Generating Hierarchical Explanations on Text Classification via Feature Interaction Detection

Hanjie Chen, Guangtao Zheng, Yangfeng Ji

19 October 2020

Presenter: Sanchit Sinha

https://qdata.github.io/deep2Read/

Motivation

- Interpretability is important for reaffirming - reliability and trustworthiness of models
- Overcoming “black-box” nature of deep learning models
- Ideal goal of any interpretability technique - human understandable explanations
- Hierarchical explanations - superior to simple attributions (proven by prior work)
- Designing a model agnostic explainability approach - to works for any model - LSTM, BERT, etc.

Background

- Model agnostic Explainability:
  - Leave one out
  - LIME
  - Shapley Value based
- Shapley Values:
  - Shapley value of a feature: contribution of that feature to the output

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\;(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S))$$

- Hierarchical Explanations:
  - Proven to be most human friendly
  - ACD (discussed before) uses CD values as distance metric for bottom up clustering
  - Interactions between phrases important

Related Work

- Shapley Interactions between features - Owen, 1972; Grabisch, 1997; Fujimoto et al., 2006
- CD and ACD - Murdoch et al, Singh et al.
- LIME
- Consistent Individualized Feature Attribution for Tree Ensembles - Lundberg et al. (Shap values for trees)
- BERT

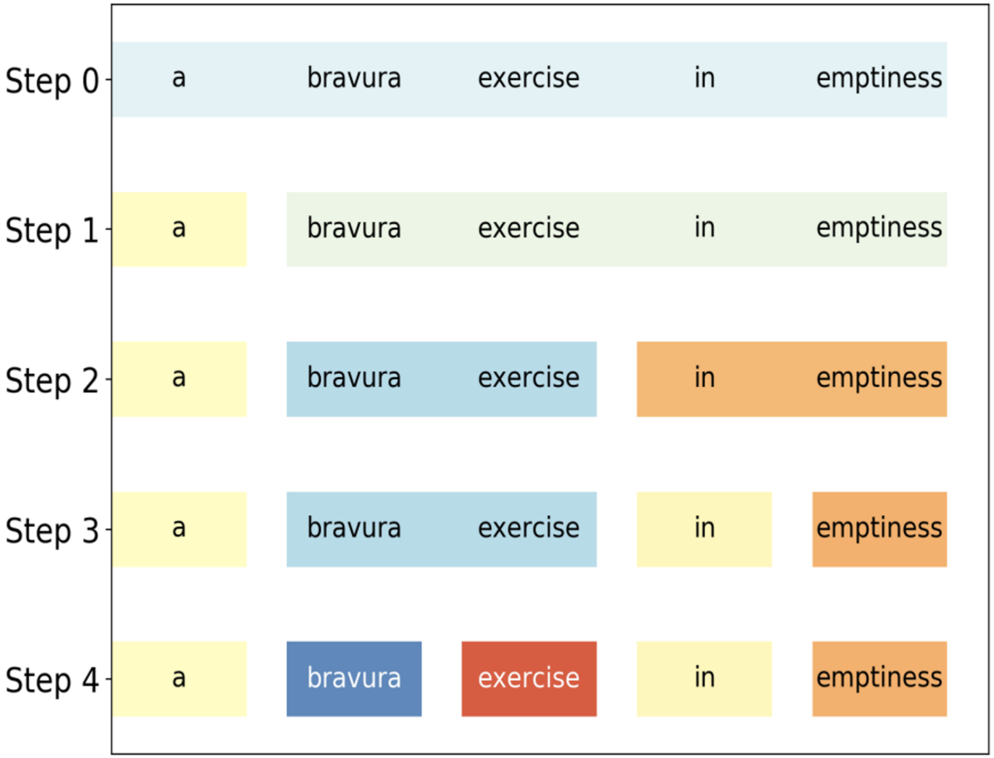
Claim / Target Task

- Designing a model agnostic explainability method
- **Detecting feature interaction**: Use Shapley interaction values between two phrases as the “cut” metric - the lesser the value, the earlier the “cut”.
- **Quantifying feature importance:** Assign importance values to each phrase at all levels
- Improving metrics wrt current SOTA:
    - AOPC
    - Log Odds
    - Cohesion Score

Data Summary

- All datasets are for Text classification
- 10% held out for dev set
- SST-2: Binary Labels - 6920/872/1821 examples in the train/dev/test set
- IMDb : Binary Labels - 25000/25000 examples in the train/test set

An Intuitive Figure Showing WHY Claim



(a) HEDGE for LSTM on the SST.

