An Analysis of Logic Rule Dissemination in Sentiment Classifiers

Shashank Gupta, Mohamed Reda Bouadjenek, and Antonio Robles-Kelly

School of Information Technology, Deakin University, Waurn Ponds Campus, Geelong, VIC 3216, Australia

Abstract. Disseminating and incorporating logic rules in deep neural networks has been extensively explored for sentiment classification. Methods that are proposed for that goal rely on a component that aims to capture and model logic rules, followed by a sequence model to process the input sequence. While these methods claim to effectively capture syntactic structures that affect sentiment, they only show improvement in terms of accuracy to support their claims with no further analysis. Focusing on the *A-but-B* rule, we propose a new method to analyze and study the ability of these methods to identify the *A-but-B* structure, and to make their classification decision based on the *B* conjunct. Specifically, we rely on LIME, a model-agnostic framework that aims to explain the predictions of any classifier in an interpretable and faithful manner. Our experiments show that (a) accuracy is misleading in assessing these methods, (b) not all these methods are effectively capturing the *A-but-B* structure, (c) often, the underlying sequence model is what captures the syntactic structure, and (d) the best method classifies less than 12% of test examples based on the *B* conjunct.

Keywords: Sentiment Classification · Logic Rules · Explainable AI

1 Introduction

Methods of disseminating and incorporating logic rules in Deep Neural Networks have been extensively explored for sentiment classification. The two main methods developed for that purpose are: (i) Iterative Knowledge Distillation method [1] and (ii) the Contextualized Word Embeddings approach [2]. Briefly, these methods rely on a component aimed at capturing and modeling logic rules (e.g., the teacher network in the Iterative Distillation method and the ELMo model [3] in the Contextualized Word Embeddings approach), followed by a sequence model to process the input sequence, (e.g., a RNN).

The authors of these two methods claim that they effectively capture syntactic structures in the input sentence that affect its sentiment, but they have only used the improvement in terms of accuracy to support their claim with no further analysis. However, achieving a high classification accuracy does not necessarily mean that a method has effectively captured and encoded rules and other textual syntactic structures. For example, let's consider the sentence *"the*

casting was not bad but the movie was awful" that has an *A-but-B* structure – a component *A* followed by *but* which is then followed by a component *B*. In this example, the conjunction is interpreted as an argument for the second conjunct, with the first functioning concessively [4–6]. While a sentiment classifier can correctly identify that this sentence has a negative sentiment, it may fail to infer it's decision based *exclusively* on the *B* part of the sentence (i.e., *"the movie was awful"*), but instead, it may based it's decision on individual negative words also present in Part *A* (i.e., *"bad"*).

While focusing on the *A-but-B* syntactic structure and sentiment classification, we propose in this paper to study the ability of the aforementioned methods to: (i) effectively identifying the *A-but-B* structure in an input sentence, and to (ii) make their classification decision based on the *B* conjunct of a sentence. Specifically, we rely on a post-hoc explanation framework called *LIME* that explains the predictions of any classifier by providing feature attribution scores. We use *LIME* to estimate the impact of each conjunct in a sentence with an *A-but-B* structure on the decision made by a classifier. We validate our findings with an exhaustive experimental evaluation on the SST2 dataset [6] by testing various sentiment classifiers designed for logic rules dissemination. Among numerous findings, we show that: (a) accuracy is misleading in assessing methods for capturing logic rules, (b) not all methods are effectively capturing the *A-but-B* structure, (c) their sequence model is often what captures the syntactic structure, and (d) the best method bases its decision on the *B* conjunct in less than 25% of test examples.

2 Logic rules dissemination methods

In this section, we first describe the neural network architecture we use for sequence modeling, before discussing the main methods we analyse for logic rules dissemination in that architecture.

2.1 Network architecture

The backbone neural network [7, 8] we use throughout this paper is depicted in Figure 1. Three 1D CNN sequence layers (kernel size of 3, 4, and 5) process the word embeddings of an input sequence in parallel in order to extract diverse features and pass the concatenated features into a feed-forward binary classification layer with a sigmoid activation to extract the sentiment of the input sentence – 0 for a negative sentiment and 1 for a positive sentiment. In the next subsections, we will discuss the methods we analyze in this article that aim to incorporate and disseminate logic rules in the neural network architecture depicted in Figure 1.

