Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Oliver Leicht, Sebastian Lunz

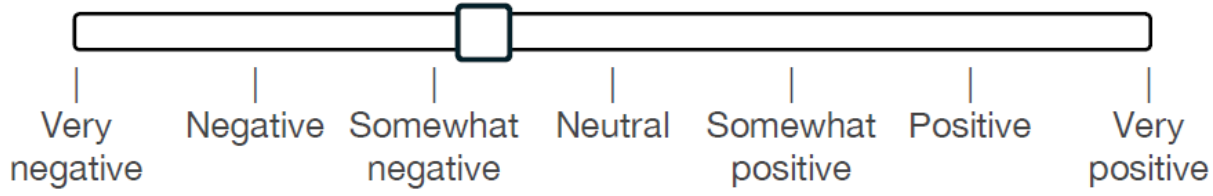
University of Cambridge

14/11/2017

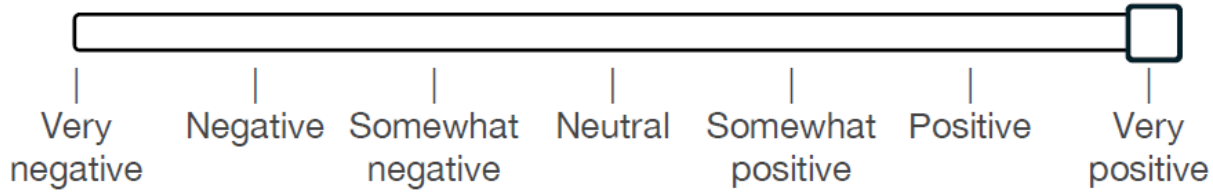
Aims

Classify sentiment class of given sentence or short phrase

nerdy folks



phenomenal fantasy best sellers



Difficulty: Context information highly relevant

- This film doesn't care about cleverness
- There are slow parts, but it has enough spice to keep it interesting

Word embedding

- Given dictionary V - very high dimensional when using natural embedding into $\mathbb{R}^n$
- Relevant information for sentiment prediction can be represented in lower dimensional space
- Require map $L : \mathbb{R}^{|V|} \to \mathbb{R}^d$ that preserves relevant sentiment information
- Solution: Learn linear map L

Sentiment classification

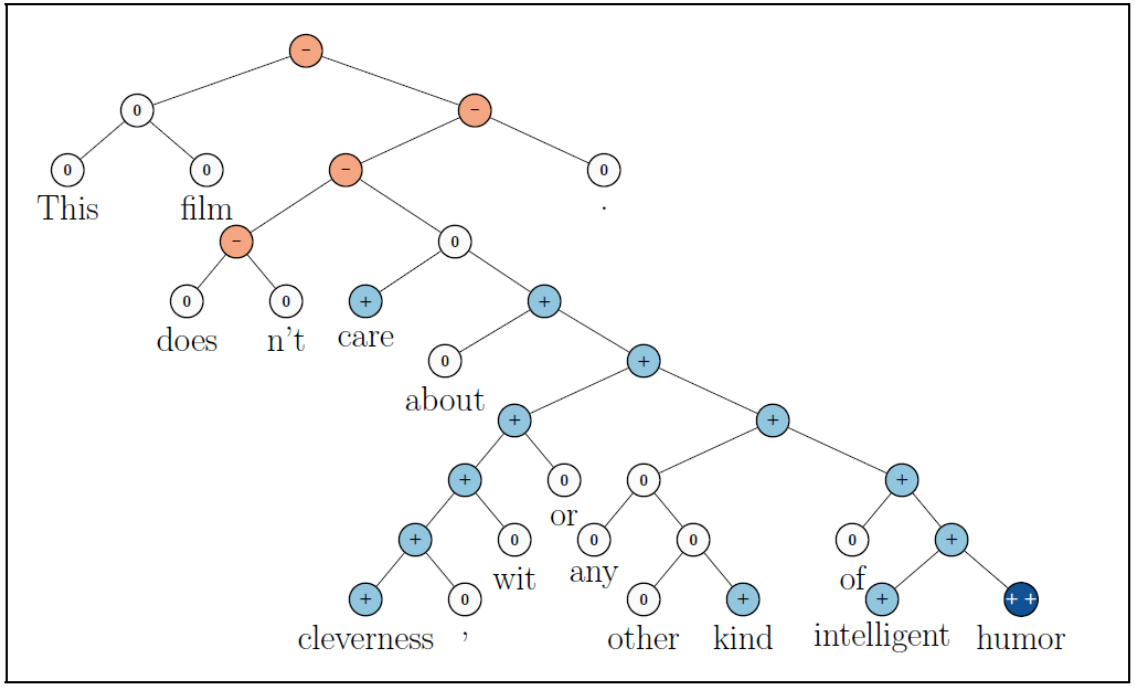
- Given single word representation a, need way to classify sentiment content of a
- Solution: Learn linear map $W_S : \mathbb{R}^d \rightarrow \mathbb{R}^5$.
- For representation a, distribution over sentiment classes given by $softmax(W_s(a))$

'marvelous' $\rightarrow^L$ abstract representation $a \in \mathbb{R}^d$

$\rightarrow^{W_S}$ 95% very positive, 5% positive

Phrases and Parse Trees

Idea: Put Sentence into tree representation.



Words on leafs, every internal knot has exactly two children.

Sentiments on Parse Trees

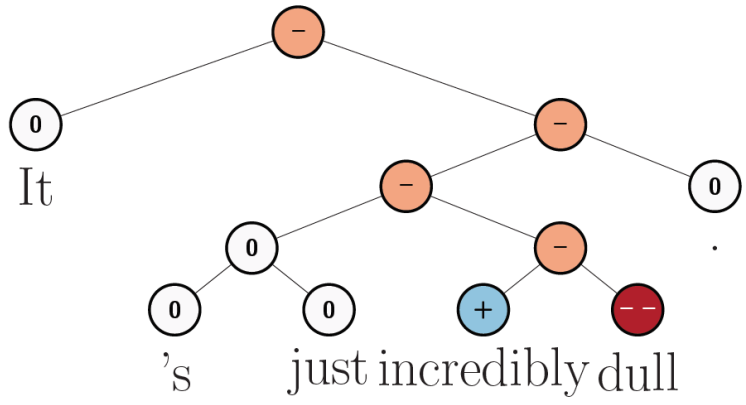
- Start with knot p who has two labeled children with word representations a and b.
- Compute value p via

$$p = \Phi_\Theta(a, b)$$

  for some *learned* map $\Phi : \mathbb{R}^{2d} \to \mathbb{R}^d$
- Importance of word context and connections built into network architecture.

Loss quantification on labeled tree

Using learned operators L, W_S and Φ_Θ obtain sentiment labeling of parse tree



Compare to correct labeling t_j^i, compute loss (cross - entropy)

$$\sum_{i \in \{knots\}} \sum_{j=1}^{5} t_j^i \cdot \log y_j^i + \lambda \|\Theta\|^2$$

with Θ list of all weights.

Recursive Neural Networks (RNNs)

Use the map

$$\Phi_\Theta(a, b) = f\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

- f is an element-wise non-linearity, the paper chose $f = \tanh$
- $W \in \mathbb{R}^{d \times 2d}$, so $|\Theta| = 2d^2$
- interactions between nodes only indirectly through f

Matrix-Vector RNN

$$\tilde{\Phi}_\Theta(a, A, b, B) = \left(f\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix} \right), f\left(W_M \begin{bmatrix} A \\ B \end{bmatrix} \right) \right)$$

- