An Analysis of Logic Rule Dissemination in Sentiment Classifiers

Anonymous Author(s)

ABSTRACT

Disseminating and incorporating logic rules in deep neural networks has been extensively explored for various natural language processing tasks, in particular for sentiment classification. Iterative rule knowledge distillation and the contextualized word embeddings are the two main methods that have been proposed for that purpose. Both methods rely on a component supposed to capture and model logic rules, followed by a sequence model to process the input sequence, i.e., 1D CNN or RNN. While these two methods claim that they effectively capture syntactic structures that affect sentiment, they only provide improvement in terms of accuracy to support their claims with no further analysis. Focusing on the *A-but-B* syntactic structure, 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 experimental evaluation on the SST2 dataset shows that (a) accuracy is misleading in assessing methods for capturing logic rules, (b) that not all these methods are effectively capturing the *A-but-B* structure, (c) that for some methods the underlying sequence model is the one that captures to some extent the syntactic structure, and (d) that for the best method less than 12% of test examples are classified based on the *B* conjunct, indicating that a lot of research needs to be done in this topic.

Keywords: Sentiment Classification, Logic Rules, Explainable AI.

1 INTRODUCTION

Methods for disseminating and incorporating logic rules and other textual syntactic structures in deep neural networks have been extensively explored for various natural language processing tasks including, question answering [1], machine translation [2], and sentiment classification [3]. The ultimate goal is to model and transfer various human interpretable textual, logic and syntactic rules to a neural network in order to improve its effectiveness and accuracy.

The two main methods that have been recently proposed for encoding logic rules in a neural network for sentiment classification are: (i) the iterative rule knowledge distillation method [3] and (ii) the contextualized word embeddings approach [4]. Briefly, the two 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 [5] in the contextualized word embeddings approach), followed by a sequence model to process the input sequence, (e.g., a 1D CNN or a RNN).

It is worth noting that, while the authors of these two methods claim that they effectively capture syntactic structures in the sentence that affect its sentiment, they have only used the improvement in terms of accuracy to support their claims with no further analysis. However, achieving a high classification accuracy does not necessarily mean that a method has effectively captured and encoded rules and text syntactic structures. For example, let's consider the sentence *"the casting was terrific but the movie was horrible"* that has an *A-but-B* structure – a component *A* being followed by *but* which is followed by a component *B*. In this example, the conjunction is interpreted as an argument for the second conjunct, with the first functioning concessively [6–8]. While a sentiment classifier can correctly identify that this sentence has a negative sentiment, it may fail to infer it's decision based only on the *B* part of the sentence (i.e., *"the movie was horrible"*), but instead, it may based it's decision on individual negative words also present in Part *A* (i.e., *"terrific"*). Thus, we argue in this paper that the accuracy of a classifier does not necessarily indicate that it has effectively captured textual structures.

While focusing on the *A-but-B* syntactic structure and sentiment classification, we propose in this paper to analyze and 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 an AI explainability approach using the *LIME* framework that aims to explain the predictions of any classifier in an interpretable and faithful manner. Briefly, 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 perform an exhaustive experimental evaluation on the SST2 dataset by testing various methods for encoding and disseminating logic rules in sentiment classifiers. Among numerous findings, we show that (a) accuracy is misleading in assessing methods for capturing logic rules, (b) that not all methods are effectively capturing the *A-but-B* structure, (c) that their sequence model is often the one that captures to some extent the syntactic structure, and (d) that the best method bases its decision on the *B* conjunct in less than 12% of test examples, indicating that a lot of research needs to be done here.

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 [9, 10] we use throughout this paper is depicted in Figure 1. Briefly, the network takes as input a sequence as token indexes, which are first processed by an embedding layer to be converted into dense vectors of fixed size. Next, three 1D CNN sequence models (kernel size of 3, 4, and 5) process the embeddings in parallel in order to extract diverse features and representations from the input sequence. These 1D CNN sequence models may learn various internal properties of the sequence that are useful for sentiment classification. Finally, the outputs of the three 1D CNNs are concatenated before being fed 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.

