Layer-wise Analysis of Bert Model for Sentiment Analysis

Yamei Tu, Jason Yao, Mohammad Samavatian

The Ohio State University
{tu.253,yao.712,samavatian.1}@osu.edu

1 Introduction

Recently, there are many work focus on pre-trained latent, compositional vector representations of text data[10][3][8], which can be utilized in lots of downstream tasks like as classification and clustering. The vector can capture the complex word and phrase interactions, the challenge is how to handle compositional semantics to reach predictions.

There are exists two different types of explanations trying to open the black box and link the compositional semantics to final predictions. The first type is to construct the model with intrinsically interpretable structures, for example, in [2] they proposed to automatically search for position in source sentence that are related to final predictions, instead of using a fixed-length vector. The other type is post-hoc explanation which targets to explain the results not restricted to the model. The typical work flow for post-hoc explanation is try to assign importance score to the individual input. However, the problem is that they may not work for compositional semantics, since the the role of phrase is not just the sum of words.

Contextual decomposition(CD) [9] goes beyond the additive assumption and compute the contribution solely made by a word/phrase to the model prediction, by decomposing the output variables of the neural network at each layer. Using the contribution scores so derived, these algorithms generate hierarchical explanation on how the model captures phrase interactions in making predictions. Extensions of CD presented in order to apply them to larger and deeper models like GCD [6] and ACD [11] or different network architectures [4]. Former works on contextual decomposition have not studied phrase interactions in a formal context. As a result, this line of works focused on exploring model-specific decompositions based on their empirical performance. In contrast, another work revisited the definition of phrase interactions with a formal perspective [5]. In [5] a formal way presented to quantify context independent importance of each individual word/phrase. They proposed two explanation algorithms according to their formulation, namely the Sampling and Contextual Decomposition algorithm (SCD), which overcomes the weakness of contextual decomposition algorithms, and the Sampling and OCclusion algorithm (SOC), which is simple, model-agnostic,and performs competitively against prior lines of algorithms. All mentioned methods use method and algorithms to interpretation of the model based on the final output of the model. A very important aspect of the interpretation and explanation techniques is to evaluating the model and aid the scientist to tune their models and hyper-parameters. In this work we go one step deeper the use the last two methods (SCD and SOC) to visualizes and explain intermediate values in multi-layer networks and show the evolution of their prediction and their discovered phrase interaction. This can help us to see how multiple layers contribute in the final prediction and also how they assist in more accurate understanding of natural language processing. We designed a framework to capture the contribution of each encoder layer in the Bert transformer devlin2018bert. We studied the effect of adding encoder layers and excluding encoder layers to the ability of the model to find phrase interaction. We showed that not all layers are effective equally and their contribution to phrase interactions is different; prior layers focus on whole input and later layers focus on smaller phrases of the input. The rest of this paper is organized as follows: Section 2 provides background and a brief overview of related works information. Section 3 details the framework and implementation of our method. Section 4 presents the experimental setup and final results and Section 5 concludes.

2 Background and related works

2.1 Vector Representation Analysis

There exists lots of methods to analysis what kinds of information is captured in the vector representation of text in NLP area. We divide it into two types: hypothesis-driven and data-driven. Hypothesis-driven methods include *probes* or *diagnostic classifiers*, which propose whether specific inputs can be decoded from the hidden state, and run experiments to test the hypothesis. It requires much training, work and computation for one hypothesis once a time.

Another type is data-driven, including gradient-based and contextual decomposition. [1]. The main idea it to define the relevance vector at the output player in the forward pass, keeping the interesting dimensions as non-zero. Then propagate the relevance backwards through the network. Another example is Contextual Decomposition(CD)[9]. The key point is to decompose the model outputs into two sets of contributions. CD decomposes the output of each layer **h** into two terms, β and γ where β represents the contribution from the given phrase **p** and γ denotes the contribution involving other factors. The input is decomposed as:

$$\beta = x_t, \gamma = 0(word \in p) \tag{1}$$

$$\beta = 0, \gamma = x_t(word \notin p) \tag{2}$$

In the linear layer **h' = Wh+b**, given the input **h** decomposed as $\beta + \gamma$, the output of **h'** is decomposed as β**' = W**β**,** γ**' = W**γ **+b**. CD decomposed all the layers including the final prediction s(x) = $\beta + \gamma$. If the function is non-linear, we calculate the contribution as the average activation differences caused by β with γ or not :

$$\beta' = \frac{1}{2}[\sigma(\beta + \gamma) - \sigma(\gamma)] + \frac{1}{2}[\sigma(\beta) - \sigma(0)] \tag{3}$$

