RNN for Sentiment Analysis: Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Borui(Athena) Ye

University of Waterloo

borui.ye@uwaterloo.ca

July 15, 2015

Overview

Paper Information

Richard Socher, Alex Perelygin, Jean Y.Wu, Jason Chuang, Christopher D. Manning, Andrew Y. 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, pp 1631-1642. 2013.

Introduction

Semantic Compositionality : to calculate in a systematic way the polarity values of larger syntactic constituents as some function of the polarities of their subconstituents[1].

Corpus (Sentiment Treebank)

- 11,855 sentences based on extracted from movie reviews [2]
- 215,154 phrases parsed from sentences using Stanford Parser[3], each annotated by 3 annotators.

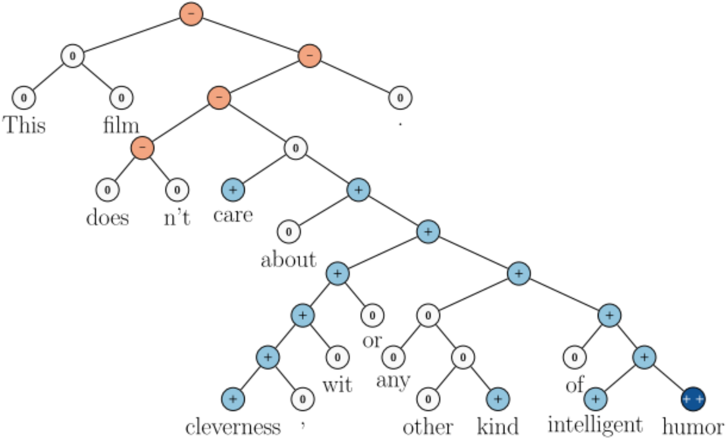
---

[1] K. Moilanen and S. Pulman, "Sentiment composition," in *Proceedings of RANLP*, vol. 7, 2007, pp. 378–382.

[2] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2005, pp. 115–124.

[3] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, Association for Computational Linguistics, 2003, pp. 423–430.

Introduction (Cont.)



Introduction (Cont.)

Experiments

1. Fine-grained Sentiment For All Phrases
2. Full Sentence Binary Sentiment
3. Model Analysis: Contrastive Conjunction
4. Model Analysis: High Level Negation
   - Negating Positive Sentences
   - Negating Negative Sentences
5. Model Analysis: Most Positive and Negative Phrases

Related Work

Semantic Vector Spaces. Distributed similarities of single words. But often fail to distinguish antonyms.

- Co-occurence of a word and its context [4].
- How often a word appears in a certan syntactic context [5].

Compositionality in Vector Spaces. Most of them capture two word compositions.

- Word vector addition, multiplication, etc. [6]
- Represent phrases as matrixes and define composition method as matrix multiplication. [7]

---

[4] P. D. Turney, P. Pantel, *et al.*, "From frequency to meaning: Vector space models of semantics," *Journal of artificial intelligence research*, vol. 37, no. 1, pp. 141–188, 2010.

[5] S. Padó and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.

[6] J. Mitchell and M. Lapata, "Composition in distributional models of semantics," *Cognitive science*, vol. 34, no. 8, pp. 1388–1429, 2010.

[7] E. Grefenstette and M. Sadrzadeh, "Experimental support for a categorical compositional distributional model of meaning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2011, pp. 1394–1404.

Related Work (Cont.)

Sentiment Analysis.

- Bag-of-words representations [8].
- Extracting features or polarity shifting rules on syntactic structures [9]

Recursive Neural Models Will be covered later.

[8] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[9] L. Polanyi and A. Zaenen, "Contextual valence shifters," in *Computing attitude and affect in text: Theory and applications*, 2006, pp. 1–10.

