Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs

W. James Murdoch, Peter J. Liu, and Bin Yu

ICLR 2018

Presented by: Dhruv Kumar

For: STAT-946

Outline

- Introduction
- Previous Work
- LSTMs
- Main Idea/Approach of the paper
- Experiments
- Conclusion/Thoughts

Introduction/Motivation

- Deep Neural Networks have achieved SOTA in many fields mainly because of their ability to model complex and non-linear interactions.

- Lack interpretability. Considered Black-Box models.

- The paper introduces a decomposition based algorithm for analysing predictions made by LSTMs.

Previous Work

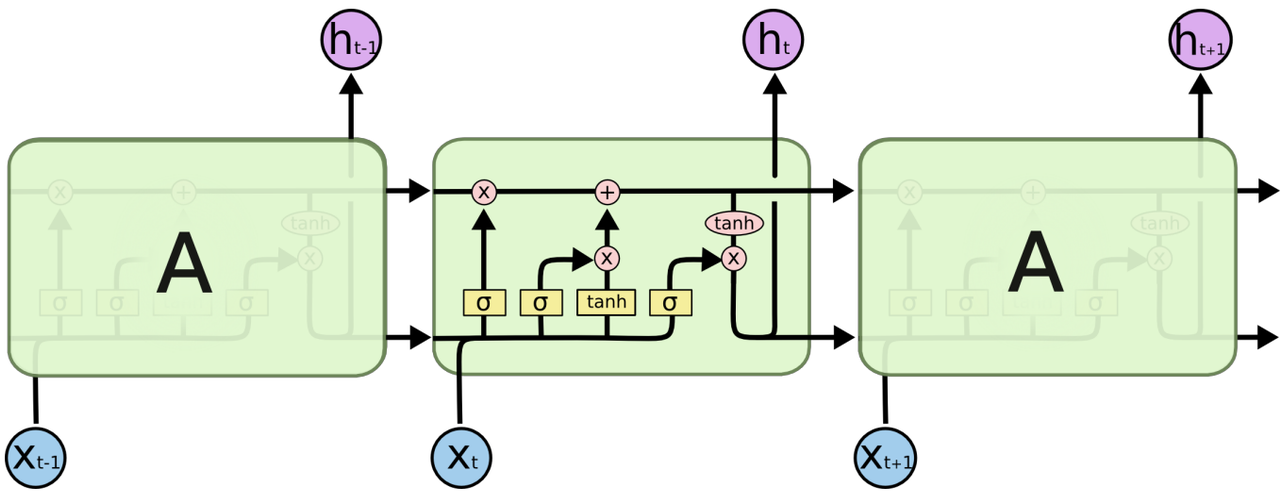
- Computing Word Level Importance Scores
- Decomposition-based for CNNs
- Analysing the gate activations
- Attention-based models (Indirect approach)

LSTMs

- A special kind of RNN which effectively handles long term dependencies.

- Gives a little more control than GRUs.

- Have a memory cell state which runs through the entire network.



- Following are the update equations

$$o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o)$$

$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f)$$

$$i_t = \sigma(W_i x_t + V_i h_{t-1} + b_i)$$

$$g_t = tanh(W_g x_t + V_g h_{t-1} + b_g)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot tanh(c_t)$$

- Intuitively we can think of the forget gate as how much previous memory(information) do we want to forget; input gate as controlling whether or not to let new input in; g gate controlling what do we want to add and finally the output gate as controlling how much the current information(at current time step) should flow out.

- The final hidden state is fed into a multinomial logistic regression.

Contextual Decomposition

- Given a word/phrase/sentence provide a decomposition of the output of a trained LSTM model as a sum of two contributions.
    - Resulting solely from the given phrase
    - Involving atleast in part, elements outside of the phrase
- The method does not change the underlying architecture or the accuracy of the model.

- Let the arbitrary input phrase be $x_q, \ldots, x_r$ where 1≤q≤r≤T, where T represents the length of the sentence. CD decomposes the output and cell state c_t, h_t respectively as

$$h_t = \beta_t + \gamma_t$$

$$c_t = \beta_t^c + \gamma_t^c$$

- Using this decomposition the final softmax output can be written as

$$p = SoftMax(W\beta_T + W\gamma_T)$$

- Mirroring the recursive nature of LSTMs, we recursively compute our decompositions.

