PRELIMINARY VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference

Building Interpretable Interaction Trees for Deep NLP Models

**Die Zhang, HuiLin Zhou, Hao Zhang, Xiaoyi Bao,
Da Huo, Ruizhao Chen, Xu Cheng, Mengyue Wu, Quanshi Zhang***
Shanghai Jiao Tong University
{zizhan52, zhouhuilin116, 1603023-zh, zjbaoxiaoyi}@sjtu.edu.cn
{sjtuhuoda, stelledge, xcheng8, mengyuewu, zqs1022}@sjtu.edu.cn

Abstract

This paper proposes a method to disentangle and quantify interactions among words that are encoded inside a DNN for natural language processing. We construct a tree to encode salient interactions extracted by the DNN. Six metrics are proposed to analyze properties of interactions between constituents in a sentence. The interaction is defined based on Shapley values of words, which are considered as an unbiased estimation of word contributions to the network prediction. Our method is used to quantify word interactions encoded inside the BERT, ELMo, LSTM, CNN, and Transformer networks. Experimental results have provided a new perspective to understand these DNNs, and have demonstrated the effectiveness of our method. *The code will be released when the paper is accepted.*

Introduction

Deep neural networks (DNNs) have shown promise in various tasks of natural language processing (NLP), but a DNN is usually considered as a black-box model. In recent years, explaining features encoded inside a DNN has become an emerging direction. Based on the inherent hierarchical structure of natural language, many methods use latent tree structures of language to guide the DNN to learn interpretable feature representations (Choi, Yoo, and Lee 2018; Drozdov et al. 2019; Shen et al. 2018, 2019; Shi et al. 2018; Tai, Socher, and Manning 2015; Wang, Lee, and Chen 2019; Yogatama et al. 2016). However, the interpretability usually conflicts with the discrimination power (Bau et al. 2017). There is a considerable gap between pursuing the interpretability of features and pursuing superior performance.

Therefore, in this study, we aim to explain a trained black-box DNN in a post-hoc manner, so that the explanation of the DNN does not affect its performance. This is essentially different from previous studies of designing new network architectures or losses to learn interpretable features, *e.g.* physically embedding tree structures into a DNN.

Given a trained DNN, in this paper, we propose to analyze interactions among input words, which are used by the DNN to make a prediction. Our method generates a tree structure to objectively reflect interactions among words. Mathematically, the interaction of several words is quantified as the difference of the contribution between the case when these words contribute jointly to the prediction and the case when each individual word contributes independently to the prediction. The interaction between words may bring either positive or negative effects on the prediction. For example, the word *green* and the word *hand* in the sentence *he is a green hand* have a strong and positive interaction to the prediction of the person's identity, because the words *green* and *hand* indicate a "novice" jointly, rather than work individually to represent a hand with a green color.

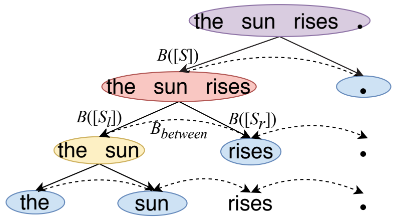


Figure 1: A tree to represent interactions among words. The tree is built to explain a trained DNN. Each leaf node (blue) represents an input word in the sentence. Each non-leaf node encodes the significance of interactions within a constituent.

The core challenge in this study is to guarantee the objectiveness of the explanation. *I.e.* the tree needs to reflect true interactions among words without significant bias. The Shapley value is widely considered as a unique unbiased estimation of the word contribution (Lundberg and Lee 2017), which satisfies four desirable properties, *i.e. linearity, dummy, symmetry and efficiency* (Grabisch and Roubens 1999). Thus, we define the interaction benefit among words based on the Shapley value. Let us consider a constituent with m words. $\phi_1, \phi_2, \ldots, \phi_m$ denote numerical contributions of each word to the prediction of a DNN, respectively. ϕ_{all} represents the numerical contribution of the entire constituent to the prediction. Hence, $B = \phi_{all} - \sum_{i=1}^{m} \phi_i$ measures the interaction benefit of this constituent. If $B > 0$, interactions among these m words have positive effects on the prediction; otherwise, negative effects. Here, $\phi_1, ..., \phi_m, \phi_{all}$ can be computed as Shapley values.

*Corresponding author. Quanshi Zhang is with the John Hopcroft Center and the MoE Key Lab of Artificial Intelligence, AI Institute, at the Shanghai Jiao Tong University, China.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Given a trained DNN and an input sentence with n words, Figure 1 shows the tree structure that reflects word interactions encoded inside the DNN. In the tree, n leaf nodes represent n input words. Each non-leaf node corresponds to a constituent of the input sentence. A parent node connects two child nodes with significant interaction benefits. We use the parent node to encode interactions among its child subconstituents. More specifically, there are two types of interactions among words, *i.e.* (1) interactions within a constituent and (2) interactions between constituents.

• **Interactions within a constituent** exist among any two or more words in the constituent. For the sentence "*the sun is shining in the sky*," interactions within the constituent *in the sky* consist of interactions among all combinations of words, including interactions (1) between (*in*, *the*), (2) between (*the*, *sky*), (3) between (*in*, *sky*) and (4) among (*in*, *the*, *sky*).

• **Interactions between constituents.** In the aforementioned sentence, interactions between the constituent *the sun* and its adjacent constituent *is shining* are composed of all potential interactions among all combinations of words from the two constituents, including interactions between (1) (*the*, *is*), (2) (*the*, *shining*), (3) (*sun*, *is*), (4) (*sun*, *shining*), (5) (*the*, *is shining*), (6) (*sun*, *is shining*), (7) (*the sun*, *is*), (8) (*the sun*, *shining*), (9) (*the sun*, *is shining*).

The tree selects and encodes the most salient interactions among words, in order to reveal the signal processing in a DNN. We further propose additional metrics to diagnose interactions among words, *e.g.* the quantification of interactions within a constituent, the quantification of interactions between two adjacent constituents, and ratios of interactions that are modeled and unmodeled by the tree.

