CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

ConvTextTM: An Explainable Convolutional Tsetlin Machine Framework for Text Classification

BIMAL BHATTARAI

University of Agder

UiA University of Agder Norway

Introduction

- Recent advances in AI often come with increased computational complexity and decreased transparency.
- Traditional models like Linear Regression, Logistic Regression, and Decision Trees can be highly interpretable but are far from achieving state-of-art accuracy.
- Human-level interpretability is achieved in TM essentially through the use of the bag-of-words (BOW) approach.
- ConvTextTM breaks down the text into a sequence of text fragments
- ConvTextTM eliminates the dependency on a corpus-specific vocabulary.

Background

- Interpretable machine learning has a long history dating back to Breiman's research on decision trees and random forests.
- Several approaches for explaining models prediction but not interpretability.
- Few works on global and local interpretability using deep learning methods.
- Recent rule-based approach such as TM is successful with interpretable learning and explainable logic.

Tsetlin Machine

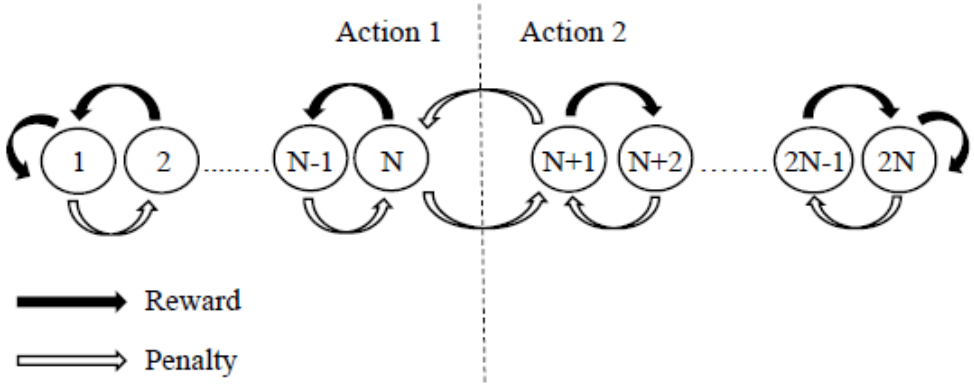
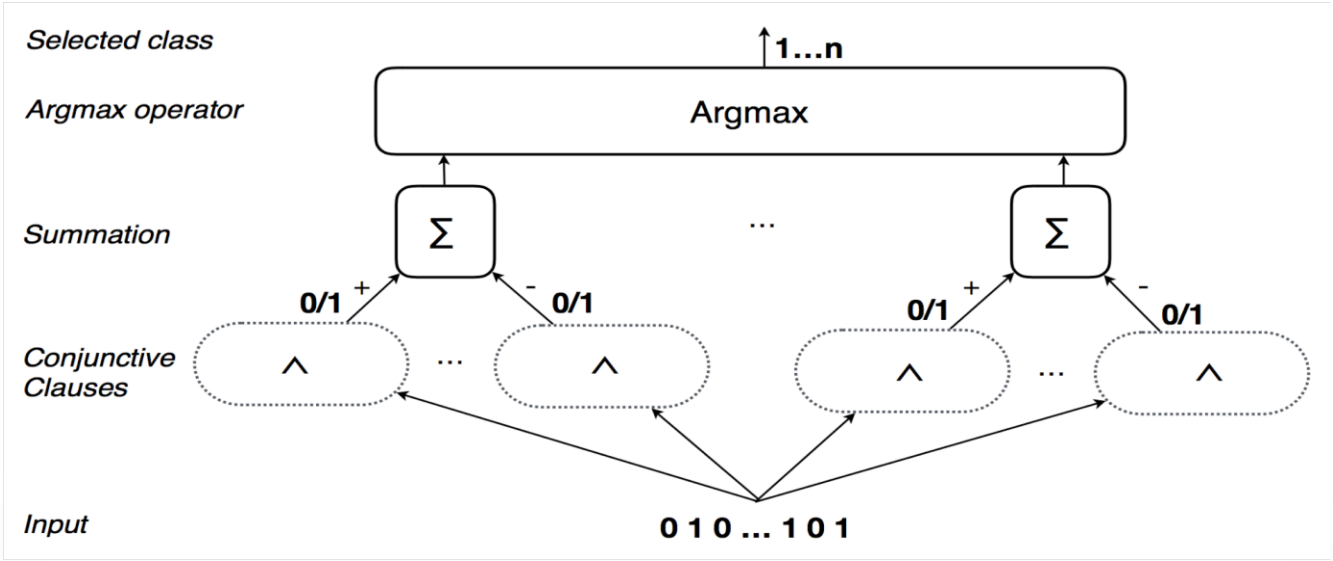


