Towards Hierarchical Importance Attribution: Explaining Compositional Semantics for Neural Sequence Models

-Xisen Jinx, Junyi Dux, Zhongyu Weiy, Xiangyang Xuey, Xiang Ren

ICLR 2020

March 20, 2020

Presenter: Rishab Bamrara

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

Motivation

- Existing flat, word level explanations of predictions hardly unveil how neural networks handle compositional semantics to reach predictions.

- Hence, we go for Hierarchical explanation systems like CD/ACD.

- The key challenge => context independent importance of a phrase.

Related Work

Interpretability of neural networks has been studied with various techniques:

- Tenney et al. (2019): probing learned features with auxiliary tasks
- Bahdanau et al., (2015): designing models with inherent interpretability.
- Kadar et al., (2017): Input occlusion
- Sundararajan et al., (2017): Additive feature attribution methods prediction
- Murdoch et. al., (2018): CD
- Singh et. al., (2019): ACD

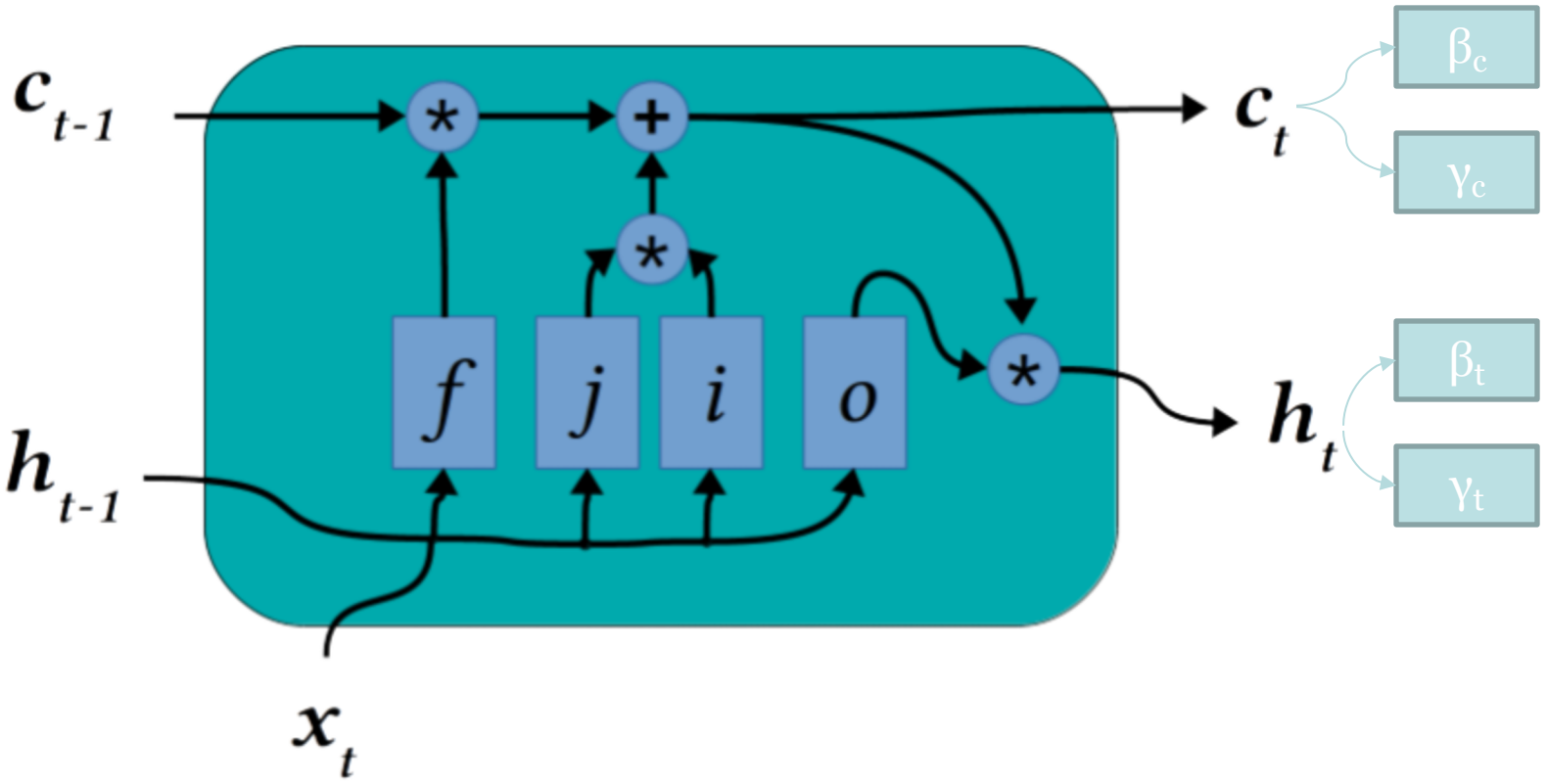
Background

Additive feature attribution methods => explain a model prediction by assigning importance scores to individual input variables.

- [Drawback: explaining compositional semantics]

CD/ACD => **Go beyond the additive assumption** and compute the contribution solely made by a word/phrase to the model prediction.

Background (Contd.)



https://blog.exxactcorp.com/5-types-lstm-recurrent-neural-network/

Background (Contd.)

$$
\begin{aligned}
h_t &= o_t \odot \tanh(c_t) \\
&= o_t \odot [L_{\tanh}(\beta_t^c) + L_{\tanh}(\gamma_t^c)] \\
&= o_t \odot L_{\tanh}(\beta_t^c) + o_t \odot L_{\tanh}(\gamma_t^c) \\
&= \beta_t + \gamma_t
\end{aligned}
$$

CD computes the contribution solely from the phrase p as the average activation differences caused by assuming is present or absent as:

$$
\boldsymbol{\beta}' = \frac{1}{2}[\sigma(\boldsymbol{\beta} + \boldsymbol{\gamma}) - \sigma(\boldsymbol{\gamma})] + \frac{1}{2}[\sigma(\boldsymbol{\beta}) - \sigma(0)] \tag{2}