Theoretically, our method can be used as a generic tool to analyze various DNNs, including the BERT (Devlin et al. 2018), ELMo (Peters et al. 2018), LSTM (Hochreiter and Schmidhuber 1997), CNN (Kim 2014) and Transformer (Vaswani et al. 2017). Experimental results have demonstrated the effectiveness of our method.

Contributions of this paper can be summarized as follows. (1) We propose a method to extract and quantify interactions among words. (2) A tree structure is automatically generated to represent salient interactions encoded in a DNN. (3) We further design six metrics to analyze interactions, which provides new perspectives to understand DNNs.

Related Work

Hierarchical representations of natural language. Many studies integrated hierarchical structures of natural language into DNNs for better representations (Tai, Socher, and Manning 2015; Dyer et al. 2016; Wang et al. 2019; Wang, Lee, and Chen 2019). Chung, Ahn, and Bengio (2017) revised an RNN to learn the hierarchical structure of sequential data. Shen et al. (2019) designed a novel recurrent architecture to automatically capture the latent tree structure of an input sentence. Other studies learned syntactic parsers (Drozdov et al. 2019; Htut, Cho, and Bowman 2019; Kitaev, Cao, and Klein 2019; Li, Mou, and Keller 2019; Li and Eisner 2019; Mrini et al. 2019), although these methods pursued a high parsing accuracy, instead of explaining the DNN. Essentially, the learning of the syntactic parser aimed to make the parser fit syntactic structures defined by human experts. In contrast, we intend to provide a method to analyze DNNs in a post-hoc manner, without being affected by the subjective bias from humans.

Post-hoc explanations of DNNs: Some studies measured the representation capacity to understand DNNs (Guan et al. 2019; Cheng et al. 2020; Liang et al. 2020). Voita, Sennrich, and Titov (2019) studied how token representations changed from layer to layer. Reif et al. (2019); Raganato and Tiedemann (2018) exploited the attention weights of models to analyze syntactic and semantic information encoded in internal representations. Yogatama et al. (2018) evaluated the ability of various RNNs to capture syntactic dependencies. Another line of research was to estimate word importance to the prediction based on Shapley values (Shapley 1953), such as SHAP (Lundberg and Lee 2017), L/C-Shapley (Chen et al. 2019). Murdoch, Liu, and Yu (2018) estimated contributions of input words to the prediction of an LSTM as well as inter-word relationships.[1] Singh, Murdoch, and Yu (2019) and Jin et al. (2020) generated hierarchical explanations for word/phrase importance.

Unlike above studies of estimating attribution/saliency /contribution/importance of input units, we focus on interactions among words encoded inside DNNs. Janizek, Sturmfels, and Lee (2020) explained pairwise feature interactions by extending the Integrated Gradients explanation method. Greenside et al. (2018) identified interactions between all pairs of discrete features in an input DNA sequence. Cui, Marttinen, and Kaski (2019) estimated global pairwise interactions from a trained Bayesian neural network. Tsang, Cheng, and Liu (2018) detected statistical interactions from the weights of feedforward neural networks. Tsang et al. (2018) proposed to separate feature interactions based on regularization, and could only be applied to fully connected multilayer perceptrons. Lundberg, Erion, and Lee (2018) defined SHAP interaction values to quantify interaction effects between two features. Chen, Zheng, and Ji (2020) generated hierarchical explanations of DNNs based on the SHAP interaction value. Chen and Jordan (2019) used a *"predefined"* syntactic constituency structure to assign an importance score to each word, and to quantify interactions[2] between sibling nodes on a parse tree. This study had considerable impacts, but it did not learn the linguistic structure.

However, these studies mainly focus on interactions between two variables (Janizek, Sturmfels, and Lee 2020; Greenside et al. 2018; Cui, Marttinen, and Kaski 2019; Lundberg, Erion, and Lee 2018; Chen, Zheng, and Ji 2020) or are limited to multilayer perceptron architectures (Tsang, Cheng, and Liu 2018; Tsang et al. 2018). Instead, we aim to quantify interactions among multiple variables in DNNs with arbitrary architectures without any prior linguistic structure. More specifically, our method uses a tree to organize the extracted interactions hierarchically.

Shapley values. The Shapley value (Shapley 1953) was first introduced in game theory. Given a game with multi-

[1] Although they called the inter-word relationships *interactions*, such interactions had essential difference from our interactions.

[2] The deviation of composition from linearity.

ple players, each player is supposed to pursue a high award. Some players may form a coalition to pursue more awards. The Shapley value is widely considered as a unique unbiased approach to fairly allocating the total award of a coalition to each player (here, the award of each player is also termed the *contribution* of this player).

Given a game v with n players, $N = \{1, 2, ..., n\}$, let $2^N = \{S|S \subseteq N\}$ denote all the potential subsets of N. $v\colon 2^N \mapsto \mathbb{R}$ is a set function mapping from each subset to a real number. For any subset of players $S \subseteq N$, $v(S)$ represents the score obtained by the set of players S. $v(\varnothing)$ represents the baseline score without any players. Thus, $v(S) - v(\varnothing)$ corresponds to the award obtained by players in S. Considering the player $a \notin S$, if player a joins S, $v(S \cup \{a\}) - v(S)$ is considered as the marginal award/contribution of player a. The Shapley value $\phi(a)$ is an unbiased estimation of numerical contribution of player a in the game as follows.

$$\phi(a) = \sum_{S\subseteq N\setminus\{a\}} \frac{(|N| - |S| - 1)!|S|!}{|N|!}(v(S\cup\{a\})-v(S)) \quad (1)$$

The fairness of Shapley values is ensured by the four following properties (Weber 1988):

- Linearity property: If two games v and w are combined into a single game $v + w$, then the Shapley value of each player $a\in N$ can be added, *i.e.* $\phi(a|v+w)=\phi(a|v)+\phi(a|w)$.
- Dummy property: A dummy player $a\in N$ satisfies $\forall S\subseteq N\setminus\{a\}, v(S\cup\{a\})=v(S)+v(\{a\})$. Then, $\phi(a) = v(\{a\})- v(\varnothing)$, *i.e.* player a has no interaction to any coalition.
- Symmetry property: Given two players $a, b \in N$, if $\forall S\subseteq N\setminus\{a, b\}, v(S\cup\{a\}) = v(S\cup\{b\})$, then $\phi(a) = \phi(b)$.
- Efficiency property: The overall award can be distributed to all players, *i.e.* $\sum_{a\in N} \phi(a) = v(N) - v(\varnothing)$.