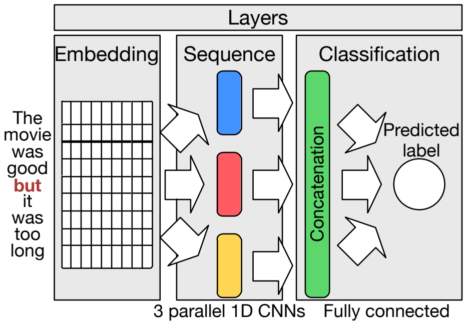


Figure 1: Neural network used in the experiments.

2.2 Iterative Rule Knowledge Distillation

The Iterative rule knowledge distillation method proposed by Hu et al. [3] 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. To integrate the information encoded in the rules, Hu et al. [3] have proposed to train the network via knowledge distillation as proposed in Hinton et al. [11] where hard targets are provided through labelled training data and soft targets are constructed through rule constrained projection of posterior $p_\theta(y|x)$ as proposed in Posterior Regularization [12].

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:

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

where P_{true} denotes the distribution implied by the ground truth, $\ell(\bullet,\bullet)$ denotes the cross-entropy function, and π is a hyperparameter that needs to be tuned to calibrate the relative importance of the two objectives. Following the terminology in [11], p_θ is called a "student" network and q is called a "teacher" network, which is intuitively analogous to human education where a teacher is aware of systematic general rules and instructs students. 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 Word Embeddings

It has been hypothesised that the 1D CNN sequence models in the neural network of Figure 1 may not be able to learn interesting relationships from the input tokens, among which the *A-but-B* structure [4]. To verify this hypothesis, we propose to use two different context-free word embeddings in which each token is mapped to a unique vector independent of its context. This approach can also serve as an ablation study to analyze the effectiveness of the rule knowledge distillation method discussed in the previous section. We employ the following embeddings:

Word2vec: which is one of the most popular methods to efficiently create word embeddings developed by Mikolov et al. [13]. In brief, Word2vec embeddings are computed from a two-layer neural network. Word2vec maps each token to a vector space, typically of several hundred dimensions, where word vectors are positioned in the vector space such that words that share common contexts (semantically similar) are located close to each other in the space.

Glove: is an unsupervised learning algorithm for obtaining vector representations for words developed by Pennington et al. [14]. Training is performed on the non-zero entries of a global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. A matrix factorization algorithm is applied to efficiently extract the embeddings.

2.4 Contextual Word Embeddings

The two word embedding methods described above provide a unique and fixed vector for each word in the vocabulary. 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. [4] 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. [5]. 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. [4] 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. [15]. Briefly, the BERT is model based on Encoder Transformer blocks [16], which processes each element of the input sequence by incorporating and estimating the influence of other elements in the sequence to create embeddings.

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. Note that the neural network model depicted in Figure 1 can be instantiated using various configuration options described in the previous section. For example, we can use static Glove with the rule knowledge distillation method, fine-tuning ELMo to capture logic rules, or simply use static Word2vec to rely on the 1D CNN sequence models to capture *A-but-B* structures.

We will seek to analyse all possible configuration options in Section 4. To do this, we rely on LIME [17], which is a model-agnostic framework that aims to explain the predictions of any classifier in an interpretable and faithful manner. *LIME* relies on local surrogate models to explain individual predictions of black box machine learning models. Surrogate models are trained to approximate the predictions of the underlying black box model in order to explain individual predictions. *LIME* has been used to explain outputs of various machine learning models ranging from a simple logistic regression to complex deep neural network like Inception network [17]. The output of *LIME* is a list that reflects the contribution of each feature to the prediction of a test sample. This provides local interpretability, and it also allows to determine which feature changes will have most impact on the prediction.