Disambiguating Interaction between gates

- Let's assume that the non-linear operations of the gates can be represented in a linear fashion.

$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f)$$
$$= L_\sigma(W_f x_t) + L_\sigma(V_f h_{t-1}) + L_\sigma(b_f)$$

- The products between gates also become linear sums of contributions from the 2 factors mentioned before.
- Here we derive equations for the case when $q \leq t \leq r$

- Terms are determined to derive solely from the specified phrase if they involve products from some combination of $\beta_{t-1}, \beta^c_{t-1}, x_t$ and b_i or b_g (but not both). When t is not within the phrase, products involving x_t are treated as not deriving from the phrase.

$$
\begin{aligned}
f_t \odot c_{t-1} &= (L_\sigma(W_f x_t) + L_\sigma(V_f \beta_{t-1}) + L_\sigma(V_f \gamma_{t-1}) + L_\sigma(b_f)) \odot (\beta^c_{t-1} + \gamma^c_{t-1}) \\
&= ([L_\sigma(W_f x_t) + L_\sigma(V_f \beta_{t-1}) + L_\sigma(b_f)] \odot \beta^c_{t-1}) + (L_\sigma(V_f \gamma_{t-1}) \odot \beta^c_{t-1} + f_t \odot \gamma^c_{t-1}) \\
&= \beta^f_t + \gamma^f_t
\end{aligned}
$$

- Similarly

$$i_t \odot g_t = \beta^u_t + \gamma^u_t$$

- After the decomposition of two components of our memory cell we can sum their contributions

$$
\begin{aligned}
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
&= \beta_t^f + \gamma_t^f + \beta_t^u + \gamma_t^u \\
&= \beta_t^f + \beta_t^u + \gamma_t^f + \gamma_t^u \\
&= \beta_t^c + \gamma_t^c
\end{aligned}
$$

- Now it is relatively easy to compute the cell's output.

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

- Decomposing the output gate like the rest of the gates does not lead to improvements.

Linearizing the activation functions

- The problem that we wish to solve can be represented as shown

$$tanh(\sum_{i=1}^{N} y_i) = \sum_{i=1}^{N} L_{tanh}(y_i)$$

- In cases where y had a natural ordering we could have used the differences of partial sums as a linearisation technique

$$L^{'}_{tanh}(y_k) = tanh(\sum_{j=1}^{k} y_j) - tanh(\sum_{j=1}^{k-1} y_j)$$

- But in our case the terms do not follow any particular ordering.

- While calculating i_t we could either write it as a sum of $W_i x_t, V_i h_{t-1}, b_i$ or $b_i, V_i h_{t-1}, W_i x_t$. Thus we take the average over all the possible orderings.

$$L_{tanh}(y_k) = \frac{1}{M_N} \sum_{i=1}^{M_N} [tanh(\sum_{j=1}^{\pi_i^{-1}(k)} y_{\pi_i(j)}) - tanh(\sum_{j=1}^{\pi_i^{-1}(k)-1} y_{\pi_i(j)})]$$

Experiments

- The evaluation is done on the task of sentiment analysis.
- Three main results are evaluated for
  - Standard problem of word-level importance scores.
  - Word and phrase level importance in cases of compositionality.
  - Instances of positive and negative negation.
- Stanford Sentiment Treebank(SST) and Yelp Polarity(YP) datasets are used.

Baselines

- 4 state-of-the-art baselines are used:
  - Cell Decomposition(Murdoch & Szlam, 2017)
  - Integrated Gradients(Sundararajan et al., 2017)
  - Leave One Out(Li et al., 2016)
  - Gradient times input(gradient of the output probability with respect to the word embeddings is computed which is finally reported as a dot product with the word vector)

Unigram(Word) Scores

- CD scores for individual words are extracted from the LSTM are compared on similarity with the logistic regression coefficients.