Proposed Solution

- The intuition of the algorithm can be divided into two parts.
  - Part 1: Finding the "split point" where we divide a phrase into 2 least interacting phrases
  - Part2: Quantifying the impact of a phrase on the prediction

Algorithm 1 Hierarchical Explanation via Divisive Generation

Input: text $\boldsymbol{x}$ with length n, and predicted label $\hat{y}$
Initialize the original partition $\mathcal{P}_0 \leftarrow \{\boldsymbol{x}_{(0,n]}\}$
Initialize the contribution set $\mathcal{C}_0 = \emptyset$
Initialize the hierarchy $\mathcal{H} = [\mathcal{P}_0]$
for $t = 1, \ldots, n-1$ **do** — Levels = words-1
  Find $\boldsymbol{x}_{(s_i, s_{i+1}]}$ and j by solving Equation 1
  Update the partition
  $\mathcal{P}'_t \leftarrow \mathcal{P}_{t-1} \backslash \{\boldsymbol{x}_{(s_i, s_{i+1}]}\}$
  $\mathcal{P}_t \leftarrow \mathcal{P}'_t \cup \{\boldsymbol{x}_{(s_i, j]}, \boldsymbol{x}_{(j, s_{i+1}]}\}$ — Creating partition
  $\mathcal{H}.add(\mathcal{P}_t)$
  Update the contribution set $\mathcal{C}$ with
  $\mathcal{C}'_t \leftarrow \mathcal{C}_{t-1} \cup \{(\boldsymbol{x}_{(s_i, j]}, \psi(\boldsymbol{x}_{(s_i, j]}))\}$
  $\mathcal{C}_t \leftarrow \mathcal{C}'_t \cup \{(\boldsymbol{x}_{(j, s_{i+1}]}, \psi(\boldsymbol{x}_{(j, s_{i+1}]}))\}$
end for
Output: $\mathcal{C}_{n-1}, \mathcal{H}$ — Value, Structure

Proposed Solution

- Part 1: Finding the split -
  - Step 0: The whole sentence is a partition.
  - Step 1: For every phrase pair, calculate the following:

$$\min_{\boldsymbol{x}_{(s_i,s_{i+1}]}\in\mathcal{P}} \min_{j\in(s_i,s_{i+1})} \phi(\boldsymbol{x}_{(s_i,j]}, \boldsymbol{x}_{(j,s_{i+1}]} \mid \mathcal{P}),$$

  - Creating phrase: {word 0, word 1,n} {word 0,1, word 2,n}
  - Selecting the split point where the phi function value is the least
  - Phi value is basically the Shapley value for interaction between 2 phrases
- Calculating the phi function value:
  - Is the same as the Shapley value. Here M-1 is the size of partition after removing 2 interacting phrases

$$\phi(j_1,j_2\,|\,\mathcal{P}) = \sum_{S\subseteq\mathcal{N}_m\setminus\{j_1,j_2\}} \frac{|S|!(M-|S|-2)!}{(M-1)!}\gamma(j_1,j_2,S),$$

Proposed Solution

- Original Shapley:

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n - |S| - 1)!}{n!}(v(S \cup \{i\}) - v(S))$$

- Modified Shapley interaction

$$\begin{aligned}\gamma(j_1, j_2, S) = \mathbb{E}[f(\boldsymbol{x}') \,|\, S \cup \{j_1, j_2\}] - \mathbb{E}[f(\boldsymbol{x}') \,|\, S \cup \{j_1\}] \\ - \mathbb{E}[f(\boldsymbol{x}') \mid S \cup \{j_2\}] + \mathbb{E}[f(\boldsymbol{x}') \mid S],\end{aligned}$$

- The size of neighbourhood (M) is set to a smaller value due to words having highest interaction with words before or after the most rather than far away.
- This means that complexity for each word is fixed at +/- 2 which is polynomial, but if the whole sentence is taken the complexity multiplies by ‘n’ (size of sentence)