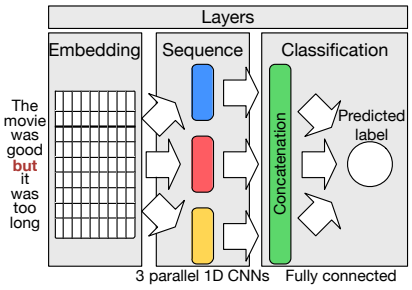


Fig. 1: Neural network.

2.2 Iterative Rule Knowledge Distillation

The Iterative rule knowledge distillation method proposed by Hu et al. [1] aims to transfer the domain knowledge encoded in first order logic rules into a neural network defined by a conditional probability $p_\theta(y|x)$ where θ is a parameter to learn. Specifically, during training, a posterior $q(y|x)$ is constructed by projecting $p_\theta(y|x)$ into a subspace constrained by the rules to encode the desirable properties, by using the following loss:

$$\begin{aligned} \min_{q,\xi\geq 0} \quad & KL(q(y|x)||p_\theta(y|x)) + C\sum_{x\in X}\xi_x \\ s.t. \quad & (1-\mathbb{E}_{y\leftarrow q(\bullet|x)}[r_\theta(x,y)]) \leq \xi_x \end{aligned}$$

where $q(y|x)$ denotes the distribution of (x, y) when x is drawn uniformly from the train set X and y is drawn according to $q(\bullet|x)$, and $r_\theta(x, y) \in [0, 1]$ is a variable that indicates how well labeling x with y satisfies the rule. The closed form solution for $q(y|x)$ is used as soft targets to imitate the outputs of a rule-regularized projection of $p_\theta(y|x)$, which explicitly includes rule knowledge as regularization terms.

Next, the rule knowledge is transferred to the posterior $p_\theta(y|x)$ through knowledge distillation optimization objective:

$$(1-\pi)\times\mathcal{L}(p_\theta, P_{true}) + \pi\times\mathcal{L}(p_\theta, q)$$

where P_{true} denotes the distribution implied by the ground truth, $\mathcal{L}(\bullet,\bullet)$ denotes the cross-entropy function, and π is a hyperparameter that needs to be tuned to calibrate the relative importance of the two objectives. Overall, the Iterative rule knowledge distillation method is agnostic to the network architecture, and thus is applicable to general types of neural models such as the one depicted in Figure 1.

2.3 Contextual Word Embeddings

Traditional word embeddings methods like Word2Vec [9] and Glove [10] do not capture the local context of the word in a sentence. However, language is complex and context can completely change the meaning of a word in a sentence. Hence, contextual word embeddings methods have emerged as a way to capture the different nuances of the meaning of words given the surrounding text. Krishna et al. [2] have advocated that contextualized word embeddings might capture logic rules and thus disseminate that latent information in the 1D CNN sequence models of the neural network in Figure 1. In the following, we briefly review two of the main context word embedding methods we use in our experiments.

ELMo: stands for Embeddings from Language Models is a pre-trained model developed by Peters et al. [3]. Instead of using a fixed embedding for each word, ELMo looks at the entire sentence before assigning each word in it an embedding.

It uses a bi-directional LSTM trained on a specific task to be able to create those embeddings. Krishna et al. [2] proposed to use ELMo in their method.

BERT: stands for Bidirectional Encoder Representations from transformers.This is also a pre-trained model developed by Devlin et al. [11]. Briefly, the BERT is model based on Encoder Transformer blocks [12], which processes each element of the input sequence by incorporating and estimating the influence of other elements in the sequence to create embeddings.

To further test the hypothesis proposed by Krishna et al. [2], we conduct experiments with two different context-free word embeddings namely Word2vec developed by Mikolov et al. [9] and Glove developed by Pennington et al. [10] in which each token is mapped to a unique vector independent of its context. These word embeddings are used as an ablation study to analyze the effectiveness of the rule knowledge distillation method discussed in the previous section.