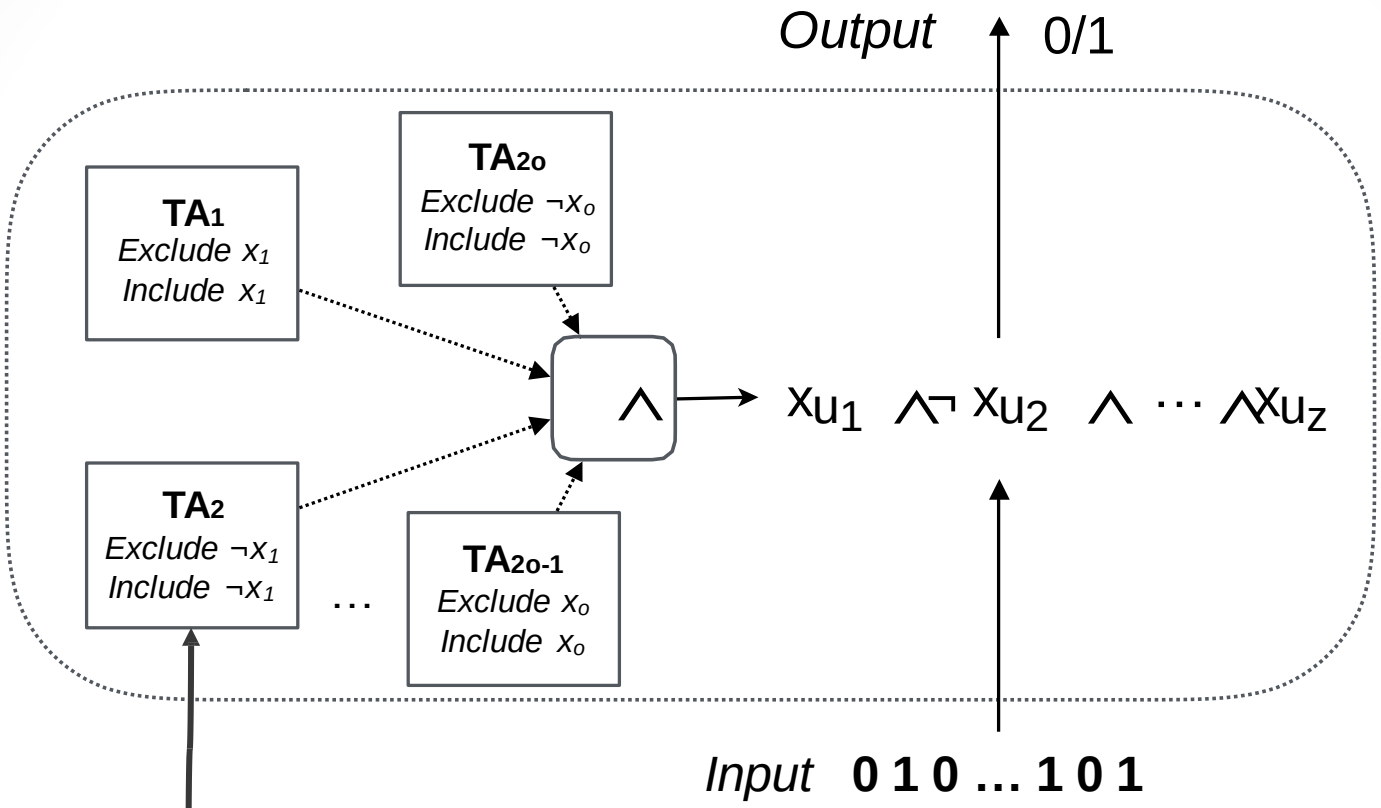
Fig. 1: Transition graph of a two-action Tsetlin Automaton.