Stanford Sentiment Treebank

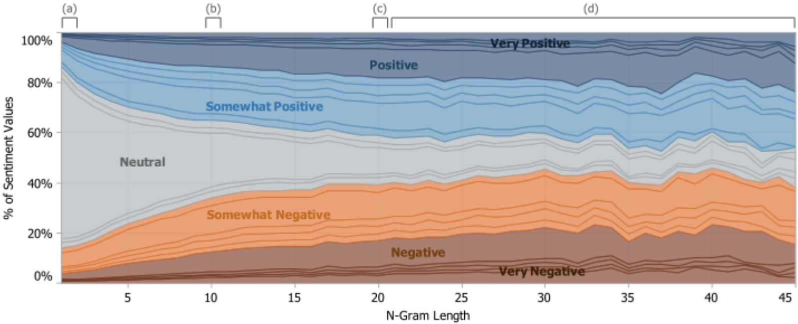
Data retrieval and processing:

- Get movie review excerpts from the `rottentomatoes.com`, which includes 10,662 sentences, half positive, half negative.
- Parse sentences using the Stanford Parser.
- Using Amazon Mechanical Turk to label the resulting 215,154 phrases.

Statistics

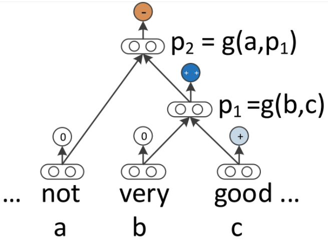
Findings:

1. Most of the short n-grams are neural;
2. Longer n-grams are evenly distributed;
3. Extreme sentiment degrees rarely happen.



Recursive Neural Models

Tri-gram example of bottom up fashion:



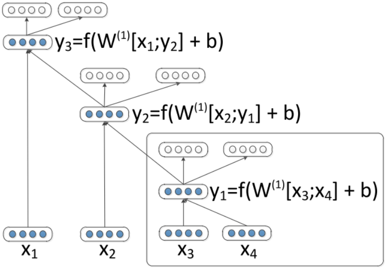
Initialization

- Initialize each word vector using uniform distribution: $U(-r, r)$, where $r = 0.0001$.
- Stack word vectors into matrix $L \in \mathbb{R}^{d \times |V|}$, where d is vector dimension, $|V|$ is vocabulary size.

RNN: Recursive Neural Network[10]

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

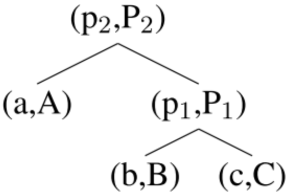
where $f = \tanh, W \in \mathbb{R}^{d\times 2d}$



[10] R. Socher, J. Pennington, E. H. Huang, *et al.*, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2011, pp. 151–161.

MV-RNN: Matrix-Vector RNN[11]

Main Idea: represent every node in the parse tree both as a vector and a matrix.



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

where $W, W_M \in \mathbb{R}^{d\times 2d}$

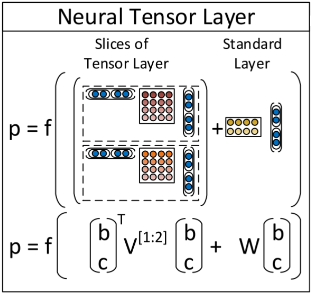
[11] R. Socher, B. Huval, C. D. Manning, *et al.*, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Association for Computational Linguistics, 2012, pp. 1201–1211.

MV-RNN: Matrix-Vector RNN (Cont.)

Problem: size of parameters becomes very large and depends on the size of the vocabulary.
Solution: use a simple powerful composition function with a fixed number of parameters.

RNTN: Recursive Neural Tensor Network

Main Idea: use the same, tensor-based compostition function for all nodes.



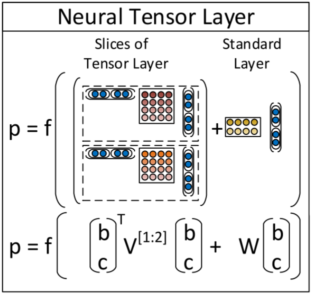
Definition

- $h \in \mathbb{R}^d$: output of the tensor product
- $V^{[1:d]} \in \mathbb{R}^{2d \times 2d \times d}$: tensor that defines multiple bilinear forms.
- $V^{[i]} \in \mathbb{R}^{2d \times 2d}$: each slice of $V^{[1:d]}$.

$$h = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix}; h_i = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[i]} \begin{bmatrix} b \\ c \end{bmatrix}$$

RNTN: Recursive Neural Tensor Network (Cont.)