Due to the exponential number of sets in N, the computation of Shapley values is NP-hard. A sampling-based method (Castro, Gómez, and Tejada 2009) can be used to approximate Shapley values.

Algorithm

Interactions in game theory

Interactions between two players. In game theory, some players may interact with each other, and form a coalition to win a higher award. The interaction between two players is quantified as the additional award when the two players collaborate *w.r.t.* when they play individually. Considering that the Shapley value is an unbiased estimation of each player's award/contribution (Lundberg and Lee 2017), we quantify interactions based on the Shapley value. Suppose that there are n players $N = \{1, 2, ..., n\}$ in a game v. Without loss of generality, we randomly select a pair of players $a, b \in N$. Shapley values of players a and b are denoted by $\phi(a)$ and $\phi(b)$, respectively. If players a and b cooperate to form a coalition $S_{ab} = \{a, b\}$, we can consider this coalition as a new singleton player, which is represented using brackets, $[S_{ab}]$. *In this way, the game can be considered to have* $n - 1$ *players, and one of them is the singleton player* $[S_{ab}]$. *I.e.* a and b always appear together in the game. The interaction benefit between a and b is defined as $B([S_{ab}]) = \phi^{N\setminus\{a,b\}\cup\{[S_{ab}]\}}([S_{ab}])-(\phi^{N\setminus\{b\}}(a)+ \phi^{N\setminus\{a\}}(b))$. $N \setminus \{a, b\} \cup \{[S_{ab}]\}$ represents the set of players in N excluding a, b and being added a new singleton player $[S_{ab}]$. The absolute value of the interaction benefit $|B([S_{ab}])|$ represents the significance of the interaction. $B([S_{ab}]) > 0$ indicates a cooperative relationship between players a and b. Whereas, $B([S_{ab}]) < 0$ indicates an adversarial relationship between players a and b.

Extension to interactions among multiple players. We extend the two-player interaction to interactions among multiple players. When the game has n players, let us consider a subset of players $S \subsetneq N$ as a coalition, which is regarded as a new singleton player $[S]$. The interaction benefit of the coalition S is defined as follows.

$$B([S]) = \phi^{(N\setminus S)\cup\{[S]\}}([S]) - \sum\nolimits_{a\in S} \phi^{(N\setminus S)\cup\{a\}}(a) \quad (2)$$

In this way, the interaction benefit measures the additional award/contribution brought by the singleton player $[S]$ *w.r.t.* the individual award/contribution of each player computed in Equation (1) without requiring all players in S to appear together. The Shapley value $\phi^{(N\setminus S)\cup\{[S]\}}([S])$ is computed only considering the set of players when we remove all players in S from N and add a new singleton player $[S]$ in the game. Similarly, $\phi^{(N\setminus S)\cup\{a\}}(a)$ is computed only considering the set of players when we remove all players in S from N and add the player a. If $B([S])$ is greater/less than 0, interactions of players in S have positive/negative effects, revealing the cooperative/adversarial relationship among players.

Furthermore, players in S can be divided into two disjoint subsets S_1, S_2 (*i.e.* $S_1\cap S_2 = \varnothing, S_1\cup S_2 = S$). Accordingly, the interaction benefit can be decomposed into three terms:

$$B([S]) = B([S_1]) + B([S_2]) + B_{\mathit{between}}(S_1, S_2) \quad (3)$$

The first and second terms $B([S_1])$ and $B([S_2])$ indicate interaction benefits among players within S_1 and S_2, respectively. The third term $B_{\mathit{between}}(S_1, S_2)$ indicates interaction benefits among players selected from both S_1 and S_2. $B_{\mathit{between}}(S_1, S_2)$ will be introduced in detail later.

Properties of interaction benefits. The overall interaction benefit, $B([S]), S \subseteq N$, can be decomposed into elementary interaction components $I^N(S)$. The elementary interaction component was originally proposed in (Grabisch and Roubens 1999). The elementary interaction component $I^N(S)$ measures the marginal benefit received from the coalition $[S]$, from which benefits of all potential smaller coalitions $S' \subsetneq S$ are removed. For example, let $S = \{a, b, c\}$. Then, $I^N(S)$ measures interactions caused by $[S] = (a, b, c)$, and ignores all potential interactions caused by coalitions of $(a, b), (a, c), (b, c), (a), (b), (c)$. Therefore, the elementary interaction component is formulated as follows.

$$I^N(S) = I^{(N\setminus S)\cup\{[S]\}}([S]) - \sum_{S'\subsetneq S, S'\neq\varnothing} I^{(N\setminus S)\cup S'}(S') \quad (4)$$

In particular, for any singleton player $[S]$, we have $I^{(N\setminus S)\cup\{[S]\}}([S]) = \phi^{(N\setminus S)\cup\{[S]\}}([S])$. Thus, we can compute $I^N(S)$ via dynamic programming. We prove that

$B([S])$ can be decomposed into elementary interaction components (the supplementary material shows the proof).

$$B([S]) = \sum\nolimits_{S' \subseteq S, |S'|>1} I^{(N \setminus S) \cup S'}(S') \tag{5}$$

Fine-Grained analysis of interactions between two sets of players

Interactions between two sets of players $B_{between}(S_1, S_2)$ can be further decomposed into three parts ψ^{inter}, ψ_1^{intra}, ψ_2^{intra}. Please see the supplementary material for the proof.