$$

Background (Contd.)

```
        o = sigmoid(np.dot(W_io, word_vecs[i]) + np.dot(W_ho, prev_rel_h + prev_irrel_h) + b_o)
        rel_contrib_o, irrel_contrib_o, bias_contrib_o = decomp_three(rel_o, irrel_o, b_o, sigmoid)
        new_rel_h, new_irrel_h = decomp_tanh_two(relevant[i], irrelevant[i])
        #relevant_h[i] = new_rel_h * (rel_contrib_o + bias_contrib_o)
        #irrelevant_h[i] = new_rel_h * (irrel_contrib_o) + new_irrel_h * (rel_contrib_o + irrel_contrib_o + bias_contrib_o)
        relevant_h[i] = o * new_rel_h
        irrelevant_h[i] = o * new_irrel_h

    W_out = model.hidden_to_label.weight.data

    # Sanity check: scores + irrel_scores should equal the LSTM's output minus model.hidden_to_label.bias
    scores = np.dot(W_out, relevant_h[T - 1])
    irrel_scores = np.dot(W_out, irrelevant_h[T - 1])

    return scores, irrel_scores

def decomp_three(a, b, c, activation):
    a_contrib = 0.5 * (activation(a + c) - activation(c) + activation(a + b + c) - activation(b + c))
    b_contrib = 0.5 * (activation(b + c) - activation(c) + activation(a + b + c) - activation(a + c))
    return a_contrib, b_contrib, activation(c)

def decomp_tanh_two(a, b):
    return 0.5 * (np.tanh(a) + (np.tanh(a + b) - np.tanh(b))), 0.5 * (np.tanh(b) + (np.tanh(a + b) - np.tanh(a)))
```

https://github.com/jamie-murdoch/ContextualDecomposition

Background (Contd.)

CD evaluating interactions between two neighboring phrases:

$$\mathcal{I}(\mathbf{p}_1, \mathbf{p}_2) = \phi(\mathbf{p}_1; \mathbf{p}_2, \mathbf{x}) - [\phi(\mathbf{p}_1, \mathbf{x}) + \phi(\mathbf{p}_2, \mathbf{x})] \tag{3}$$

where p1; p2 notes for concatenation of two phrases.

Similar in its form to marginal interactions in cooperative game theory, where each word corresponds to a player, and each phrase corresponds to a coalition of players.

Background (Contd.)