2.2 N-Context Independent Importance

Given a phrase $p := x_{i:j}$ appearing in a specific input $x_{1:T}$, the N-context independent importance is defined as the output difference after masking out the phrase p, marginalized over all the possible N-word contexts, denoted as $\hat{x}_\sigma$, around p in the input x. For instance, to evaluate the context independent importance up to one word of *very* in the sentence *The film is very interesting* in a sentiment analysis model, we sample possible adjacent words before and after the word *very*, and average the prediction differences after some practice of masking the word *very*. Context independent importance is formally written as,

$$\phi(p,x) = E_{\hat{x}_\sigma}[s(x_{-\sigma};\hat{x}_\sigma) - s(x_{-\sigma} \setminus p;\hat{x}_\sigma)] \tag{4}$$

- $x_{-\sigma}$: resulting sequence after masking out an N-word context surrounding the phrase p from the input x.
- $\hat{x}_\sigma$: a N-word sequence sampled from a distribution $p(\hat{x}_\sigma | x_{-\sigma})$, which is conditioned on the phrase p as well as other words in the sentence x.
- $s(x_{-\sigma};\hat{x}_\sigma)$: denotes the model prediction score after replacing the original context words $x_{-\sigma}$ with a sampled N-word context $\hat{x}_\sigma$.
- $x \setminus p$: denotes the operation of masking out the phrase p from the input sentence x.

2.3 Sampling and Contextual Decomposition Algorithm(SCD)

The formulation in CD causes β' terms to be context dependent because the calculation relies on γ terms that involves information about context words of the phrase p. To eliminate the dependence, in [5] activation decomposition step is modified the in CD and propose Sampling and Contextual Decomposition algorithm (SCD), where it defines β' as the expected activation difference caused by β for possible γ associated with β.

$$\beta' = E_\gamma[\sigma(\beta+\gamma) - \sigma(\gamma)] = E_h[\sigma(h) - \sigma(h-\beta)] \tag{5}$$

By taking an expectation over γ, the dependencies eliminated. The decomposition defined above is a layer-wise application of the N-context independent importance in equation 4 , where the masking operation x\p is implemented as calculating $h-\beta$ following the line of CD algorithms. E_γ is the expectation over γ, to eliminate the dependence. E_γ is estimated in the following steps: (1)sampling a set of N-word contexts with respect to phrase p and (2) replacing the specific N-word context of p with the sampled context (3) feed each of them into the classifier model (4) record the inputs of each activation functions for each input (5) the decomposition of i-th non-linear activation function is calculated as:

$$\beta' = \frac{1}{|S_h^{(i)}|} \sum_{h \in S_h^{(i)}} [\sigma(h) - \sigma(h-\beta)] \tag{6}$$

$S_h^{(i)}$ denotes the obtained sample set of i-th non-linear activation function.

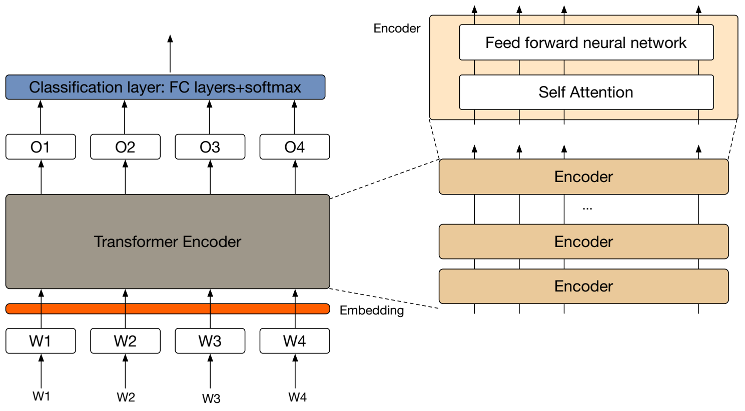

Figure 1: Bert model architecture.