3 Methodology

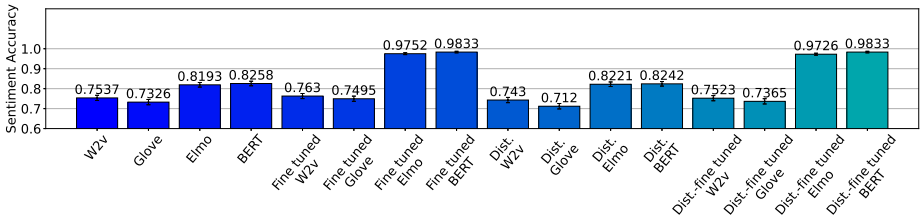
As mentioned earlier, our main goal in this paper is to assess each sentiment classifier for it's ability to correctly classify a test example with an *A-but-B* structure only on the basis of the B conjunct. Specifically, given a sentence S which is an ordered sequence of terms $[t_1 t_2 \cdots t_n]$, *LIME* assigns a weight w_n to each term t_n in S where a positive weight indicates that t_n contributes and supports the positive class, and a negative weight indicates how much t_n supports the negative class. In order to estimate how much a term t_n contributes to the final decision of the classifier, we propose to normalize its weight as follows:

$$\tilde{w_n} = \begin{cases} w_n \times P(y=1|S), & \text{if } w_n \geq 0 \\ |w_n| \times P(y=0|S), & \text{otherwise} \end{cases} \tag{1}$$

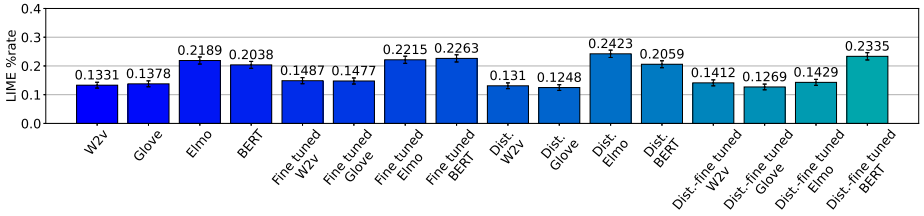
where $P(y=c|S)$ is the probability to predict class c given sentence S. Hence, every sentence in our test set is mapped to a vector $[\tilde{w}_1 \tilde{w}_2 \cdots \tilde{w}_n]$ with $\tilde{w}_n$ indicating how much the word t_n contributed to the final decision of the classifier. Next, given a sentence that contains an *A-but-B* structure, we define the normalized weights $\tilde{W}(A) = [\tilde{w}_0 \cdots \tilde{w}_{i-1}]$ and $\tilde{W}(B) = [\tilde{w}_{i+1} \cdots \tilde{w}_n]$ as respectively the left and right sub-sequences w.r.t the word *"but"* indexed by i. Finally, we compute an expectation over weights as follows: $\mathbb{E}_A(W) = \sum_{\tilde{w}_k \in \tilde{W}(A)} \tilde{w}_k$ and $\mathbb{E}_B(W) = \sum_{\tilde{w}_k \in \tilde{W}(B)} \tilde{w}_k$, and we propose to conclude that a classifier has based its classification prediction by relying on the B conjunct if: $\mathbb{E}_B(W) > \mathbb{E}_A(W)$ ***and*** p-value ≤ 0.05 – this condition aims to make sure that the observed difference is statistically significant.

4 Experimental evaluation

In this section, we fist describe the dataset we have used in our evaluation before discussing the obtained results.



(a) Sentiment Accuracy of Classifiers with 95% confidence interval.



(b) Proportion of test examples that have been *correctly* classified based on the B conjunct according to the *LIME* framework with 95% confidence interval.

Fig. 2: Results obtained on SST2 dataset.