Given a set of context words S, the *marginal interaction* between p1 and p2 is defined as,

$$\mathcal{I}_S(\mathbf{p}_1, \mathbf{p}_2) = v(S \cup \mathbf{p}_1; \mathbf{p}_2) - v(S) - [v(S \cup \mathbf{p}_1) - v(S) + v(S \cup \mathbf{p}_2) - v(S)] \quad (4)$$

Eq. 3 can be interpreted as marginal interactions if Ø(p; x) could correspond to the term v(S ∪ p) - v(S) in Eq. 4.

However in CD/ACD, assigned importance scores depend on all other words in the sentence mathematically.

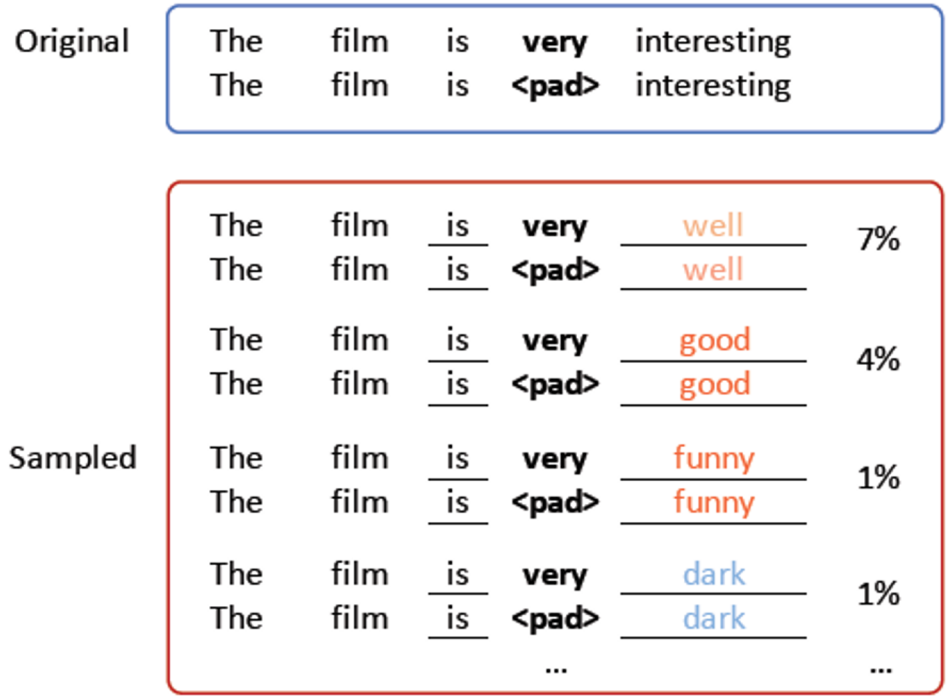
Claim / Target Task

- Propose a mathematically sound way to quantify context independent importance of words and phrases for hierarchical explanations.
- Based on the formulation, develop two effective hierarchical explanation algorithms, namely SCD and SOC.

Proposed Solution

N-Context Independent Importance: Defined as the output difference after masking out the phrase p, marginalized over all the possible N-word contexts, denoted as $x_{\delta}^{\wedge}$, around p in the input x.

Eg. The film is ***very*** interesting. (1-word context)



Proposed Solution (Contd.)

Context independent importance is formally written as:

$$\phi(\mathbf{p}, \mathbf{x}) = \mathbb{E}_{\hat{\mathbf{x}}_\delta}[s(\mathbf{x}_{-\delta}; \hat{\mathbf{x}}_\delta) - s(\mathbf{x}_{-\delta} \backslash \mathbf{p}; \hat{\mathbf{x}}_\delta)], \quad (5)$$

$\mathbf{x_{-\delta}}$ => resulting sequence after masking out an N-word context surrounding the phrase p from the input x.

$\mathbf{x_{\delta}}$^ => N-word sequence sampled from a distribution $p(x_\delta ^ / x_{-\delta})$.

$\mathbf{s(x_{-\delta};x_{\delta}^)}$ => model prediction score after replacing the original context words $x_{-\delta}$ with a sampled N-word context x_δ^.

x\p => the operation of masking out the phrase p from the input sentence x.