2.4 Sampling and Occlusion Algorithm(SOC)

Input occlusion algorithm in [7] calculate the importance of phrase **p** specific to an input example **x** by observing the prediction difference after replacing the phrase **p** with padding tokens, noted as $\mathbf{0}_{\mathbf{p}}$,

$$\phi(\mathbf{p},\mathbf{x}) = s(\mathbf{x}) - s(\mathbf{x}_{-\mathbf{p}};\mathbf{0}_{\mathbf{p}}) \tag{7}$$

The importance score $\mathbf{0}_{\mathbf{p}}$ is dependent on all the neighbor words of **p** in **x**, replacing the N-context words of phrase **p** with words in sampled one eliminates the dependence and leads to the SOC algorithm.

Formally, we calculate context independent difference $\phi(\mathbf{p},\mathbf{x})$ as following steps: (1) sampling a set of N-word contexts of the give phrase **p** (2) replace the N neighbor words of **p** in **x** with sampled text $\hat{\mathbf{x}}_\delta$ (3) replacing the phrase **p** with padding tokens (4) Let S note for the set of samples, for each $\hat{\mathbf{x}}_\delta$ in S, compute model prediction differences (5)The importance $\phi(\mathbf{p},\mathbf{x})$ is then calculated as the average prediction differences

$$\phi(\mathbf{p},\mathbf{x}) = \frac{1}{|S|}\sum_{x_\delta \in S}[s(\mathbf{x}_{-\delta};\mathbf{x}_\delta) - s(\mathbf{x}_{-\{\delta,\mathbf{p}\}};\mathbf{x}_\delta;\mathbf{0}_{\mathbf{p}})] \tag{8}$$

2.5 Bert Model

Bidirectional Encoder Representations from Transformers(BERT) is state-of-the-art language model to translate a sentence to high-dimensional embedding, further used in multiple downstream tasks, such as question answering, sentiment analysis.

The key component of BERT is the transformer encoder, an attention-based mechanism that learns the contextual information between words. The encoding is a stack of encoders, there are two variants of Bert model. Large includes 24 layers of encoders, in this project, we utilize the base version, containing 12 layers of encoders. The encoders have the same structure, each one is consists of two sub-layers. First, the input goes through the self-attention layer, which can capturing the pair wise word interactions between words. Then, the outputs of self-attention layer are fed into a feed-forward neural network. The structure can be illustrated in Figure 1

2.6 Agglomerative clustering

Agglomerative clustering introduced by [11] produces a hierarchical clustering of the input features, along with the contribution of each cluster to the final prediction. At a high-level, agglomerative hierarchical clustering using CD or OC interaction as the joining metric to determine which clusters to join at each step. This procedure builds the hierarchy by starting with individual features and iteratively combining them based on the interaction scores provided by CD/OC. After initializing by computing the scores of each feature individually, the algorithm iteratively selects all groups of features within k% of the highest-scoring group and adds them to the hierarchy. Each time a new group is added to the hierarchy, a corresponding set of candidate groups is generated by adding individual contiguous features to the original group. For text, the candidate groups correspond to adding one adjacent word onto the current phrase. Candidate groups are ranked according to the interaction score, which is the difference between the score of the candidate and original groups. Clustering terminates after an application-specific criterion is met. For sentiment classification, it stops once all words are selected. With the agglomerative clustering algorithm, explanation effectively identifies phrase-level classification patterns without evaluating all possible phrases in the sentence even when a predefined hierarchy does not exist.

3 Layer-wise Model interpretation

In this section we will explain our framework for evaluating the intermediate values of the Bert model encoders. As it is shown in Figures 1, Bert model has multiple encoder layers followed by pooling and FC layers. We used two methods to study the effect of the encoders and their contribution into the final score of words or phrases. In order to capture the layers contribution we got the intermediate vector values and feed them directly to the latent layers. Then we used the two discussed mechanism, SCD and SOC in order to calculate the score of the phrases and finally visualize the variance of scores of input features through the layers. Figure 2 shows two approaches for evaluating the contribution of encoder layers. We studied the effect of adding encoder layers. In other words how adding more encoder layers will contribute to the score changes of a phrase (Figure 2-a). We get the output of each encoder and send them separately to FC layers. We called this as adding layer approach. In the second approach we observed the influence of excluding encoders (Figure 2-b). In this approach we exclude one encoder at a time and get the output of last encoder. Therefore, there exists 11 encoders out of 12 every time and one is dropped.