Proposed Solution

- Part 2 - Quantifying feature importance
  - Basically how far away from the decision boundary is our prediction
  - The farther away, the more important the feature to the prediction value

$$\psi(\boldsymbol{x}_{(s_i,s_{i+1}]}) = f_{\hat{y}}(\boldsymbol{x}_{(s_i,s_{i+1}]}) - \max_{y' \neq \hat{y}, y' \in \mathcal{Y}} f_{y'}(\boldsymbol{x}_{(s_i,s_{i+1}]}),$$

  - This makes the algorithm model agnostic

Experimental Results - Metrics

- AOPC - "average change in the prediction probability on the predicted class over all test data" - Remove top k% words and then measure the drop in performance. Higher is better

$$\mathrm{AOPC}(k) = \frac{1}{N}\sum_{i=1}^{N}\{p(\hat{y} \mid \boldsymbol{x}_i) - p(\hat{y} \mid \tilde{\boldsymbol{x}}_i^{(k)})\},$$

- Log Odds - "averaging the difference of negative logarithmic probabilities on the predicted class over all of the test data before and after masking the top r% features with zero paddings" - occlusion. Lower is better

$$\text{Log-odds}(r) = \frac{1}{N}\sum_{i=1}^{N}\log\frac{p(\hat{y} \mid \tilde{\boldsymbol{x}}_i^{(r)})}{p(\hat{y} \mid \boldsymbol{x}_i)}.$$

- Cohesion Score - Permuting each word of the sentence and calculate interactions. Higher is better.

$$\text{Cohesion-score} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}\sum_{q=1}^{Q}(p(\hat{y} \mid \boldsymbol{x}_i) - p(\hat{y} \mid \bar{\boldsymbol{x}}_i^{(q)}))$$

Experimental Results - AOPC/Log Odds

Datasets	Methods	LSTM		CNN		BERT	
		AOPC	Log-odds	AOPC	Log-odds	AOPC	Log-odds
SST	Leave-one-out	0.441	-0.443	0.434	-0.448	0.464	-0.723
	CD	0.384	-0.382	-	-	-	-
	LIME	0.444	-0.449	0.473	-0.542	0.134	-0.186
	L-Shapley	0.431	-0.436	0.425	-0.459	0.435	-0.809
	C-Shapley	0.423	-0.425	0.415	-0.446	0.410	-0.754
	KernelSHAP	0.360	-0.361	0.387	-0.423	0.411	-0.765
	SampleShapley	0.450	-0.454	0.487	-0.550	0.462	-0.836
	HEDGE	**0.458**	**-0.466**	**0.494**	**-0.567**	**0.479**	**-0.862**
IMDB	Leave-one-out	0.630	-1.409	0.598	-0.806	0.335	-0.849
	CD	0.495	-1.190	-	-	-	-
	LIME	0.764	-1.810	0.691	-1.091	0.060	-0.133
	L-Shapley	0.637	-1.463	0.623	-0.950	0.347	-1.024
	C-Shapley	0.629	-1.427	0.613	-0.928	0.331	-0.973
	KernelSHAP	0.542	-1.261	0.464	-0.727	0.223	-0.917
	SampleShapley	0.757	-1.597	0.707	-1.108	0.355	-1.037
	HEDGE	**0.783**	**-1.873**	**0.719**	**-1.144**	**0.411**	**-1.126**

Experimental Results - Cohesion

Methods	Models	Cohesion-score	
		SST	IMDB
HEDGE	CNN	0.016	0.012
	BERT	0.124	0.103
	LSTM	0.020	0.050
ACD	LSTM	0.015	0.038

Experimental Results - Coherence/AMT

Methods	Coherence Score
Leave-one-out	0.82
ACD	0.68
LIME	0.85
L-Shapley	0.75
C-Shapley	0.73
KernelSHAP	0.56
SampleShapley	0.78
HEDGE	**0.89**

Table 4: Human evaluation of different interpretation methods with LSTM model on the IMDB dataset.

Models	Accuracy	Coherence scores
LSTM	0.87	0.89
CNN	0.90	0.84
BERT	0.97	0.75

Table 5: Human evaluation of HEDGE with different models on the IMDB dataset.

Experimental Analysis

- **Explanation method works well**
  - Works better than LIME, ACD, Shapley
- **BERT gives very high accuracy but is not very interpretable**