Attribution Method	Stanford Sentiment	Yelp Polarity
Gradient	0.375	0.336
Leave one out (Li et al., 2016)	0.510	0.358
Cell decomposition (Murdoch & Szlam, 2017)	0.490	0.560
Integrated gradients (Sundararajan et al., 2017)	0.724	0.471
Contextual decomposition	0.758	0.520

Table 4: Correlation coefficients between logistic regression coefficients and extracted scores.

Identifying Dissenting Subphrases

- CD can correctly identify the sentiment for subphrases in a phrase(atmost 5 words) where the polarity differs.

Attribution Method	Heat Map									
Gradient	used	to	be	my	favorite	not	worth	the	time	
Leave One Out (Li et al., 2016)	used	to	be	my	favorite	not	worth	the	time	
Cell decomposition (Murdoch & Szlam, 2017)	used	to	be	my	favorite	not	worth	the	time	
Integrated gradients (Sundararajan et al., 2017)	used	to	be	my	favorite	not	worth	the	time	
Contextual decomposition	used	to	be	my	favorite	not	worth	the	time	

Legend: Very Negative | Negative | Neutral | Positive | Very Positive

Table 1: Heat maps for portion of yelp review with different attribution techniques. Only CD captures that ”favorite” is positive.

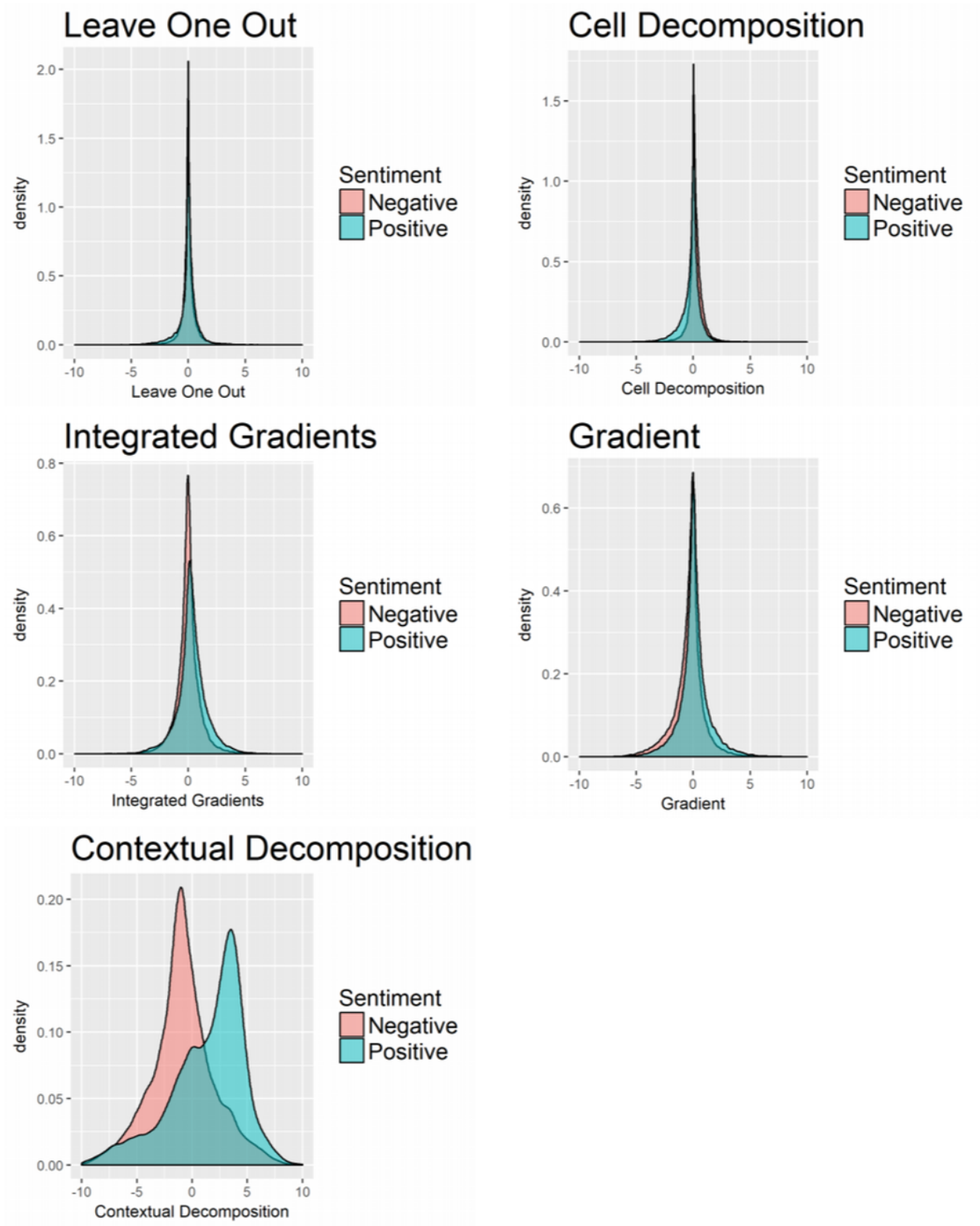