$$B_{between}(S_1, S_2) = \psi^{inter} + \psi_1^{intra} + \psi_2^{intra} \tag{6}$$

where

$$\begin{aligned}
\psi^{inter} &= \sum\nolimits_{L \subseteq S, L \not\subseteq S_1, L \not\subseteq S_2, |L|>1} I^{(N \setminus S) \cup L}(L) \\
\psi_1^{intra} &= \sum_{L \subseteq S_1, |L|>1} I^{(N \setminus S) \cup L}(L) - \sum_{L \subseteq S_1, |L|>1} I^{(N \setminus S_1) \cup L}(L) \\
&= B([S_1])|_{N'=(N \setminus S_2)} - B([S_1]) \\
\psi_2^{intra} &= \sum_{L \subseteq S_2, |L|>1} I^{(N \setminus S) \cup L}(L) - \sum_{L \subseteq S_2, |L|>1} I^{(N \setminus S_2) \cup L}(L) \\
&= B([S_2])|_{N'=(N \setminus S_1)} - B([S_2])
\end{aligned} \tag{7}$$

ψ^{inter} represents all potential interaction benefits caused by sets of players whose elements are selected from both S_1 and S_2. $B([S_1])|_{N'=(N \setminus S_2)}$ denotes interaction benefits of the singleton player $[S_1]$, when the set of players in the game is $N' = (N \setminus S_2)$. ψ_1^{intra} indicates the difference of internal interactions among players in S_1 *w.r.t.* the absence and presence of players in S_2.

Interactions encoded inside a DNN

We aim to analyze interactions among words, which are encoded inside a trained DNN. Given an input sentence with n words, we construct a tree to disentangle and quantify interactions among input words. We first introduce Shapley values of input words *w.r.t.* the prediction of the DNN. Here, we consider each word as a player, and the scalar output of a DNN as the aforementioned score v in the game. If a DNN has a scalar output, we can take the scalar output as the score v. If the DNN outputs a vector for multi-category classification, we select the score before the softmax layer corresponding to the predicted class as the score. To compute $v(S)$, we mask words in $N \setminus S$ in the input sentence, and feed the modified input into the DNN. The embedding of the masked word is set to a dummy vector, which refers to a padding of the input to the DNN. Then, the Shapley value of each word/player a is approximated using a sampling-based method (Castro, Gómez, and Tejada 2009).

As Figure 1 shows, we construct a binary tree with n leaf nodes. Each leaf node represents a word, while each non-leaf node represents a constituent. Two adjacent nodes with strong interactions will be merged into a node in the next layer. For each sub-structure of a parent node S with two child nodes S_l and S_r, according to Equation (3), $B([S])$ can be recursively decomposed into the sum of interaction benefits between two child nodes of all non-leaf nodes. Please see the supplementary material for the proof.

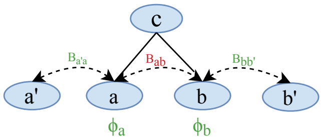


Figure 2: Interaction benefits between constituents. The interaction benefit B_{ab} is more significant than $B_{a'a}$ and $B_{bb'}$, so the tree merges a and b to form a coalition c.

$$\begin{aligned}
B([S]) &= B([S_l]) + B([S_r]) + B_{between}(S_l, S_r) \\
&= B([S_{ll}]) + B([S_{lr}]) + B([S_{rl}]) + B([S_{rr}]) \\
&+ B_{between}(S_{ll}, S_{lr}) + B_{between}(S_{rl}, S_{rr}) \\
&+ B_{between}(S_l, S_r) = \sum_{H \in non\text{-}leaf\ nodes} B_{between}(H_l, H_r)
\end{aligned} \tag{8}$$

Metrics for interactions and the construction of a tree

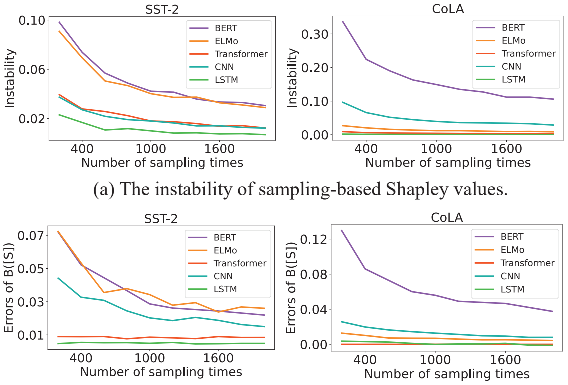
Metrics for interactions. Besides $B([S_l])$, $B([S_r])$ and $B_{between}(S_l, S_r)$, we define three additional metrics to provide insightful analysis of interactions among words. Let us consider a sub-structure of a parent node c (corresponding to the constituent S) and two child nodes a and b (corresponding to sub-constituents S_l and S_r). As Figure 2 shows, a' is the left adjacent node of a, and b' is the right adjacent node of b. We propose the metric "*density of modeled interactions*" for a candidate coalition such as $\{a, b\}$, denoted by $r(a, b)$. This metric measures the ratio of interaction benefits between two adjacent nodes a and b to the total interaction benefits related to a and b. The density of the modeled interactions is approximated as follows.

$$\begin{aligned}
r(a, b) &= \frac{\textit{interaction benefits between a and b}}{\textit{total interaction benefits related to a and b}} \\
&\approx \frac{|B_{ab}|}{|B_{ab}| + |B_{a'a}| + |B_{bb'}| + |\phi_a| + |\phi_b|}
\end{aligned} \tag{9}$$

where $B_{ab} = B_{between}(S_a, S_b)$, ϕ_a and ϕ_b can be approximated as $\phi^{(N \setminus S_a) \cup \{[S_a]\}}([S_a])$ and $\phi^{(N \setminus S_b) \cup \{[S_b]\}}([S_b])$, respectively. To measure interaction benefits that are not represented by the tree, a metric called "*density of unmodeled interactions*" denoted by $s(a, b)$ is given.

$$\begin{aligned}
s(a, b) &= \frac{\textit{unmodeled interaction benefits}}{\textit{total interaction benefits related to a and b}} \\
&\approx \frac{|B_{a'a}| + |B_{bb'}|}{|B_{ab}| + |B_{a'a}| + |B_{bb'}| + |\phi_a| + |\phi_b|}
\end{aligned} \tag{10}$$

Note that neither $r(a, b)$ nor $s(a, b)$ is an accurate estimation of the ratio of interactions. If two constituents are far away (*e.g.* not adjacent), their interaction benefits are usually small and sometimes can be neglected. Therefore, we only consider interaction benefits between adjacent nodes (*i.e.* $B_{a'a}$, B_{ab}, $B_{bb'}$). We have demonstrated very little effects of such neglection in Table 1. In addition, according to Equation (6), we have $B_{between}(S_l, S_r) = \psi^{inter} + \psi_l^{intra} +$



(a) The instability of sampling-based Shapley values.