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 contexts $C(A) = [\tilde{w}_{i-k} \cdots \tilde{w}_{i-1}]$ and $C(B) = [\tilde{w}_{i+1} \cdots \tilde{w}_{i+k}]$ as respectively the left and a right sub-sequences w.r.t the word *"but"* indexed by i and given a window size k (in our experiments we set $k = 10$). Finally, we propose to conclude that a classifier has based its classification prediction by relying on the B conjunct if: $\mu_{C(B)} > \mu_{C(A)}$ ***and*** p-value ≤ 0.05, where μ is the mean – the second condition aims to make sure that the 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.

4.1 Dataset

Our experiments (as well as those presented in Hu et al. [3] and Krishna et al. [4]) are based on the Stanford Sentiment Treebank (SST2) dataset [8], 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 [3, 4, 8].

4.2 Performance evaluation

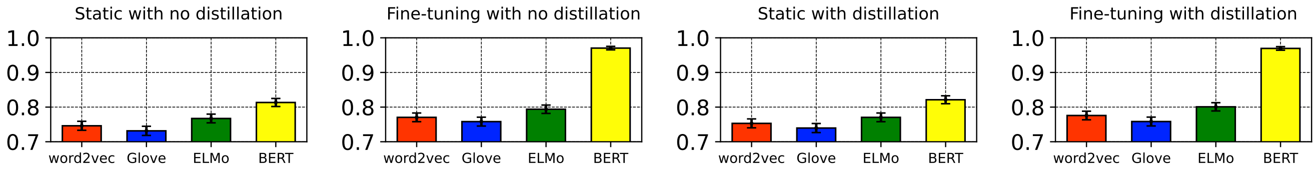
In this section, we discuss the results of our analysis of logic rules dissemination methods in sentiment classifiers. The configuration options that were considered are the following: {Word2vec, Glove, ELMo, BERT} × {Static, Fine-tuning} × {no distillation, distillation}, which gives a total of 16 classifier analysed on sentences with an *A-but-B* structure. To summarize all the results obtained over all the above configurations, Figures 2 and 3 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: First, we observe that the distillation model described in Hu et al. [4] is ineffective as it gives almost no improvement in terms of accuracy as also noted in in [4]. Second, we note that fine-tuning all embeddings provides a statistically significant improvement of accuracy for almost all methods as shown in Figure 2e. Finally, it is clear that the best method is BERT, followed by ELMo, followed by either Glove or Word2vec.

Rule dissemination analysis: In Figures 3a- 3d 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 12% 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 [3, 4] is far from being achieved. This suggests that there is still a lot of research to be done on this NLP topic. Second, BERT provides the best performance which indicates that probably the Multi-head self-attention mechanism is better at capturing logic rules. Third, we again note that there is almost no improvement between for instance Word2vec with and without distillation (Figures 3a and 3c), which simply suggests that in [3] it is the 1D CNN sequence models that are capturing to some extend the *A-but-B* structure. Finally, we note that while the ELMo method of [4] achieves a statistically significant improvement in terms of accuracy w.r.t Word2vec and Glove (Figure 2), the performance obtained for the rule dissemination analysis is disappointing as it doesn't outperform Word2vec or Glove. Moreover, we observe that in a few cases, Word2vec provides even a statistically significant improvement. This indicates that accuracy is misleading and that the ELMo-based method described in [4] is also ineffective in capturing logic rules, but it instead bases its decision based on individual words in a sentence.

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) not all methods are effectively capturing *A-but-B* structure, (b) that their underlying sequence model is often the one that captures to some extent the syntactic structure, and (c) that for the best method less than 12% 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 [18].