Figure 2: The distribution of attributions for positive (negative) sub-phrases contained within negative (positive) phrases of length at most five in the Yelp polarity dataset. The positive and negative distributions are nearly identical for all methods except CD, indicating an inability of prior methods to distinguish between positive and negative phrases when occurring in the context of a phrase of the opposite sentiment

High-Level Compositionality

- CD is also better at identifying cases where a sizeable portion of a sentence has an opposite polarity from the sentence.

Attribution Method	Heat Map
Gradient	It's easy to love Robin Tunney – she's pretty and she can act – but it gets harder and harder to understand her choices.
Leave one out (Li et al., 2016)	It's easy to love Robin Tunney – she's pretty and she can act – but it gets harder and harder to understand her choices.
Cell decomposition (Murdoch & Szlam, 2017)	It's easy to love Robin Tunney – she's pretty and she can act – but it gets harder and harder to understand her choices.
Integrated gradients (Sundararajan et al., 2017)	It's easy to love Robin Tunney – she's pretty and she can act – but it gets harder and harder to understand her choices.
Contextual decomposition	It's easy to love Robin Tunney – she's pretty and she can act – but it gets harder and harder to understand her choices.

Legend: Very Negative | Negative | Neutral | Positive | Very Positive

Table 2: Heat maps for portion of review from SST with different attribution techniques. Only CD captures that the first phrase is positive.