(b) Errors of the estimated interaction benefits.

Figure 3: Evaluation of the reliability of the method.

ψ_r^{intra}. Therefore, we define the following metric to measure the ratio of inter-constituent interactions.

$$t = |\psi^{inter}|/(|\psi^{inter}| + |\psi_l^{intra} + \psi_r^{intra}|) \tag{11}$$

Construction of a tree. We use the metric $r(a, b)$ in Equation (9) to quantify the significance of interactions between two adjacent constituents, and to guide the construction of the tree. We are given a trained DNN and an input sentence. The DNN can be trained for various tasks, such as sentiment classification, and the estimation of linguistic acceptability. We construct the tree in a bottom-up manner. Let Ω denote the set of current candidate nodes to merge. In the beginning, each word a_i of the input sentence is initialized as a leaf node, $\Omega = \{a_1, a_2, ..., a_n\}$. In each step, we compute the value of each pair of adjacent nodes $r(a_i, a_{i+1})$. Then, we select and merge two adjacent nodes with the largest value of $r(a_i, a_{i+1})$. In this way, we use a greedy strategy to build up the tree, so that salient interactions among words are represented.

Experiments

Instability and accuracy of Shapley values. According to Equation (1), the accurate computation of Shapley values is NP-hard. Castro, Gómez, and Tejada (2009) proposed a sampling-based method to approximate Shapley values with polynomial computational complexity. In order to evaluate the instability of $B([S])$, we quantified the change of the instability of Shapley values along with the increase of the number of sampling times. Let us compute the Shapley value $\phi(a)$ for each word by sampling T times. We repeated such a procedure of computing Shapley values two times. Then, the instability of the computation of Shapley values was measured as $2||\boldsymbol{\phi} - \boldsymbol{\phi}'||/(||\boldsymbol{\phi}|| + ||\boldsymbol{\phi}'||)$ where $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ denoted two vectors of word-wise Shapley values computed in these two times. The overall instability of Shapley values was reported as the average value of the instability of all sentences. Figure 3 (a) shows the change of the instability of Shapley values along with the number of sampling times T. When $T \geq 1000$, we obtained stable Shapley values.

In addition, we also evaluated the accuracy of the estimation of interaction benefits $B([S])$. The problem was that the ground truth value of $B([S])$ was computed using the NP-hard brute-force manner, according to Equation (1). Considering the NP-hard computational cost, we only conducted such evaluations on sentences with no more than 10 words. The average absolute difference (*i.e.* the error) between the estimated $B([S])$ and its ground truth value over all sentences is reported in Figure 3 (b). We found that the estimated interaction benefits were accurate enough when the number of sampling times was greater than 1000.

# of merges	BERT	ELMo	CNN	LSTM
1	0.00	0.02	0.01	0.06
2	0.00	0.06	0.02	0.13
3	0.00	0.12	0.02	0.19
4	0.03	0.15	0.07	0.15
5	0.03	0.16	0.07	0.14

Table 1: The rate of incorrect extractions of word interactions, which verifies the assumption that effects of non-adjacent nodes can be neglected on the SST-2 dataset.

We found that the BERT model exhibited much higher instability and errors than other models in Figure 3. It was because the BERT model had much stronger representation power than other models, and thus encoded more complex interactions, which was also verified in (Guan et al. 2019). Thus, the BERT model required more sampling times.

Effects of non-adjacent nodes. To compute the density of modeled interactions $r(a, b)$, we only considered interaction benefits between two adjacent nodes, and assumed that interactions of non-adjacent nodes were much less significant than those of adjacent nodes. To verify this assumption, we defined the following metric to quantify the interaction benefit $r'(a, c)$ between two non-adjacent nodes a and c, and evaluated whether the most salient interaction between adjacent nodes a, b detected by our method was more significant than interactions between all potential non-adjacent nodes. We use $r'(a, c) = |B_{ac}|/(|B_{ac}| + |B_{a'a}| + |B_{aa''}| + |B_{c'c}| + |B_{cc''}| + |\phi_a| + |\phi_c|)$ to quantify the interaction density between non-adjacent nodes a and c, where a' and a'' were the left and right adjacent nodes of a, c' and c'' were the left and right adjacent nodes of c. If the interaction density $r(a, b)$ estimated by our method was higher than that between potential non-adjacent nodes, we considered this as a correct extraction of word interactions. Table 1 reports the rate of incorrect extractions of word interactions over all sentences during the construction of the tree. Based on this assumption, our method performed correctly in most cases.

Correctness of the extracted interaction. We aimed to evaluate whether the extracted interaction objectively reflected the true interaction in the model, but the core challenge was that it was impossible to annotate ground-truth interactions between words. It was because the human's understanding of word interactions was not necessarily equivalent to objective interactions encoded in a DNN. In this way, we conducted the following two experiments to evaluate the correctness of the extracted interactions.

	BERT	ELMo	CNN	LSTM	Transformer
Ours	**0.037**	**0.133**	**0.063**	**0.036**	**0.012**
HEDGE	0.033	0.131	0.023	0.004	0.008

Table 2: Comparisons of cohesion-scores for explanations of NLP models trained on the SST-2 dataset (Socher et al. 2013).

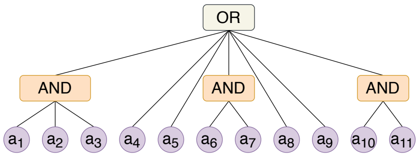

OR
AND
AND
AND
a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11}

Figure 4: An example of AND-OR models. Each leaf node is a binary variable.