4.1 Dataset

Our experiments (as well as those presented in Hu et al. [1] and Krishna et al. [2]) are based on the Stanford Sentiment Treebank (SST2) dataset [6], which is a binary sentiment classification dataset. The dataset consists of 9,613 single sentences extracted from movie reviews, where sentences are labelled as either positive or negative each accounting for about 51.6% and 48.3%. A total of 1,078 sentences contain the *A-but-B* syntactic structure which accounts for about 11.2% of the dataset. We report our results only on test examples that contain an *A-but-B* syntactic structure to demonstrate the ability of a classifier to capture *A-but-B* pattern. Hence, all classifier are trained, tuned, and tested using stratified nested k-fold cross-validation and evaluated primarily according to accuracy. These sentences are identified simply by searching for the word "but" as proposed in [1, 2, 6].

4.2 Performance evaluation

In this section, we discuss the results of our analysis of logic rules dissemination methods in sentiment classifiers. The configuration options that were considered are the following: {Word2vec, Glove, ELMo, BERT} $\times$ {Static, Fine-tuning} $\times$ {no distillation, distillation}, which gives a total of 16 classifier analysed on sentences with an *A-but-B* structure. To summarize all the results obtained over all the above configurations, Figures 2a and 2b show the accuracy and the

ability of the methods to base their classification decisions on the B conjunct. From these results, we make the following observations:

Accuracy analysis: In Figure 2a, we observe that the distillation model described in Hu et al. [1] is ineffective as it gives almost no improvement in terms of accuracy as also noted in in [2]. Second, we note that fine-tuning all embeddings provides a statistically significant improvement of accuracy for almost all methods Finally, it is clear that the best method is BERT, followed by ELMo, followed by either Glove or Word2vec.

Rule dissemination analysis: In Figure 2b we show the proportion of test examples that have been *correctly* classified based on the B conjunct using our method described in Section 3. Briefly, we first observe that for all methods, less than 25% of the test examples are effectively classified based on the B conjunct, which shows that the intend of these methods as described by their authors in [1, 2] is far from being achieved. This suggests that there is still a lot of research to be done on this NLP topic. Second, we again note that there is almost no improvement between for instance Word2vec with and without distillation (Figures 2a and 2b), which simply suggests that in [1] it is the 1D CNN sequence models that are capturing to some extend the *A-but-B* structure. Finally, we note that some models although having higher sentiment accuracy performs poorly on rule dissemination performance and vice-versa. For example, Dist. Elmo and Dist. BERT have similar sentiment accuracy in figure 2a but Dist. Elmo outperforms Dist. BERT by a statistically significant margin on rule dissemination performance in figure 2b. Similar phenomenon can be observed for Dist.-fine tuned Elmo and BERT models where later outperforms former even though having similar sentiment accuracy. This indicates that accuracy is misleading and there is no correlation between sentiment accuracy and actual rule dissemination performance.

5 Conclusion

This paper gives an analysis and a study of logic rules dissemination methods on their ability to identify *A-but-B* structures while making their classification decision based on the B conjunct. We proposed an assessment method based on the *LIME* framework for that goals. Our experimental evaluation shows that (a) accuracy is misleading to assess whether the classifier based its decision as per B conjunct (b) not all methods are effectively capturing *A-but-B* structure, (c) that their underlying sequence model is often the one that captures to some extent the syntactic structure, and (d) that for the best method less than 25% of test examples are effectively classified based on the B conjunct, indicating that a lot of research needs to be done in this topic. Limitations of this analysis include that it relies on *LIME* which has a few drawbacks such as it provides non robust explanations and that it suffers from label and data shift. Future work includes exploring more robust explanation methods such as Grad-CAM [13].

References

1. Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany, August 2016. Association for Computational Linguistics.
2. Kalpesh Krishna, Preethi Jyothi, and Mohit Iyyer. Revisiting the importance of encoding logic rules in sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4743–4751, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
3. Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.
4. Robin Lakoff. If's, and's and but's about conjunction. In Charles J. Fillmore and D. Terence Langndoen, editors, *Studies in Linguistic Semantics*, pages 3–114. Irvington, 1971.
5. Diane Blakemore. Denial and contrast: A relevance theoretic analysis of "but". *Linguistics and Philosophy*, 12(1):15–37, 1989.
6. 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*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.
7. Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.
8. Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 253–263, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.
9. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
10. Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
11. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
12. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan,

and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
13. Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.