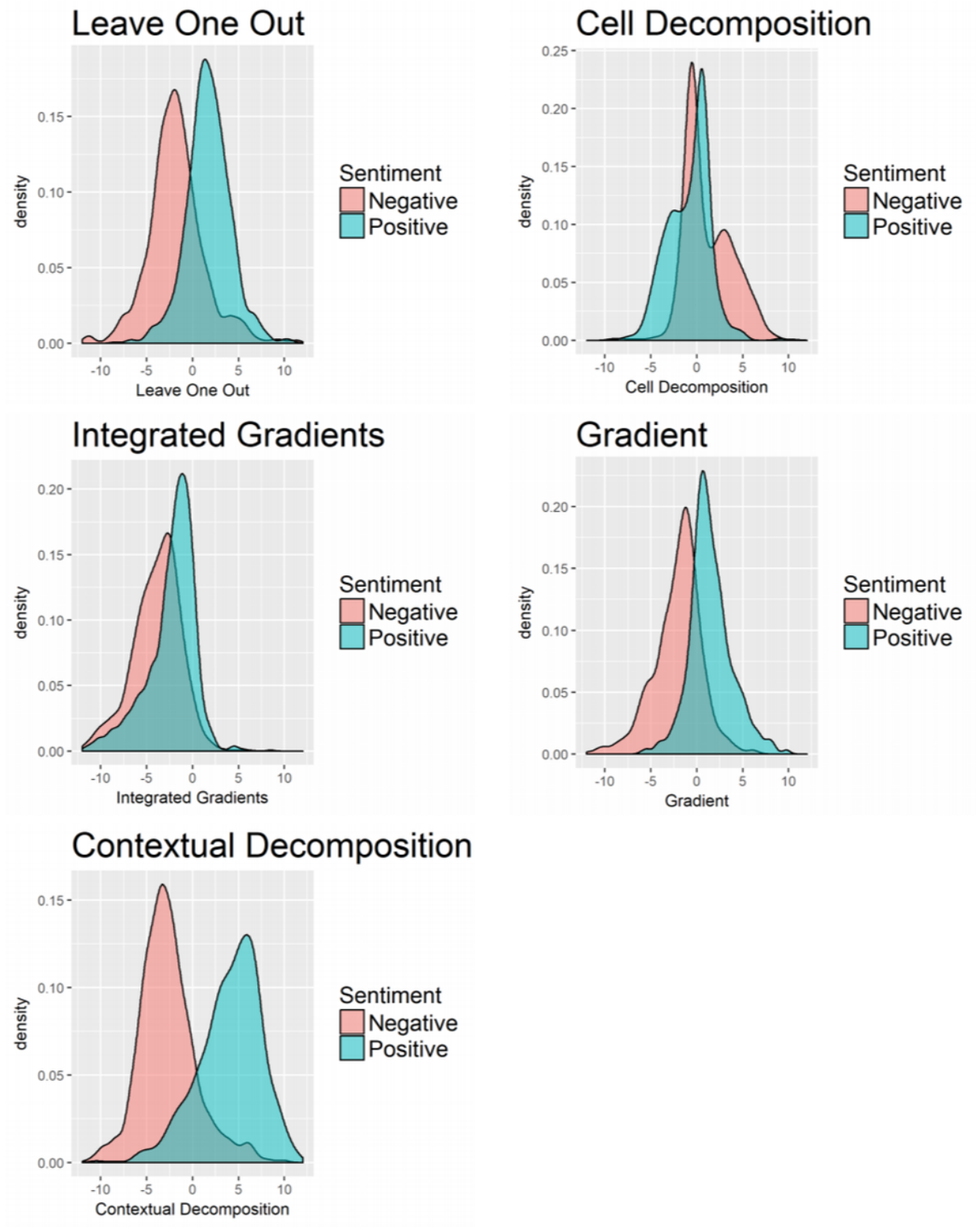


Figure 3: Distribution of positive and negative phrases, of length between one and two thirds of the full review, in SST. The positive and negative distributions are significantly more separate for CD than other methods, indicating that even at this coarse level of granularity, other methods still struggle.

