo, O.-C., Kawsar, F., & Mathur, A. (2021). Low-power audio keyword spotting using Tsetlin machines. *Journal of Low Power Electronics and Applications*, *11*(2), 18.

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

Liu, H., Yin, Q., & Wang, W. Y. (2018). Towards explainable nlp: A generative explanation framework for text classification. arXiv preprint arXiv:1811.00196.

Mohammad, S. M. (2018). Word affect intensities. Paper presented at: Proceedings of the 11th edition of the Language Resources and Evaluation Conference.

Mohammad, S. M., & Turney, P. D. (2013). Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, *29*(3), 436–465.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Interpretable machine learning: Definitions, methods, and applications. arXiv preprint arXiv:1901.04592.

Nielsen, F. Å. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. arXiv preprint arXiv:1103.2903.

Phoulady, A., Granmo, O.-C., Gorji, S. R., & Phoulady, H. A. (2019). The weighted tsetlin machine: Compressed representations with clause weighting. arXiv preprint arXiv:1911.12607.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" explaining the predictions of any classifier. Paper presented at: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135–1144).

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

Saha, R., Granmo, O.-C., & Goodwin, M. (2020). Mining interpretable rules for sentiment and semantic relation analysis using Tsetlin machines. Paper presented at: International Conference on Innovative Techniques and Applications of Artificial Intelligence (pp. 67–78).

Saha, R., Granmo, O.-C., Zadorozhny, V. I., & Goodwin, M. (2021). A relational tsetlin machine with applications to natural language understanding. arXiv preprint arXiv:2102.10952.

Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

Shafik, R., Wheeldon, A., & Yakovlev, A. (2020). Explainability and dependability analysis of learning automata based ai hardware. Paper presented at: In 2020 IEEE 26th International Symposium on On-Line Testing and Robust System Design (IOLTS). (pp. 1–4).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. Paper presented at: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631–1642).

Tausczik, Y. R., & Pennebaker, J. W. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, *29*(1), 24–54.

Tsetlin, M. L. (1961). On behaviour of finite automata in random medium. *Avtom I Telemekhanika*, *22*(10), 1345–1354.

Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., & MacNeille, P. (2017). A Bayesian framework for learning rule sets for interpretable classification. *The Journal of Machine Learning Research*, *18*(1), 2357–2393.

Wheeldon, A., Shafik, R., Yakovlev, A., Edwards, J., Haddadi, I., & Granmo, O.-C. (2020). Tsetlin machine: A new paradigm for pervasive ai. Paper presented at: Proceedings of the Scona Workshop at Design, Automation and test in Europe (date).

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. Paper presented at: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (pp. 347–354).

Xuelan, F., & Kennedy, G. (1992). Expressing causation in written english. *RELC Journal*, *23*(1), 62–80.

Yadav, R. K., Jiao, L., Granmo, O.-C., & Goodwin, M. (2021a). Human-level interpretable learning for aspect-based sentiment analysis. Paper presented at: The thirty-fifth AAAI Conference on Artificial Intelligence (AAAI-21).

Yadav, R. K., Jiao, L., Granmo, O.-C., & Goodwin, M. (2021b). Interpretable classification of word sense disambiguation using Tsetlin machine. Paper presented at: 13th International Conference on Agents and Artificial Intelligence (ICAART 2021).

AUTHOR BIOGRAPHIES

Rupsa Saha received her MTech degree in information and communication technology with specialization in machine intelligence from DAIICT, India in 2014. She is currently pursuing her PhD on Tsetlin Machines with the Centre for Artificial Intelligence Research, University of Agder, Norway. Her research interests include machine learning, NLP and chatbots.

Ole-Christoffer Granmo is a Professor and Founding Director of Centre for Artificial Intelligence Research (CAIR), University of Agder, Norway. He obtained his master's degree in 1999 and the PhD degree in 2004, both from the University of Oslo, Norway. Dr Ole-Christoffer Granmo has authored in excess of 140 refereed papers with 6 best paper awards, encompassing learning automata, bandit algorithms, Tsetlin Machines, Bayesian reasoning, reinforcement learning, and computational linguistics. He has further coordinated 7+ Norwegian Research Council projects and graduated more than 60 master- and PhD students. Dr Ole-Christoffer Granmo is also a co-founder of the Norwegian Artificial Intelligence Consortium (NORA). Apart from his academic endeavours, he co-founded the company Anzyz Technologies AS.

Morten Goodwin received the BSc and MSc degrees from the University of Agder, Norway, in 2003 and 2005, respectively, and the PhD degree from Aalborg University Department of Computer Science, Denmark, in 2011, on applying machine learning algorithms on eGovernment indicators, which are difficult to measure automatically. He is a Professor with the Department of ICT, the University of Agder, Deputy director for Centre for Artificial Intelligence Research, a public speaker, and an active researcher. His main research interests include machine learning, including swarm intelligence, deep learning and adaptive learning in medicine, games and chatbots. He has more than 100 peer reviews of scientific publications. He has supervised more than 110 student projects, including Master and PhD theses within these topics, and more than 90 popular science public speaking events, mostly in Artificial Intelligence.

How to cite this article: Saha, R., Granmo, O.-C., & Goodwin, M. (2021). Using Tsetlin Machine to discover interpretable rules in natural language processing applications. *Expert Systems*, e12873. https://doi.org/10.1111/exsy.12873