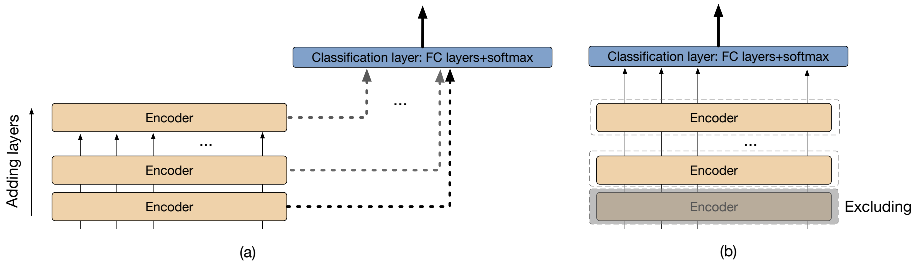

Figure 2: (a) Adding encoder layer and (b) excluding encoder layer approaches.

4 Evaluation

4.1 Experimental Setup

We used the implementation of the [5] for the initial setup[1]. We used the Bert model for text classification. We used sentiment analysis dataset known as the Stanford Sentiment Treebank-2 (SST-2) dataset. [12]. The SST-2 dataset provides sentiment polarity scores for all the phrases on the nodes of the constituency parsing trees with positive/negative labels for each sentence in the corpus. The Transformer model is fine-tuned from pretrained BERT models, which have 12 layers and 768 hidden units of per representation. We also pre-trained a language model on the training set which is used for sampling in the SCD and SOC algorithms. We evaluated 50 samples and extracts the scores for each word/phrase. Then By applying the agglomerative clustering algorithm and defining a threshold score the hierarchies are constructed automatically.

4.2 Results

4.2.1 Effect of adding encoder layers

Figures 3 and 4 shows the evolution of hierarchical explanations and phrase interaction for a sample input for SCD and SOC algorithms respectively. In SOC the final classification start from negative and gradually reach to positive prediction while in SOC it is positive in the beginning with less severity. This is true for most of the samples more or less. As illustrated the prediction is stable after first half of the layers. Through more layers, more accurate phrase interaction will be discovered.

As illustrated in Figure 5A, we can see when we add more layers into the model, the final value increases. From the layer 6-7, the rate of increase in accelerating. One possible explanation is that earlier layers capture some syntactic information, while higher layers have semantic meanings, thus the contribution starts to explode. We would like to see whether there is some interesting patters. For each element, which might be a token or a long clause, it has 12 values corresponding to each layer. We calculate the variance for each token, and sort it. The result is illustrated in Figure 5B, the top 3 elements are always some long clause, which indicate the upper part of trees, while

[1] https://github.com/INK-USC/hierarchical-explanation

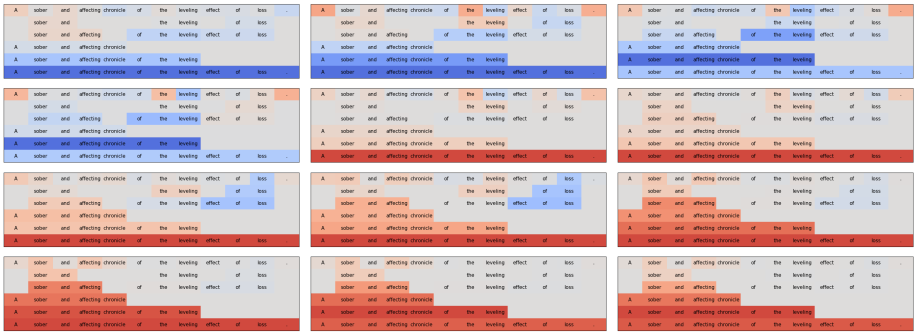

Figure 3: Evolution of phrase interaction through layers using SCD algorithm, Top left: layer 1 to bottom right: layer 12.

