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

Mitodru Niyogi
Heidelberg University / Nuremberg, Germany
niyogi@cl.uni-heidelberg.de

Abstract

Socher et al. [2013] discussed various compositional methods combining words and phrases (n-gram) for both binary (positive/negative) and 5 major classes (very negative, negative, neutral, positive, very positive) sentiment classification of words, phrases and whole sentence in a bottom-up fashion. The main contribution of the author was to propose a neural compositional model, Recursive Neural Tensor Network (RNTN) trained on a newly introduced "Stanford Sentiment Treebank" which is a parse tree based dataset with fine-grained sentiment labels. This new model architecture, RTNN achieved state-of-the-art performance of 80.7% accuracy on fine-grained sentiment prediction across all phrases and can capture the constructive conjuction and negation in sentences.

1 Introduction

As semantic vector spaces for single words are unable to capture the meaning of longer phrases properly and due to lack of presence of large and labeled compositionality resources, [Socher et al., 2013] introduced the Stanford Sentiment Treebank dataset and a powerful Recursive Neural Tensor Network(RNTN) which can accurately predict the compositional semantic effects present in a corpus.

2 Dataset

The corpus was created by parsing 11,855 sentences of movie reviews excerpt corpus with Stanford Parser which resulted in 215,154 phrases. These phrases were then randomly sampled and labeled into 25 different values using Amazon Mechanical Turk. Analyzing the treebank, the authors observed that that shorter phrases have neutral sentiments whereas longer phrases showed more polarized stronger sentiments. It has also been shown that 5 sentiment classes are enough to capture the major variation in the data based on the annotators' grading.

3 RNTN: Recursive Neural Tensor Network

The authors discussed about how to compute compositional vector representations for phrases of variable length and syntactic type. The n-gram input is parsed into a binary tree and each node is represented as a word vector represented by a $d - dimensional$ vector . The word embedding matrix L is trained jointly with the compositional models. The word vectors are used as feature input to $softmax$ classifier for each 1-gram vector to predict sentiment class probability of that node. Since the input is represented as a tree, the word vectors of child nodes are computed first and then merged together to compute the parent vector representation in a bottom-up fashion using $\tanh$ compositional vector. The authors further discussed the limitations of the previous methods such as matrix-vector Recursive Neural Network (MV-RNN) [Socher et al., 2012] due to large number of parameters as every word and longer phrase are represented in a parse tree as both vector and a matrix respectively. Word vectors and matrix are parameters which are learned during training. The proposed RTNN model by authors reduces the large number of parameters in MV-RNN by using a single powerful same tensor-based composition function across all nodes to have greater explicit interactions between the input word vectors as tensors can directly relate input vectors unlike the standard RNN. The compositionality function has the structure of a feedforward neural network layer, possibly with additions such as a tensor layer. The parameters for the compositionality function and for the vectors themselves are trained using tensor backpropagation through structure.

4 Experimental Results & Analysis

The paper offered several important insights and observations:

- For all models, cross-validation was performed over word vectors, learning rate and minibatch size for AdaGrad.

- The authors reported that performance decreased for higher batch size and word vector size. However, the optimal performance for all models was reported at batch size between 20 and 30, word vector size between 25 and 35. This confirms that RNTN model's performance enhancement is just not because of presence of higher parameters as MV-RNN has largest number of parameters.

- The authors reported that recursive models shows significant 5% drop in performance in absence of non-linearity.

- Proposed models were compared with standard Naive Bayes(NB), SVM, BiNB (NB with bigram features), VecAvg(averaging word vectors but ignores word order). On fine-grained classification for all phrases (at all node levels of the parse trees) RNTN achieved best performance, followed by MV-RNN, RNN and other models.

- For fine-grained sentiment classification for all phrases, the authors cited that for shorter phrases where negation and composition are important factors, recursive models worked pretty well whereas bag of features baselines like NB, SVM performed well only for longer sentences. The authors also showed that RNTN's accuracy upper bounds other models at most n-gram lengths. For binary classification at sentence level, RNTN achieved state-of-the-art accuracy from 82.9% to 85.4%.

- RNTN was efficient in capturing the effect of Contrastive Conjunction ('but') on overall sentiment of the sentence and performed better compared to MV-RNN by 10.8%, RNN by 13.8%(36), and biNB by 51.85% respectively.

- RNTN was able to understand the effect of negation in both positive and negative sentences. It achieved the highest accuracy for negating the positive sentences and also learnt the negation constructs beyond simple negation rules by increasing the degree of non-negative sentiment in a sentence for negation of negative sentence use-cases.

5 Discussion

5.1 Advantages of Treebank

The advantage of treebank is that powerful models have been built to predict sentiments of shorter sentences and classify difficult negation examples which were not attainable by the earlier bag of words based classifiers on traditional datasets. The binary sentiment classification task has crossed accuracy of 80% for the first time.

5.2 Advantages of Recursive Neural Networks

The ability to handle the hierarchical data representation and learn hierarchical data patterns.

Another advantage is that the long-term dependency becomes shorter and easier to interpret because of path compression between the first and last input elements in the parsed tree, when there is $\mathcal{O}(n)$ input words; a recursive neural network returns a tree of $\mathcal{O}(n \log n)$ height.

5.2.1 Disadvantages of RNTN

Due to the tree structure of RNTNs, the tensor backpropagation is time consuming for longer sentences as each node performs a softmax classifier operation. The tree structure also introduces an inductive bias to the model as it assumes that the input data follows a tree hierarchical representation. However, the network may fail to learn patterns if the input data is not hierarchical representation such as cases of poor grammatically constructed tweets and dialogues in chatbots.

The 'one size fits all' compositional mechanism used by the authors does not distinguish between content words and information routing words such as logical words and pronouns.

Another disadvantage is parsing of the input data can be slow and and ambiguous. There can be several parse trees for a single sentence representation.

Moreover, labeling the training data manually for recursive neural networks is labor intensive and more time-consuming than constructing recurrent neural networks by assigning a label to a sequence.

5.3 What else can be done?

- Use of Adam optimizer instead of AdaGrad for loss optimization
- Use of $relu$ or $leakyrelu$ instead of $\tanh$ for nonlinearity.
- The authors could have added word clouds for most positive and most negative n-grams for all models instead of a tabular representation.
- Use of pretrained word embeddings such as Glove, fasttext, word2vec instead of learning the word vector embeddings as parameters during training and observe the change in the performance of RNTN model.

6 Conclusions

The proposed RNTN model on newly introduced Sentiment TreeBank dataset has shown its ability to capture the structural composition of words and phrases in a sentence. It has also learnt to detect the impact of negation construct in complex sentences having mix of positive, negative, and neutral sentiments whereas BiNB and RNN performed poorly. The RNTN achieved 80.7% accuracy on fine-grained sentiment prediction across all phrases and a 5.4% improvement for binary single sentence sentiment classification over the previous state-of-the-art model.

References

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL `https://aclanthology.org/D12-1110`.

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. URL `https://aclanthology.org/D13-1170`.