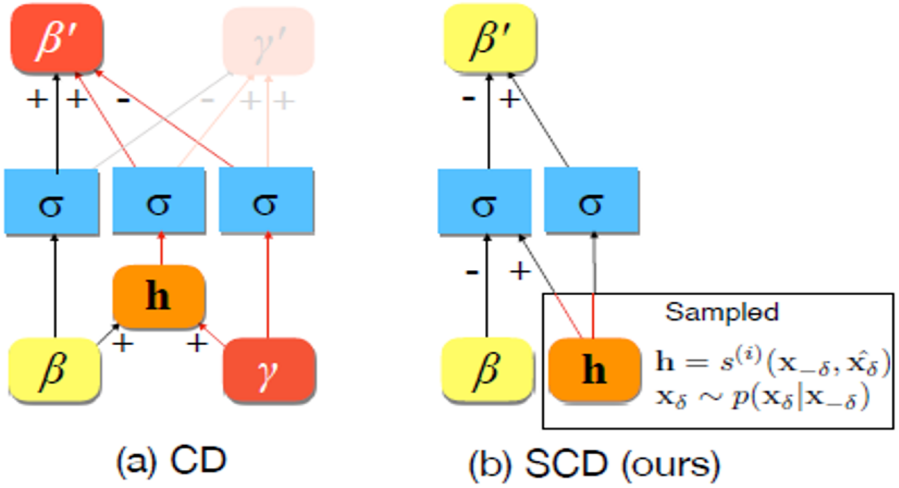
Proposed Solution (Contd.)

Sampling and Contextual Decomposition (SCD)

To eliminate the dependence, modified the activation decomposition step in CD:

$$\begin{aligned}\beta' &= \mathbb{E}_\gamma[\sigma(\beta+\gamma)-\sigma(\gamma)] \\ &= \mathbb{E}_{\mathbf{h}}[\sigma(\mathbf{h})-\sigma(\mathbf{h}-\beta)],\end{aligned} \tag{7}$$

By taking an expectation over γ, the dependence is eliminated. The masking operation x\p is implemented as calculating h-β.



(a) CD (b) SCD (ours)

Implementation

Algorithm:

1. Sampling a set of N-word contexts of the phrase p with a LSTM language model of both directions pretrained on the training set. (Gibbs sampling)
2. Replace the specific N-word context of the phrase p in the input x with sampled ones and feed each of them into the classifier model.
3. Record the inputs of each activation functions for each of input sequence.
4. The decomposition of the i-th nonlinear activation function is calculated as,

$$\beta' = \frac{1}{|\mathcal{S}_{\mathbf{h}}^{(i)}|} \sum_{\mathbf{h} \in \mathcal{S}_{\mathbf{h}}^{(i)}} [\sigma(\mathbf{h}) - \sigma(\mathbf{h} - \beta)] \qquad (8)$$

Proposed Solution (Contd.)

Sampling and Occlusion (SOC):

Input occlusion algorithms calculate the importance of p specific to an input example x by observing the prediction difference after replacing the phrase p with padding tokens, noted as O_p.

$$\phi(\mathbf{p}, \mathbf{x}) = s(\mathbf{x}) - s(\mathbf{x}_{-\mathbf{p}}; \mathbf{0}_{\mathbf{p}}) \qquad (9)$$

However, this importance score is also dependent on all the context words of p in x.

SOC calculates the average prediction difference after masking the phrase for each replacement of neighboring words in the input example.

Implementation

Algorithm:

1. Sample N-word context of the given phrase p with the trained language model.
2. For each new sample, compute the model prediction differences after replacing the phrase p with padding tokens.
3. The importance Ø(p; x) is then calculated as the average prediction differences.

$$\phi(\mathbf{p}, \mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x}_\delta \in \mathcal{S}} [s(\mathbf{x}_{-\delta}; \mathbf{x}_\delta) - s(\mathbf{x}_{-\{\delta,\mathbf{p}\}}; \mathbf{x}_\delta; \mathbf{0}_\mathbf{p})]. \tag{10}$$

MODEL AGNOSTIC

Data Summary

- **Stanford Sentiment Treebank (SST):** Standard NLP benchmark which consists of movie reviews ranging from 2 to 52 words long. In addition to labels of reviews, it also has labels for each phrase in the review.
- **Yelp Polarity:** This was obtained from the Yelp Dataset Challenge. It has train and test sets of sizes 560,000 ad 38,000 respectively. Average length of review is 160.1 words. It contains only review labels.
- **TACRED relation extraction dataset:** Contains 106,264 examples which cover 41 relation types.

Experimental Results and Analysis

Hierarchical Visualization of Important Words and Phrases:

Dataset	SST-2				Yelp Polarity		TACRED	
Model	BERT		LSTM		BERT	LSTM	BERT	LSTM
Metric	word ρ	phrase ρ	word ρ	phrase ρ	word ρ	word ρ	word ρ	word ρ
Input Occlusion	0.2229	0.4081	0.6489	0.4899	0.3781	0.6935	0.7646	0.5756
Direct Feed	0.2005	0.4889	0.6798	0.5588	0.3875	0.7905	0.1986	0.5771
GradSHAP	0.5073	0.5991	0.7024	0.5402	0.5791	0.7388	0.2965	0.6651
CD	0.2334	0.3068	0.6231	0.4727	0.2645	0.7451	0.0052	0.6508
ACD	0.3053	0.3698	0.2495	0.1856	0.3010	0.5024	0.2027	0.0291
Statistic	0.5223	0.4741	**0.7271**	0.4959	0.7294	**0.9094**	0.5324	**0.7662**
SCD	0.5481	0.6015	0.7151	**0.5664**	0.7180	0.7793	0.7980	0.6823
SOC	**0.6265**	**0.6628**	0.7226	0.5649	0.6971	0.7683	**0.7982**	0.7354

Table 1: Correlation between word & phrase importance attribution and linear model coefficients & SST-2 human annotations, achieved by baselines and our explanation algorithms.

word ρ: Pearson correlation between the coefficients learned by a linear bag-of-words model and the importance scores attributed by explanation methods. (Used in CD)

Generally, explanation algorithms that follow proposed formulations achieve highest word and phrase ρ for all the datasets and models. The corpus statistic based approximation of the context independent importance yields competitive words, but it is not competitive for phrase.

Experimental Results and Analysis

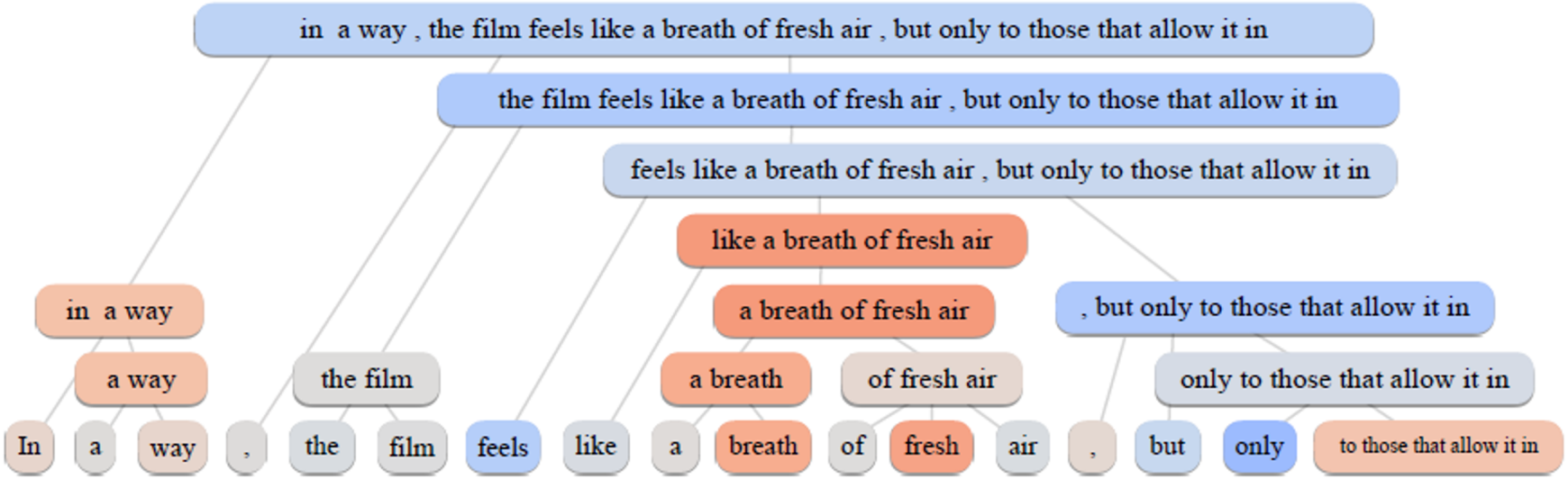
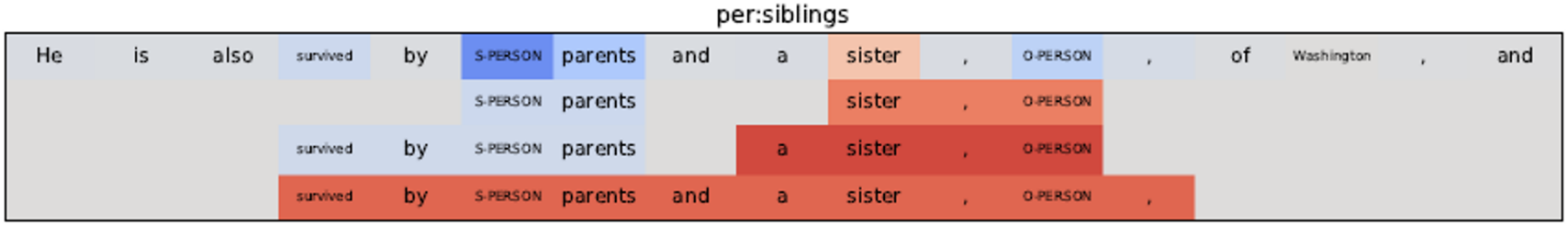


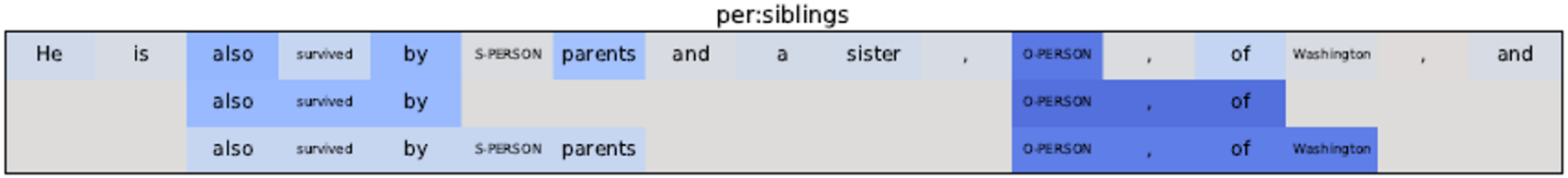
Figure 3: Hierarchical Explanation of a prediction made by the BERT Transformer model on SST-2. We generate explanations for all the phrases on the truncated constituency parsing tree, where positive sentiments are colored red and negative sentiments are colored blue. We see our method identify positive segments in the overall negative sentence, such as "a breath of fresh air"

Experimental Results and Analysis(Contd.)

Explanation as Classification Pattern Extraction from Models:



(a) SCD

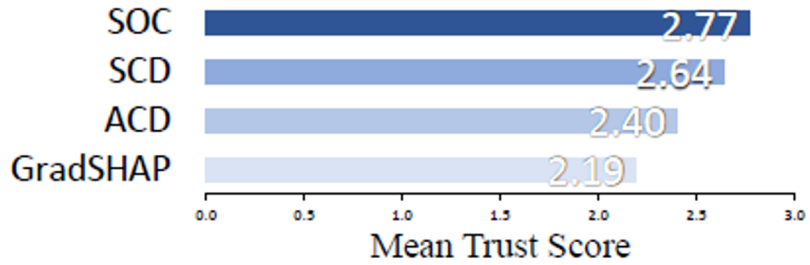


(b) CD

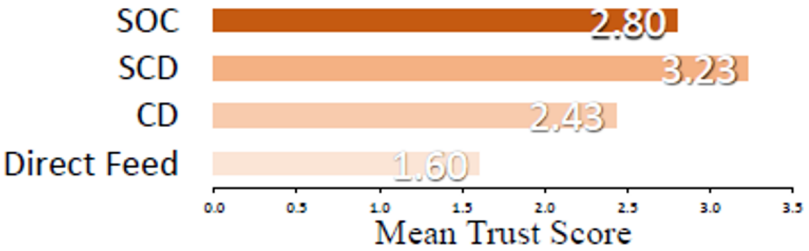
Figure 4: Extracting phrase-level classification patterns from LSTM relation extraction model with SCD. Red indicate evidence for predicting the class, and blue indicate distractor for predicting the class. By applying the agglomerative clustering algorithm and defining a threshold score, SCD effectively extract "a sister, O-Person" as a classification rule for the relation per:siblings.

Experimental Results and Analysis(Contd.)

Enhancing Human Trust of Models: Asked subjects to rank the provided visualizations based on how they would like to trust the model.



(a) SST-2

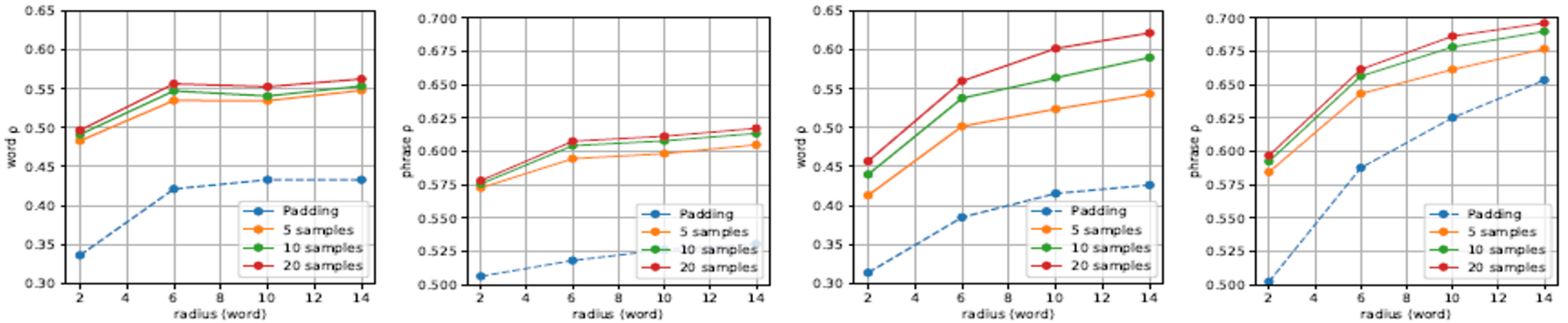


(b) TACRED

Figure 6: Human evaluation results on the Transformer model trained on SST-2 dataset and the LSTM model on TACRED dataset.

Experimental Results and Analysis(Contd.)

Parameter Analysis: Both SOC and SCD algorithms require specifying the size of the context region N and the number of samples K.



(a) SCD, word ρ, BERT (b) SCD, phrase ρ, BERT (c) SOC, word ρ, BERT (d) SOC, phrase ρ, BERT

Figure 5: Word ρ and phrase ρ curve as the size of context regions N and the number of samples K change on the BERT model trained on the SST-2 dataset. Dashed line notes for padding the context of the phrase p instead of sampling the context.

(1) sampling achieves better word and phrase ρ than padding
(2) the performance generally improves as the number of samples increase
(3) the performance improves as the size of the context region N increases at the early stage, and saturates when N grows large. It verifies words or phrases usually do not interact with the words that are far away them in the input.

Conclusion and Future Work

- Authors identify context-independence as a desirable property for bottom-up hierarchical explanation, and propose a formulation to quantify context independent importance of words and phrases.

- Experiments show that the proposed explanation algorithms generate reliable hierarchical explanations, and apply to explanation of compositional semantics, extraction of classification rules as well as improving human trust of models.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
- W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from lstms. In ICLR.
- W James Murdoch and Arthur Szlam. Automatic rule extraction from long short term memory networks. ICLR, 2017.
- Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. CoRR, abs/1612.08220, 2016. URL http://arxiv.org/abs/1612.08220.
- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. CoRR, abs/1703.01365, 2017. URL http://arxiv.org/abs/1703.01365.
- Chandan Singh, W James Murdoch, and Bin Yu. 2019. Hierarchical interpretations for neural network predictions. In ICLR.
- Lloyd S Shapley. 1997. A value for n-person games. Classics in game theory, page 69.