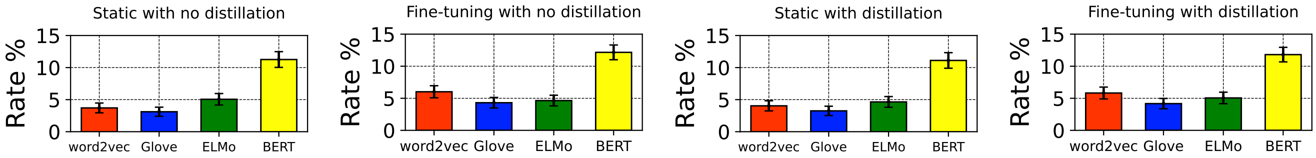


(a) (b) (c) (d)

			No distillation								With distillation							
			Static				Fine-tuned				Static				Fine-tuned			
			word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT
No distillation	Static	word2vec		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
		Glove	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
		ELMo	✓	✓		✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
		BERT	✓	✓	✓		✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗
	Fine-tuned	word2vec	✓	✓	✗	✗		✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
		Glove	✗	✓	✗	✗	✗		✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
		ELMo	✓	✓	✓	✗	✓	✓		✗	✓	✓	✓	✗	✓	✓	✗	✗
		BERT	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✗
With distillation	Static	word2vec	✗	✓	✗	✗	✗	✗	✗	✗		✗	✗	✗	✗	✗	✗	✗
		Glove	✗	✗	✗	✗	✗	✗	✗	✗	✗		✗	✗	✗	✗	✗	✗
		ELMo	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓		✗	✗	✗	✗	✗
		BERT	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓		✓	✓	✓	✗
	Fine-tuned	word2vec	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗		✗	✗	✗
		Glove	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗		✗	✗
		ELMo	✓	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✗	✓	✓		✗
		BERT	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	

(e)

Figure 2: (a)-(d) show the performance of the classifiers using accuracy with 95% confidence interval. In Figure (e), each entry (i, j) indicates if the method i outperforms the method j *and* if the improvement is statistically significant – statistically significance implies p-value$\leq$ 0.05. Figure (e) is not symmetric as the i outperforms j relationship is not a symmetric relation.



(a) (b) (c) (d)

			No distillation								With distillation							
			Static				Fine-tuned				Static				Fine-tuned			
			word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT	word2vec	Glove	ELMo	BERT
No distillation	Static	word2vec		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
		Glove	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
		ELMo	✓	✓		✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
		BERT	✓	✓	✓		✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗
	Fine-tuned	word2vec	✓	✓	✗	✗		✓	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗
		Glove	✗	✓	✗	✗	✗		✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
		ELMo	✗	✓	✗	✗	✗	✗		✗	✗	✓	✗	✗	✗	✗	✗	✗
		BERT	✓	✓	✓	✗	✓	✓	✓		✓	✓	✓	✗	✓	✓	✓	✗
With distillation	Static	word2vec	✗	✗	✗	✗	✗	✗	✗	✗		✗	✗	✗	✗	✗	✗	✗
		Glove	✗	✗	✗	✗	✗	✗	✗	✗	✗		✗	✗	✗	✗	✗	✗
		ELMo	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓		✗	✗	✗	✗	✗
		BERT	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓		✓	✓	✓	✗
	Fine-tuned	word2vec	✓	✓	✗	✗	✗	✓	✗	✗	✓	✓	✗	✗		✓	✗	✗
		Glove	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗		✗	✗
		ELMo	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗		✗
		BERT	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓	

(e)

Figure 3: (a)-(d) show the proportion of test examples that have been *correctly* classified based on the B conjunct according to the *LIME* framework with 95% confidence interval. In Figure (e), each entry (i, j) indicates if the method i outperforms the method j *and* if the improvement is statistically significant – statistically significance implies p-value$\leq$ 0.05. Figure (e) is not symmetric as the i outperforms j relationship is not a symmetric relation.

REFERENCES

[1] Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander Smola, and Le Song. Variational reasoning for question answering with knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[2] Edoardo Maria Ponti, Roi Reichart, Anna Korhonen, and Ivan Vulić. Isomorphic transfer of syntactic structures in cross-lingual NLP. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1531–1542, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Diane Blakemore. Denial and contrast: A relevance theoretic analysis of "but". *Linguistics and Philosophy*, 12(1):15–37, 1989.

[8] 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.

[9] 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.

[10] 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.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[12] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(67):2001–2049, 2010.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[18] 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.