Experiment 1: In order to quantitatively evaluate the validity of the extracted interactions among words, we adopted the metric *cohesion-score* proposed by Chen, Zheng, and Ji (2020) to justify a constituent containing significant interactions identified by our method. Given a constituent corresponding to a tree node $[p,q] = (a_p, ..., a_q)$ in a sentence $x = (a_1, ..., a_p, ..., a_q, ..., a_n)$, we picked a word in $[p,q]$ at a time, and inserted it into a random position in the sequence $(a_1, ..., a_{p-1}, a_{q+1}, ..., a_n)$. We repeated this process until there were no words left in $[p,q]$. Thus, we obtained a shuffled sentence $\tilde{x}$ from x. The cohesion-score measured the change of probability on the predicted class between $\tilde{x}$ and x as follows:

$$\textit{cohesion-score} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}\sum_{j=1}^{Q}(p(\hat{y}_i|\mathrm{x}_i) - p(\hat{y}_i|\tilde{\mathrm{x}}_i^{(j)})) \quad (12)$$

where x_i is the i-th sentence and $\hat{y}_i$ is the predicted class. $\tilde{\mathrm{x}}_i^{(j)}$ is the j-th shuffled sentence from x_i, and Q was set to 100. For each sentence, we only considered the most significant constituent (*i.e.* the tree node with the maximum $\phi^{N\setminus S\cup\{[S]\}}([S])$) during the construction of the tree. We used HEDGE (Chen, Zheng, and Ji 2020), which was a top-down method to recursively split a long sentence into shorter constituents, as the baseline. Besides, for a fair comparison, we reimplemented the HEDGE method to explain different NLP models trained on the SST-2 dataset. As Table 2 shows, our method outperformed HEDGE, which suggested that our method extracted more significant interactions within constituents than HEDGE.

Experiment 2: We constructed a dataset with ground-truth interactions between the inputs, as follows. The dataset was comprised of 2048 models. Each model was implemented as a boolean function, whose input was 11 binary variables $a_1, a_2, \cdots, a_{11} \in \{0,1\}$. The output of the model was a binary variable which consisted of AND, OR operations in a two-level tree structure (*e.g.* the tree in Figure 4). More specifically, we designed 1024 models where AND operations were in the first level, and OR operations were in the second level. The other 1024 models had OR operations in the first level, and AND operations in the second level. We evaluated whether the extracted interaction could reflect the true AND, OR constituents in the input.

	F1 score			Recall		
	AND-OR	OR-AND	**Avg.**	AND-OR	OR-AND	**Avg.**
Ours	45.02	45.62	**45.32**	96.60	98.94	**97.77**
SI	46.02	0.00	23.01	99.80	0.00	49.90
SI-abs	29.77	29.74	29.76	61.27	61.22	61.25
HEDGE	46.02	0.00	23.01	99.80	0.00	49.90
Random	-	-	13.18	-	-	27.78
LB	-	-	8.35	-	-	18.07
RB	-	-	8.35	-	-	18.07

Table 3: Comparisons of the correctness of the extracted interactions on AND-OR models and OR-AND models.

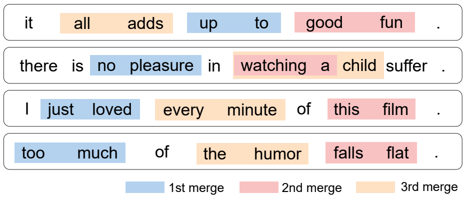

it all adds up to good fun .
there is no pleasure in watching a child suffer .
I just loved every minute of this film .
too much of the humor falls flat .
1st merge
2nd merge
3rd merge

Figure 5: Examples of the phenomenon that constituents with distinct emotional attitudes have strong interactions and are extracted in the first three steps for BERT learned on the SST-2 dataset.

The unlabeled F1 score and unlabeled recall were used to evaluate the correctness of the extracted interaction. We compared our method with six baselines. The first baseline was (Lundberg, Erion, and Lee 2018), which defined a type of two-player interaction (*i.e.* SHAP interaction), namely *SI* for short. We extended this technique to construct a tree. *I.e.* we recursively merged the two adjacent nodes with the largest SHAP interaction value. The second baseline was similar to the first one. This baseline used the largest absolute SHAP interaction value (*i.e.* the significance) to construct the tree, namely *SI-abs*. The third baseline was HEDGE (Chen, Zheng, and Ji 2020), as mentioned above. Since there was no other method to construct a tree for interactons to the best of our knowledge, the other three baselines Random, left-branching (LB) and right-branching (RB) trees (used by Shen et al. (2018) as baselines) were selected as trivial solutions. As Table 3 shows, our method outperformed all baselines. Note that theoretically, there did not exist a 100% F1 score, because the extracted binary tree was naturally different from the ground-truth n-ary tree.

Comparisons of trees generated by different DNNs. We learned DNNs for binary sentiment classification based on the SST-2 dataset (Socher et al. 2013), and learned DNNs to predict whether a sentence was linguistically acceptable based on the CoLA dataset (Warstadt, Singh, and Bowman 2018). For each task, we learned five DNNs, including the BERT (Devlin et al. 2018), the ELMo (Peters et al. 2018), the CNN proposed in (Kim 2014), the two-layer unidirectional LSTM (Hochreiter and Schmidhuber 1997), and the Transformer (Vaswani et al. 2017).