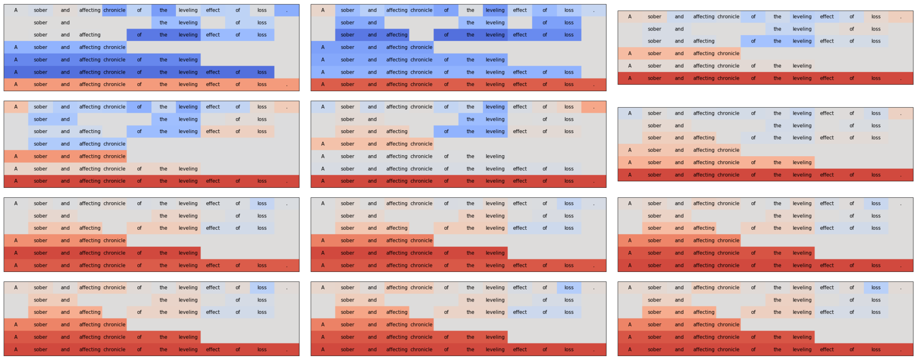

Figure 4: Evolution of phrase interaction through layers using SOC algorithms, Top left: layer 1 to bottom right: layer 12.

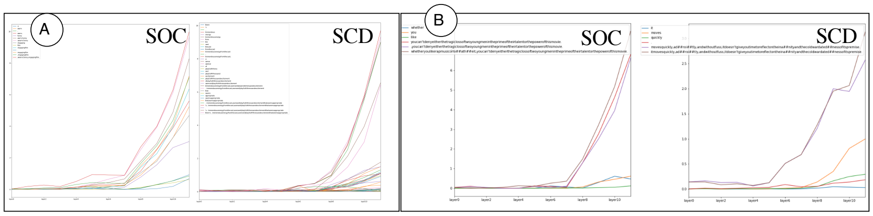

Figure 5: A: the effect of adding layers of all tokens B: the effect of adding layers of highest and lowest variance

the tokens have little variance. The reason is the sentiment of each word is easier to stabilize, the clause fluctuate a lot with multiple words.

4.2.2 Effect of excluding encoder layers

Figures 6 and 7 shows the effect of excluding layers on the phrase interaction. Since there are 11 encoder layer active the results are consistent through excluding different layers. Same as adding layer experiments SOC algorithms shows lower variance tree by tree in comparison with SCD algorithms.

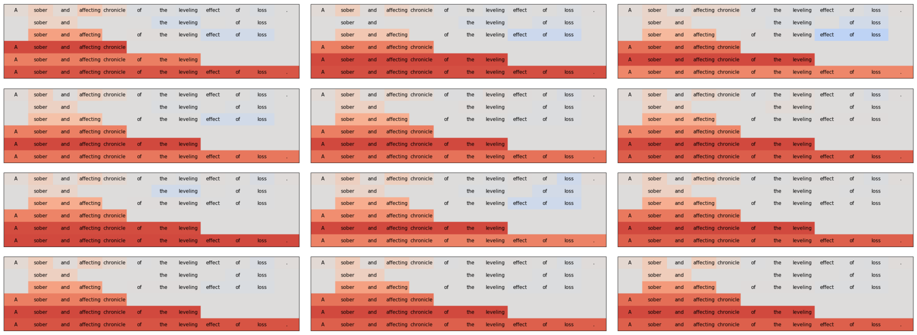

Figure 6: Effect of excluding layers on phrase interaction using SCD, Top left: layer 1 to bottom right: layer 12.

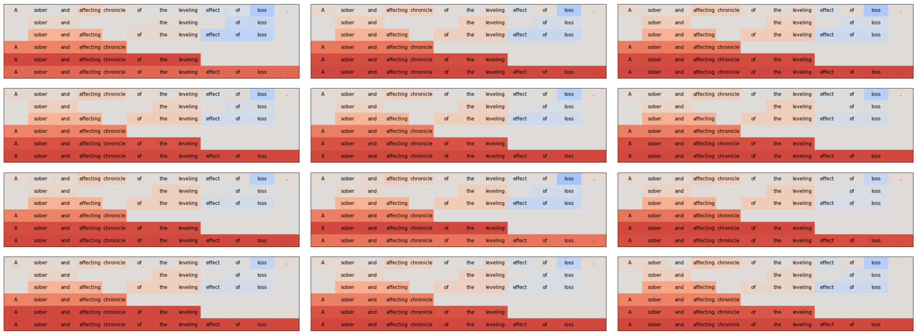

Figure 7: Effect of excluding layers on phrase interaction using SOC, Top left: layer 1 to bottom right: layer 12.