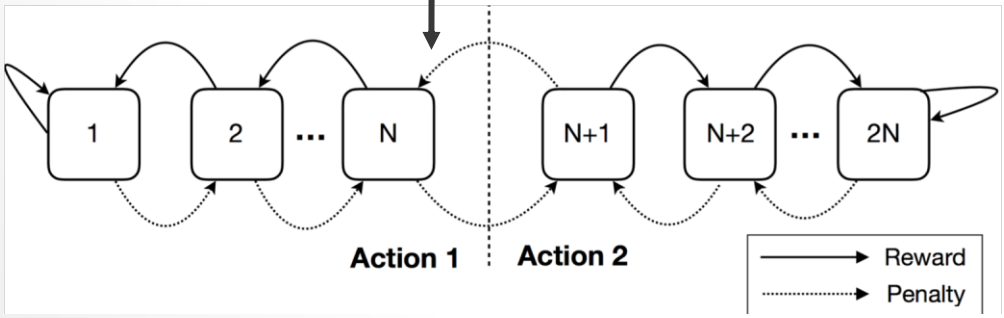


Why Tsetlin Machine?

- Tsetlin Machine (TM) is a recent rule-based approach to solve tasks like pattern recognition and data regression.
- TM has promising properties regarding computational simplicity, transparency and interpretability, when compared to deep learning.
- TM has previously performed well in some natural language processing (NLP) applications.

TM clause formation





TM Game

1. The arrival of a labelled object ***(X, y)*** start of a new game round.

2. Each Tsetlin Automaton decides whether to include or exclude its designated literal, leading to a new configuration of clauses $\boldsymbol{C}$.

3. Each clause, $\boldsymbol{C_j \in C}$, is then evaluated with $\boldsymbol{X}$ as input.

4. The final output, $\boldsymbol{y}$, of the Tsetlin Machine is decided and compared with the target output y.

5. Each Tsetlin Automaton is independently and randomly given either Reward, Inaction, Penalty feedback, based on a novel game matrix.

TM Formulation

Input $$X = (x_1, \ldots, x_o)$$

Literal set $$L = \{x_1, \ldots, x_o, \bar{x}_1, \ldots, \bar{x}_o\}$$

Clause formation $$C_j^+(X) = \bigwedge_{l_k \in L_j^+} l_k = \prod_{l_k \in L_j^+} l_k.$$

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

XOR case $$\hat{y} = u\left(x_1\bar{x}_2 + \bar{x}_1 x_2 - x_1 x_2 - \bar{x}_1\bar{x}_2\right)$$

ConvTextTM Formulation

Input

$$X = (x_k) \in \{0,1\}^{\mathcal{A}\times\mathcal{B}\times\mathcal{C}}$$

Patches

$$P = \frac{\mathcal{B}-1}{q} + 1$$

Augmented Input

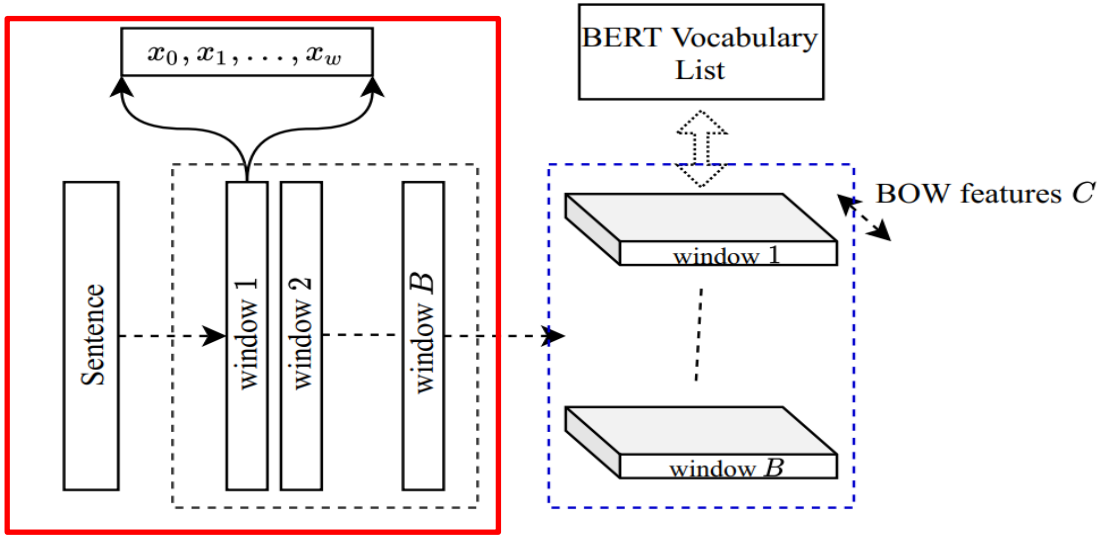
$$\mathbf{x}^p = [x_k^p] \in \{0,1\}^{\bar{\mathcal{C}}}, p \in \{1,2,\ldots,P\}$$

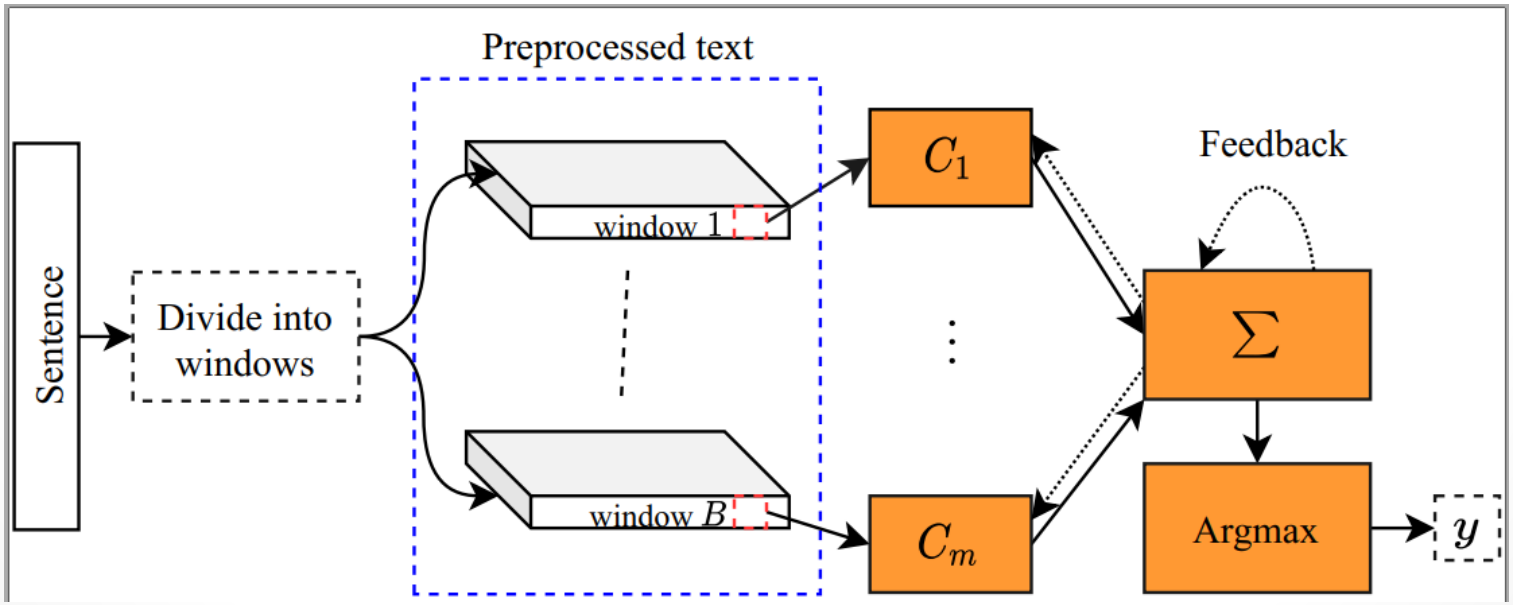
$$\mathbf{x}^p : x^p = [x_k^p] \in \{0,1\}^{\bar{\mathcal{C}}+P}$$