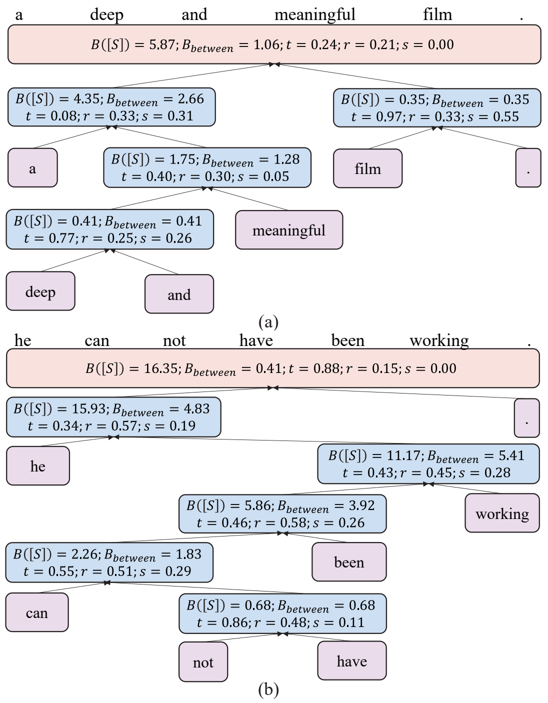

a deep and meaningful film .
$B([S]) = 5.87; B_{between} = 1.06; t = 0.24; r = 0.21; s = 0.00$
$B([S]) = 4.35; B_{between} = 2.66$ $t = 0.08; r = 0.33; s = 0.31$
$B([S]) = 0.35; B_{between} = 0.35$ $t = 0.97; r = 0.33; s = 0.55$
a
$B([S]) = 1.75; B_{between} = 1.28$ $t = 0.40; r = 0.30; s = 0.05$
film
.
$B([S]) = 0.41; B_{between} = 0.41$ $t = 0.77; r = 0.25; s = 0.26$
meaningful
deep
and
(a)
he can not have been working .
$B([S]) = 16.35; B_{between} = 0.41; t = 0.88; r = 0.15; s = 0.00$
$B([S]) = 15.93; B_{between} = 4.83$ $t = 0.34; r = 0.57; s = 0.19$
.
he
$B([S]) = 11.17; B_{between} = 5.41$ $t = 0.43; r = 0.45; s = 0.28$
$B([S]) = 5.86; B_{between} = 3.92$ $t = 0.46; r = 0.58; s = 0.26$
working
$B([S]) = 2.26; B_{between} = 1.83$ $t = 0.55; r = 0.51; s = 0.29$
been
can
$B([S]) = 0.68; B_{between} = 0.68$ $t = 0.86; r = 0.48; s = 0.11$
not
have
(b)

Figure 6: Examples of trees extracted from BERT trained on the SST-2 dataset (a) and the CoLA dataset (b), respectively. Metrics are shown in each non-leaf node.

We used our method to extract tree structures that encoded interactions among words inside various trained DNNs. Figure 6 illustrates trees extracted from BERT on different tasks. **(1)** For the sentiment analysis task, as Figure 5 shows, most trees of these DNNs usually extracted constituents with distinct positive/negative emotional attitudes in early stages. **(2)** For the linguistic acceptability task, BERT usually combined noun phrases firstly, while the subject was combined almost at last. CNN was prone to construct a tree with a "subject+verb-phrase+noun/adjective-phrase" structure. ELMo and LSTM usually extracted small constituents including a preposition or an article, *e.g.* "vacation in," "the earth." Transformer tended to encode interactions among adjacent constituents sequentially.

Analysis of significant interactions reflected by the tree: To understand how interactions among words affected the DNN to make a decision, we quantified the contribution of each word/constituent $\phi^{N\setminus S\cup\{[S]\}}([S])$ (*i.e.* ϕ_a in Equation (9)) to the model prediction with sampling times $T = 2000$ during the construction of the tree. As Figure 7 shows, the DNN encoded significant interactions (*inconsistent*, *emotional*), (*a wildly*, *inconsistent emotional*), etc. to correctly predict the whole sentence as negative. During the construction of the tree, we discovered how words/constituents interacted to affect the model prediction. This provided a better understanding of the logic encoded in the DNN.

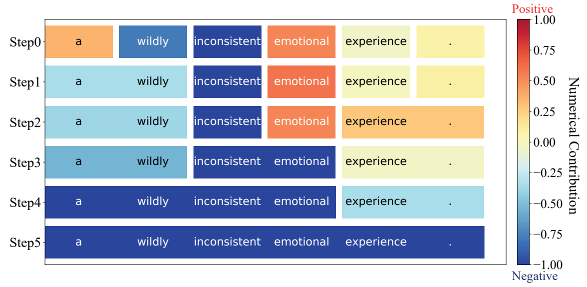


Figure 7: Effects of the extracted interactions. The extracted interactions significantly affected the contributions of constituents. For example, significant interactions between "inconsistent" and "emotional" made the positive word "emotional" negative, which eventually guided the DNN to make the correct prediction.

Dataset	BERT	ELMo	CNN	LSTM
CoLA	36.06	15.38	15.19	12.65
SST-2	17.67	16.72	11.69	29.06
	Transformer	Random	LB	RB
CoLA	3.45	15.12	2.68	60.46
SST-2	23.49	16.32	12.27	47.35

Table 4: Fitness (the unlabeled F1 score) between the extracted trees from NLP models and syntactic trees, which demonstrates that interactions encoded in a DNN are not quite related to the syntactic structure.

Comparisons of the fitness between the extracted trees and syntactic trees: Furthermore, we compared the fitness between the automatically extracted tree and the syntactic tree of the sentence. To this end, given an input sentence, we used the Berkeley Neural Parser (Kitaev and Klein 2018) to generate the syntactic tree as the ground-truth.[3] We used the unlabeled F1 score to evaluate the fitness. Experimental results are reported in Table 4, which demonstrates the logic of interactions modeled by the DNN was significantly different from human knowledge.

In addition, our method can also be applied to build a tree for interactions *w.r.t.* the computation of features in an intermediate layer.

Conclusion

In this paper, we have defined and extracted interaction benefits among words encoded in a DNN, and have used a tree structure to organize word interactions hierarchically. Besides, six metrics are defined to disentangle and quantify interactions among words. Our method can be regarded as a generic tool to objectively diagnose various DNNs for NLP tasks, which provides new insights of these DNNs.

[3] The parser's performance was good enough to take its parsing results as ground-truth.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (U19B2043 and 61906120).

Ethics Statement

This study has broad impacts on the understanding of signal processing in DNNs for NLP tasks. Our work provides researchers in the field of explainable AI with a generic tool to quantify the inter-word interactions encoded by a trained DNN. Currently, existing methods mainly focus on interactions between two variables or two words. Our research proposes new metrics to quantify interactions among multiple variables and develops a method to build up a tree representing the hierarchical structures of interactions. As a generic tool to analyze DNNs, we have applied our method to classic DNNs and have obtained several new insights on signal processing encoded in DNNs for NLP tasks.