We also compare the layerwise transition for excluding experiment, the results are somewhat different between SOC and SCD methods, shown in Figure 8. We can identify the decreasing pattern for layer 8 in SOC, while the decreasing patterns exist both in layer 3 and layer 8 in SCD. At these specific layers, they may have some positive contribution to the final output. That's why when we exclude it from the model, the final output decrease.

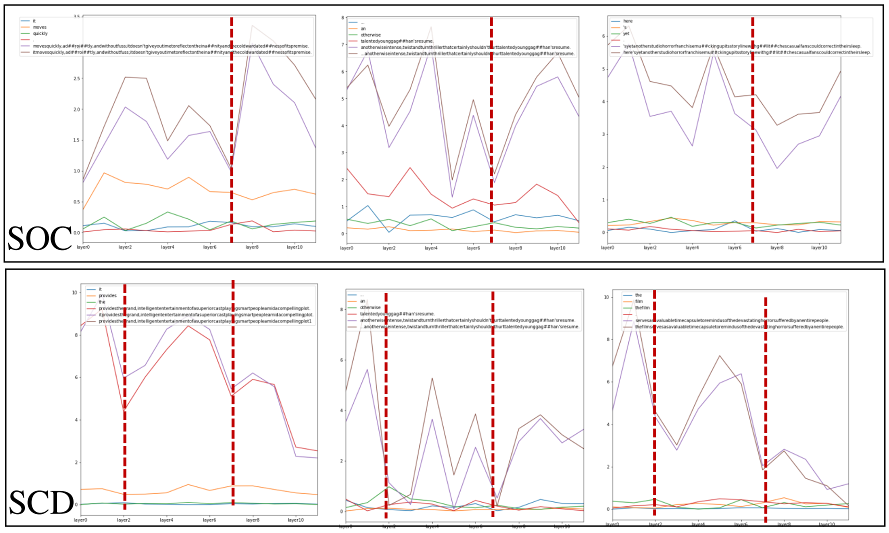

Figure 8: The effect of exclusing layers for both SOC and SCD methods

5 Conclusion

In this work we used two state of the art algorithms, SCD and SOC for not only finding the ability of Bert transformer to model the complicated word and phrase interaction but also how these interactions evolve through multiple layers. We evaluated the Bert model from two perspectives: how adding more encoder layers will help and contribute the the phrase interaction? and how excluding a layer affects the phrase interaction in the text input? The excluding approach showed that a single layer does not have distinguishable effect on phrase interaction and layers work collaboratively to discover the phrase interaction. The excluding approach is more stable then the adding approach, while we can improve the adding approach by re-training the FC layer, which is also our future direction. In this regard, we found out that prior layers mostly make decision on whole input while the later layers try to find the smaller phrase interaction in the input.

References

[1] L. Arras, G. Montavon, K.-R. Müller, and W. Samek. Explaining recurrent neural network predictions in sentiment analysis. *arXiv preprint arXiv:1706.07206*, 2017.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] F. Godin, K. Demuynck, J. Dambre, W. De Neve, and T. Demeester. Explaining character-aware neural networks for word-level prediction: Do they discover linguistic rules? *arXiv preprint arXiv:1808.09551*, 2018.

[5] X. Jin, J. Du, Z. Wei, X. Xue, and X. Ren. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*, 2019.

[6] J. Jumelet, W. Zuidema, and D. Hupkes. Analysing neural language models: Contextual decomposition reveals default reasoning in number and gender assignment. *arXiv preprint arXiv:1909.08975*, 2019.

[7] J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] W. J. Murdoch, P. J. Liu, and B. Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. *arXiv preprint arXiv:1801.05453*, 2018.

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[11] C. Singh, W. J. Murdoch, and B. Yu. Hierarchical interpretations for neural network predictions. *arXiv preprint arXiv:1806.05337*, 2018.

[12] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. 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, 2013.