Output

$$\hat{y} = u\left[\sum_{j=1}^{m/2}\bigvee_{p=1}^{P}\left(\bigwedge_{l_k\in L_j^+} l_k^p\right) - \sum_{j=1}^{m/2}\bigvee_{p=1}^{P}\left(\bigwedge_{l_k\in L_j^-} l_k^p\right)\right]$$

Text Preprocessing





Datasets Configurations

Dataset name	Train set size	Test set size	Label
PolitiFact	716	238	2
GossipCop	15,175	5,058	2
BBCSports	517	220	5
Twitter	2,176	932	3
Query	17,500	3,850	2
R8	5,478	2,189	8
WOS-5736	4,588	1,148	11

Table 1: Dataset Statistics.

Datasets	Epochs	#clauses	Threshold (T)	Sensitivity (s)	Window
PolitiFact	200	10,000	100*100	10.0	3
GossipCop	200	15,000	100*100	20.0	3
BBCSports	100	10,000	150*150	10.0	2
Twitter	100	10,000	50*50	10.0	2
Query	100	5,000	150*150	10.0	2
R8	150	10,000	100*100	10.0	2
WOS-5736	150	10,000	150*150	10.0	2

Table 2: Hyperparameter configurations.

Results

Datasets	RST	LIWC	HAN	CNN-text	LSTM-ATT	RoBERTa-MWSS	BERT	XLNet	TM	TM_{conv}	$TM_{conv}(max)$
PolitiFact	60.7	76.9	83.7	65.3	83.3	82.5	88	89.5	87.1±0.24	90.27±0.33	91.21
GossipCop	53.1	73.6	74.2	73.9	79.3	80.3	85	85.5	84.2±0.03	85.82±0.27	86.28

Table 3: Performance comparison of our model with other baseline models for fact checking.

Datasets	WMD	Deepsets	NNattention	Transformer	LSTM	BERT	XLNet	TM	TM_{conv}	$TM_{conv}(max)$
BBCSports	95.40± 0.70	74.55±20.1	95.28±0.97	95.82±1.23	95.52	99	98	96.91	96.78 ± 0.32	97.97
Twitter	71.3±0.70	70±1.62	70.91±0.62	72.21±0.47	72.1	74.71	78	71.13	70.67±0.27	71.91

Table 4: Performance comparison of our model with other baseline models for document classification.

Datasets	CNN	Fasttext	LSTM	HAN	BERT	XLNet	Human	TM	TM_{conv}	$TM_{conv}(max)$
Query	67.38	62.1	65.8	64.64	80	77	88.4	53.87	67.43±0.22	67.94
R8	95.71	96.13	96.09	-	97.8	98	-	96.16	96.32±0.11	96.43

Table 5: Performance comparison of our model with other baseline models for short text classification.

