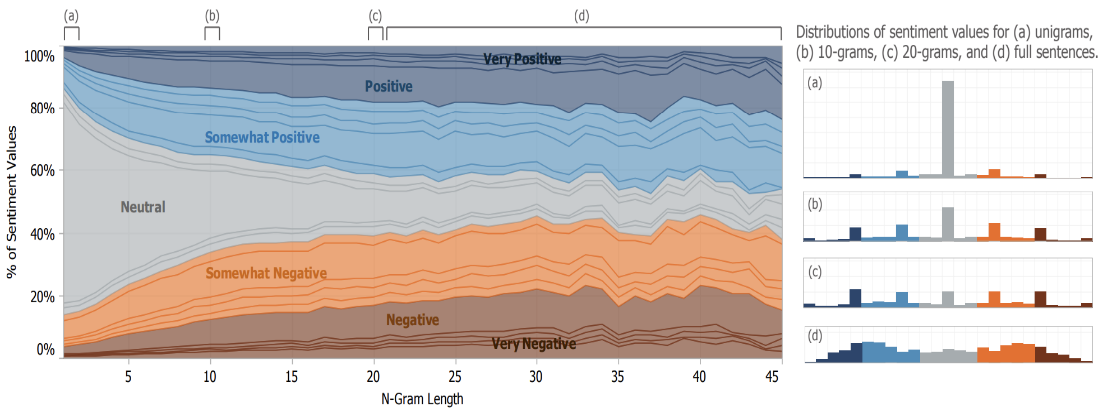
Review: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

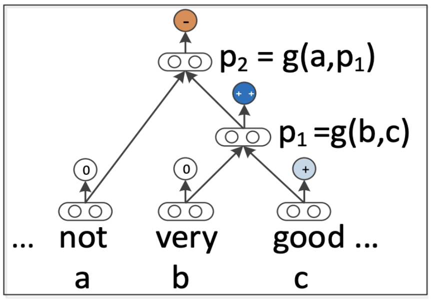
The paper aims to represent phrases and sentences as semantic vector spaces and capture their meaning through compositionality. Previous methods like bag of words don't take order of words into the account and thus fail to capture any compositional meaning. Compositional models work great as they capture the influence of one word on another word as well as the entire phrase. It is mentioned that the progress in the aforementioned direction was hindered due to the scarcity of large labeled compositional datasets. Thus a new dataset is introduced - Stanford Sentiment Treebank, which is the first corpus with fully labeled parse trees that allows for a complete analysis of the compositional effects of sentiment in language. Enabled by the new dataset, the paper uses several compositional approaches to predict positive/negative sentiments as well as more fine-grained categories of sentiments, such as Recursive Neural Networks (RNNs), Matrix-Vector Recursive Neural Networks(MV-RNNs), and finally proposes a powerful new compositional model called Recursive Neural Tensor Network(RNTNs) which outperforms all the previous recursive models and accomplishes state of the art performance.

The Stanford treebank dataset contains 10,662 movie review statements from rottentomatoes.com which after some cleaning and preprocessing is fed into the stanford parser which converts all sentences into parse trees and the resulting 215,154 phrases are labeled with the help of Amazon Mechanical Turk. This dataset is then used to predict binary positive/negative sentiments as well as fine-grained 5-class classification of sentiments(negative, somewhat negative, neutral, positive or somewhat positive). It is observed that shorter n-grams are mostly neutral and stronger sentiments are built in longer phrases.



All the models improved by 2-3 % after adding this new training data but hard negation examples were still mostly incorrect and a more powerful model was needed to solve it.

Before proposing the Recursive Neural Tensor Network model to address this problem the author discusses a couple other recursive neural networks and their limitations. All recursive neural models compute parent vectors in a bottom up fashion using a compositionality function and use node vectors as features for classifiers at that particular node. Main difference between various recursive neural models is the compositionality function that is used.



Recursive Neural Networks are the simplest member of this family and they use the following compositionality function to compute the parent vectors.

$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right), p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

f() here is tanh function which introduces non linearity necessary for efficient training of neural networks. Since the only interaction between the two vectors is the nonlinearity function, a more powerful interaction between the vector would yield better results.

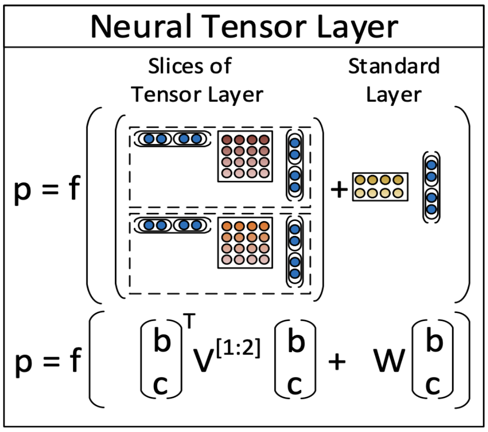
Matrix Vector Recursive Neural Networks(MV-RNNs) aims to better the results by representing every node by both vector and matrix and the compositional function works by multiplying the matrix of one node with the vector of the other and vice versa. The vector could be thought of as the sentiment score of the phrase or the word whereas the matrix is the representation of the phrase. Multiplying it in this way helps in quantizing the influence of one phrase on the other.

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right), P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right)$$

However the number of parameters in MV-RNNs depend on the vocabulary size of the dataset, and the number of parameters become too large as the vocabulary size expands.

In RNNs the vectors interact only implicitly through tanh function and ideally the interaction should be multiplicative. In MV-RNNs, the number of parameters is dependent on the vocabulary size. To solve both these limitations of RNNs and MV-RNNs, the author proposes Recursive Neural Tensor Networks (RNTNs) where the main idea is to use the same tensor based composition function across all nodes. The compositionality function of RNTNs is defined in the following way.

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]}\begin{bmatrix} b \\ c \end{bmatrix} + W\begin{bmatrix} b \\ c \end{bmatrix}\right)$$



In this function, the tensor can directly relate to input vectors. We can interpret each slice of tensor as a specific type of composition.

RNTNs are powerful in terms of capturing the structural composition and when trained on Stanford Sentiment Treebank, it achieves state of the art results for positive/negative sentiment analysis with accuracy of 85.4% beating the previous best accuracy of 80%. The accuracy of fine-grained sentiment analysis is also improved by 9.7% when compared to a bag of features baselines. It is also the only model which captures the effects of hard negation for both positive and negative phrases.

However, there are few areas where RNTNs could be improved upon. Since the RNTNs compute parent vectors in bottom up fashion it is hard to parallelize these operations. With SIMD operations getting faster everyday, it is possible for a different model to take advantage of it and achieve better results (Similar to transformers vs RNNs/LSTMs). It is mentioned in the paper that the tree structure is fixed and computing various tree structures and determining the best is not that beneficial which seems counter-intuitive as the tree structure is very influential to the vectors being computed and it seems possible to achieve vastly different output of sentiment just by changing the tree structure. The sentence level fine-grained 5 class-classification accuracy is 45.7%, only slight improvement from MV-RNNs which has accuracy of 44.4%. The model might not work if tested on sentences with incorrect grammar.

Irrespective of the above shortcomings, RNTNs are very powerful recursive models which efficiently capture compositional meaning in phrases and sentences especially the hard examples of negation in both positive and negative phrases.