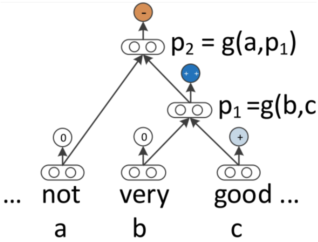
Intuitively, we can interpret each slice of the tensor as capturing a specific type of composition.



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

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

Tensor Backprop through Structure



Each node is assigned a label via:

$$y^a = \mathrm{softmax}(W_s a)$$

where $W_s \in \mathbb{R}^{5 \times d}$ is the sentiment classification matrix.

Tensor Backprop through Structure (Cont.)

Goal: minimize the KL-divergence between the predicted distribution $y^i \in \mathbb{R}^{C\times 1}$ at node i and the target distribution $t^i \in \mathbb{R}^{C\times 1}$. The error function of a sentence is:

$$E(\theta) = \sum_i \sum_j t_j^i \log y_j^i + \lambda \|\theta\|^2$$

where $\theta = (V, W, W_s, L)$.

Experiments

Two kinds of experiment:

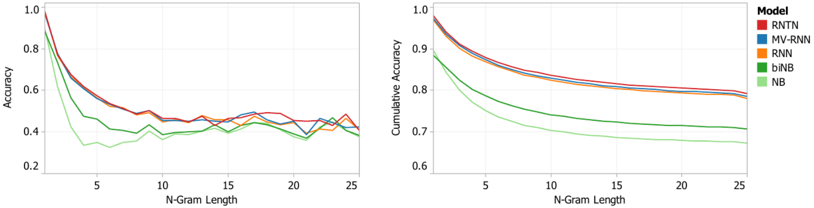
- Large quantitative evaluations on the test set.
- Linguistic phenomena: contrastive conjunction and negation.

Baselines:

- Bag-of-words features + Naive Bayes (NB)
- Bag-of-words features + SVM (SVM)
- Bag-of-bigram features + Naive Bayes (BiNB)
- Averages of neural word vectors (VecAvg)
- RNN
- MV-RNN

Sentiment Classification

1. Exp. 1: Fine-grained Sentiment For All Phrases
2. Exp. 2: Full Sentence Binary Sentiment



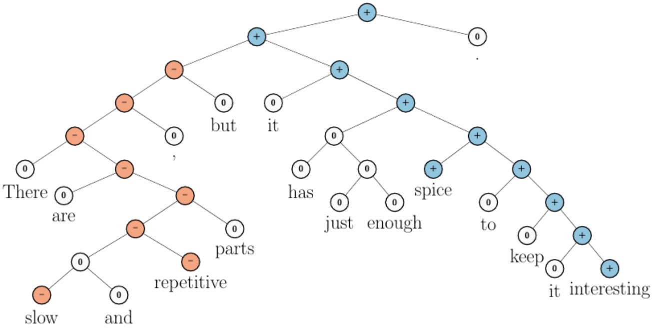
Accuracy

- Recursive models work better on shorter grams.
- RNTN upper bounds other models at most n-gram lengths.

Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Exp. 3: Model Analysis: Contrastive Conjunction

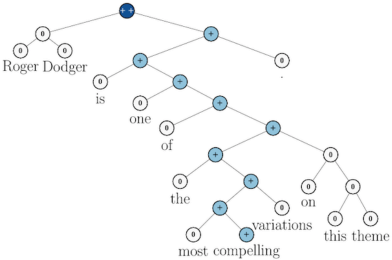
X but **Y Structure** : two phrases, X and Y, connect by “but”.



Experiment result: the test set includes 131 cases (subset of the original test set), RNTN achieve a accuracy of 41%, compared to MV-RNN (37), RNN (36) and biNB(27).

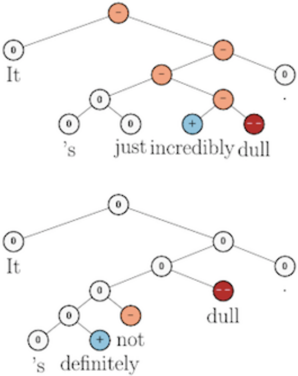
Exp. 4: Model Analysis: High Level Negation

Set 1: Negating Positive Sentences



Exp. 4: Model Analysis: High Level Negation (Cont.)

Set 2: Negating Negative Sentences



Exp. 4: Model Analysis: High Level Negation (Cont.)

Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

Thank you!