Dataset	DNN	CNN	RNN	Stacking SVM	HDLTex-CNN	BERT	XLNet	TM	TM_{conv}	$TM_{conv}(max)$
WOS-5736	86.15	88.68	89.46	85.68	90.93	90.24	90	89.47	90.73±0.27	91.28

Table 6: Performance comparison of our model with other baseline models for academic classification.

Results

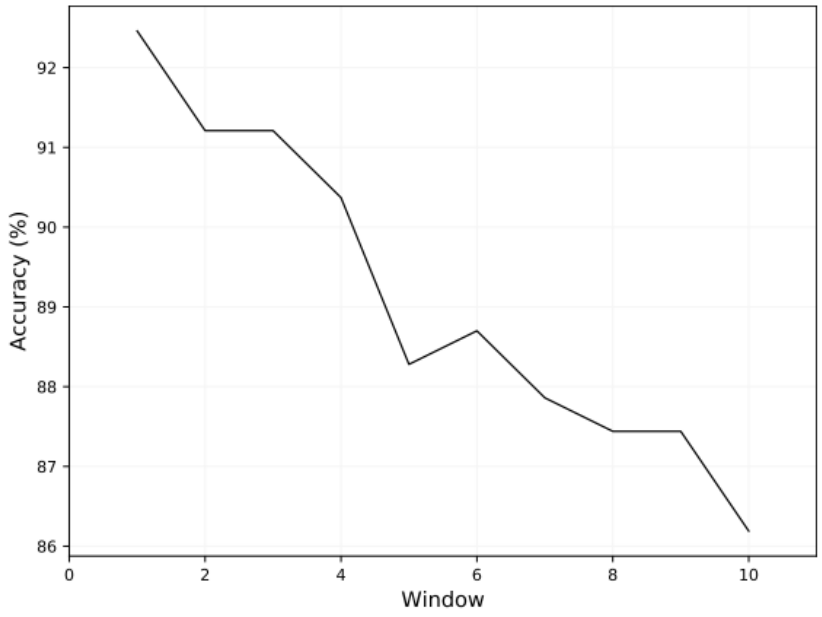


Figure 3: Accuracy *vs* Window.

Dataset	Accuracy (max)
PolitiFact	92.46
GossipCop	85.94
BBCSports	99.54
Twitter	73.67
Query	68.25
R8	96.93
WOS-5736	92.42

Table 8: Maximum accuracy obtained using window size 1 (i.e., $\mathcal{W} = 1$).

Interpretability

acq	crude	earn	grain	interest	money-fx	ship	trade
pointing	##connect	accomplish	island	tesla	kruger	farmhouse	##ities
doping	thorns	##sby	treating	##oko	ashton	road	harlow
##rified	notably	looting	surface	linux	tonight	agency	pianos
evergreen	##enter	##osed	facebook	bed	##sitor	##voking	sanskrit
phone	dramatic	endelle	trail	slams	becker	##met	plunged
premiere	chloride	confrontation	##emia	photo	bed	rockies	##cured
demonstrated	racers	temporarily	summer	mural	burning	trafficking	script
collections	fountain	werewolf	road	vaccines	handicap	likeness	theories
unwilling	##lon	segregation	garbage	families	lightning	flaming	ineffective
##liest	safety	presiding	##nity	bronze	patent	gesellschaft	extensive

Table 7: Significant features captured by our model for each class in R8 dataset.

(a) Window 0.

(b) Window 1.

Figure 4: Wordcloud visualization of local interpretability for "**acq**".

Use Example

Sentence:

"The movie was excellent and well-directed. It was one of the best movies I have ever watched"

Window 1:

["The movie was excellent and well-directed"]

Window 2:

["It was one of the best movies I have ever watched"]

Sentence:

"I need a loan for a house. I do not have a good car"

Window 1:

["I need a loan for a house"]

Window 2:

["I do not have a good car"]

CAIR CENTRE FOR ARTIFICIAL INTELLIGENCE RESEARCH

Thank you!