Capturing Negation

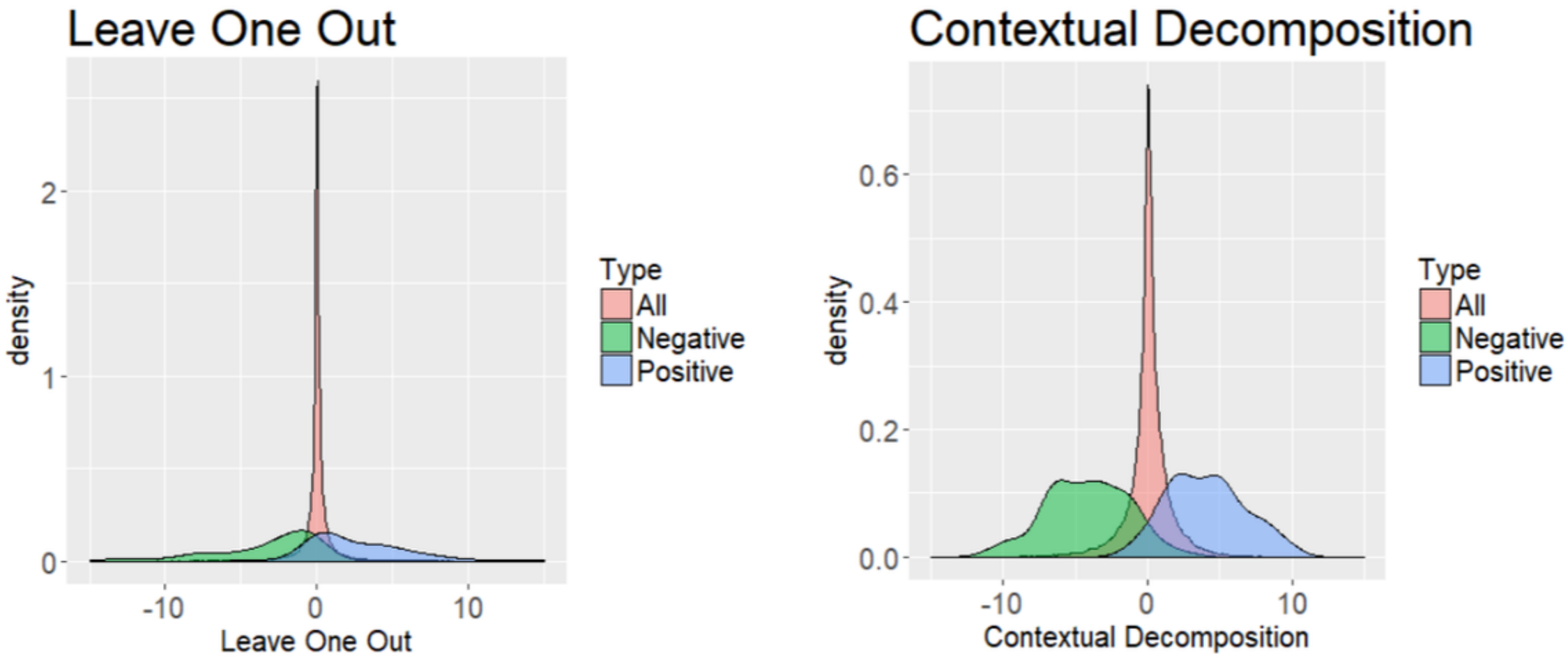


Figure 1: Distribution of scores for positive and negative negation coefficients relative to all interaction coefficients. Only leave one out and CD are capable of producing these interaction scores.

Similar Words

- A key aspect of the CD algorithm is that it helps us learn the value of a dense embedding vector (β_t) for a word or a phrase.

- For words and binary interactions, avg(β_t) is calculated across the data.

- Then, using similarity measures in the embedding space(eg. cosine similarity) we can easily find similar phrases/words

not entertaining	**not bad**	**very funny**	**entertaining**	**bad**
not funny	never dull	well-put-together piece	intelligent	dull
not engaging	n't drag	entertaining romp	engaging	drag
never satisfactory	never fails	very good	satisfying	awful
not well	without sham	surprisingly sweet	admirable	tired
not fit	without missing	very well-written	funny	dreary

Table 3: Nearest neighbours for selected unigrams and interactions using CD embeddings

Conclusion/Thoughts

- While the method itself is novel in that it moves past the traditional approach of looking just at word level importance scores; it only looks at one specific architecture which is applied to a simple problem.

- The authors don't talk about any future directions in the paper but a discussion about it happened at ICLR. Following are the importance points:

  - Look at interpreting a more complex model, for example, seq2seq. The author pointed out that he was affirmative that this model could be extended for such purposes although the computational complexity would increase since we would be predicting multiple outputs.

  - Look at whether this approach could be generalized to completely different architectures like CNN. As of now given a new model, we need to manually work out the math for the specific model. Could we develop some general approach towards this?

  - The author pointed out that they are working towards using this approach to interpret CNNs.

References

W. James Murdoch, Peter J. Liu, Bin Yu. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. ICLR 2018

Sebastian Bach, Alexander Binder, Gregoire Montavon, Frederick Klauschen, Klaus-Robert Muller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7):e0130140, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. ¨ Neural Computation, 9(8): 1735–1780, 1997.

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. arXiv preprint arXiv: 1506.02078, 2015.

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.

W James Murdoch and Arthur Szlam. Automatic rule extraction from long short-term memory networks. ICLR, 2017.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543, 2014.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685, 2017

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1631–1642, 2013.

Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. Visual analysis of hidden state dynamics in recurrent neural networks. arXiv preprint arXiv:1606.07461, 2016.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. CoRR, abs/1703.01365, 2017. URL http://arxiv.org/abs/1703.01365

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In Advances in neural information processing systems, pp. 649–657, 2015

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Thank-you for your attention.

Questions?