References

Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Castro, J.; Gómez, D.; and Tejada, J. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research* .

Chen, H.; Zheng, G.; and Ji, Y. 2020. Generating hierarchical explanations on text classification via feature interaction detection. In *ACL*. URL https://arxiv.org/abs/2004.02015.

Chen, J.; and Jordan, M. I. 2019. Ls-tree: Model interpretation when the data are linguistic. *arXiv preprint arXiv:1902.04187* .

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2019. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *International Conference on Learning Representations*.

Cheng, X.; Rao, Z.; Chen, Y.; and Zhang, Q. 2020. Explaining Knowledge Distillation by Quantifying the Knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Choi, J.; Yoo, K. M.; and Lee, S.-g. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Chung, J.; Ahn, S.; and Bengio, Y. 2017. Hierarchical Multiscale Recurrent Neural Networks. In *International Conference on Learning Representations*.

Cui, T.; Marttinen, P.; and Kaski, S. 2019. Learning Global Pairwise Interactions with Bayesian Neural Networks. *arXiv: Learning* .

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

Drozdov, A.; Verga, P.; Yadav, M.; Iyyer, M.; and McCallum, A. 2019. Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Auto-Encoders. 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)*.

Dyer, C.; Kuncoro, A.; Ballesteros, M.; and Smith, N. A. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Grabisch, M.; and Roubens, M. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of Game Theory* .

Greenside, P.; Shimko, T.; Fordyce, P.; and Kundaje, A. 2018. Discovering epistatic feature interactions from neural network models of regulatory DNA sequences. *Bioinformatics* .

Guan, C.; Wang, X.; Zhang, Q.; Chen, R.; He, D.; and Xie, X. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *International Conference on Machine Learning*, 2454–2463.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* .

Htut, P. M.; Cho, K.; and Bowman, S. R. 2019. Inducing Constituency Trees through Neural Machine Translation. *arXiv preprint arXiv:1909.10056* .

Janizek, J. D.; Sturmfels, P.; and Lee, S.-I. 2020. Explaining Explanations: Axiomatic Feature Interactions for Deep Networks. *arXiv preprint arXiv:2002.04138* .

Jin, X.; Wei, Z.; Du, J.; Xue, X.; and Ren, X. 2020. Towards Hierarchical Importance Attribution: Explaining Compositional Semantics for Neural Sequence Models. In *International Conference on Learning Representations*.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kitaev, N.; Cao, S.; and Klein, D. 2019. Multilingual Constituency Parsing with Self-Attention and Pre-Training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Kitaev, N.; and Klein, D. 2018. Constituency Parsing with a Self-Attentive Encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Li, B.; Mou, L.; and Keller, F. 2019. An Imitation Learning Approach to Unsupervised Parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Li, X. L.; and Eisner, J. 2019. Specializing Word Embeddings (for Parsing) by Information Bottleneck. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Liang, R.; Li, T.; Li, L.; Wang, J.; and Zhang, Q. 2020. Knowledge Consistency between Neural Networks and Beyond. In *International Conference on Learning Representations*.

Lundberg, S. M.; Erion, G. G.; and Lee, S.-I. 2018. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888* .

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*.

Mrini, K.; Dernoncourt, F.; Bui, T.; Chang, W.; and Nakashole, N. 2019. Rethinking Self-Attention: An Interpretable Self-Attentive Encoder-Decoder Parser. *arXiv preprint arXiv:1911.03875* .

Murdoch, W. J.; Liu, P. J.; and Yu, B. 2018. Beyond Word Importance: Contextual Decomposition to Extract Interactions from LSTMs. In *International Conference on Learning Representations*.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Raganato, A.; and Tiedemann, J. 2018. An Analysis of Encoder Representations in Transformer-Based Machine Translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Reif, E.; Yuan, A.; Wattenberg, M.; Viegas, F. B.; Coenen, A.; Pearce, A.; and Kim, B. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems 32*, 8594–8603. Curran Associates, Inc.

Shapley, L. S. 1953. A value for n-person games. *Contributions to the Theory of Games* .

Shen, Y.; Lin, Z.; wei Huang, C.; and Courville, A. 2018. Neural Language Modeling by Jointly Learning Syntax and Lexicon. In *International Conference on Learning Representations*.

Shen, Y.; Tan, S.; Sordoni, A.; and Courville, A. 2019. Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks. In *International Conference on Learning Representations*.

Shi, H.; Zhou, H.; Chen, J.; and Li, L. 2018. On Tree-Based Neural Sentence Modeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Singh, C.; Murdoch, W. J.; and Yu, B. 2019. Hierarchical interpretations for neural network predictions. In *International Conference on Learning Representations*.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

Tsang, M.; Cheng, D.; and Liu, Y. 2018. Detecting Statistical Interactions from Neural Network Weights. In *International Conference on Learning Representations*.

Tsang, M.; Liu, H.; Purushotham, S.; Murali, P.; and Liu, Y. 2018. Neural Interaction Transparency (NIT): Disentangling Learned Interactions for Improved Interpretability. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Voita, E.; Sennrich, R.; and Titov, I. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Wang, X.; Tu, Z.; Wang, L.; and Shi, S. 2019. Self-Attention with Structural Position Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Wang, Y.; Lee, H.-Y.; and Chen, Y.-N. 2019. Tree Transformer: Integrating Tree Structures into Self-Attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Warstadt, A.; Singh, A.; and Bowman, S. R. 2018. Neural Network Acceptability Judgments. *arXiv preprint arXiv:1805.12471* .

Weber, R. J. 1988. Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley* .

Yogatama, D.; Blunsom, P.; Dyer, C.; Grefenstette, E.; and Ling, W. 2016. Learning to compose words into sentences with reinforcement learning. *arXiv preprint arXiv:1611.09100* .

Yogatama, D.; Miao, Y.; Melis, G.; Ling, W.; Kuncoro, A.; Dyer, C.; and Blunsom, P. 2018. Memory Architectures in Recurrent Neural Network Language Models. In *International Conference on Learning Representations*.