**Review of Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank**

**Arvind Agrawal**

**IIT BHU(Varanasi), India**

Reading this paper was really an interesting job and it gave me insight into many things which I didn’t know prior. The main contributions of this paper as inferred by me are:-

**Sentiment TreeBank:-**

Before this dataset was introduced models were not able to improve beyond a certain level with the existing dataset because of the complexity in determining the sentiment in shorter phrases. In longer phrases also the models relied on some stronger words which tend to reflect the sentiment of the whole text and did not considered the order of the words in the text. Moreover shorter phrases were also not properly labelled so that the models can be trained on them.

Sentiment TreeBank prepared via filtering the reviews of rottentomatoes.com and used Amazon Mechanical Turk to not only provide a proper sentiment labelling of all the phrases on the basis of order of texts but also provide the labelling in 25 different possible classes. In fine-grained classification the labels were reduced to 5. The result of any task can be improved by either improving the model in some sense or the other, improving the dataset or computation power. In this case Sentiment TreeBank improved the results in all the previous models as reported by them in their paper which proved that this newly constructed dataset is properly labelled and can be used of high advantage.

However what I feel is because it filters the data with all the emojis, emoticons, hashtags etc. The dataset cannot comply with the reviews that are coming nowadays which includes all of these and can also serve as a deciding factor in sentiment analysis sometimes.

**Recursive Neural Tensor Network:-**

All the previous models were not able to increase beyond a certain limit because either they (RNN) were just combining the word/phrase information additively with the help of a non-linear composition function or they(MV-RNN) had a large number of parameters to train which depended on the size of the vocabulary which became difficult to train.

Recursive Neural Tensor Network solved both of these problems by introducing a composition function with fixed number of parameters and which also combines it’s components in some sort of multiplicative sense other than the additive sense. It’s each slice V[i] in its vectorized notation is believed to capture different type of composition and also compose aggregate meaning from smaller constituents more accurately as compared to other models due to this vectorized notation. Training the parameters with the help of KL-Divergence loss due to the actual labels representing a distribution in some way due to its long classification range was also really interesting as normally classifiers were trained with the help of loss functions specific to classification like cross entropy loss or hinge loss etc. It was also able to capture the difference between the tone/sentiment of sentences before and after the use of word **but.** The analysis on different kind of datasets which included datasets with many classes, binary, self-made datasets by negating positive and negative sentences showed the accuracy with which the model is able to capture the minute details which other models are not able to.

The combination of both Sentiment TreeBank and Recursive Neural tensor Network gave extraordinary results as it combined the advantages of both an improved training and testing dataset and also the improved model which was able to devise a composition function not so hard to train and which also captured meaning from smaller constituents and considered the order of words rather than just relying on some strongly influencing words.

Some disadvantages that I think Recursive neural tensor network may face is that they depend on well-performing parsers to provide the topological structure, they also require proper labelling of all the phrases to avoid vanishing gradient problem and which may not be available in many other language datasets.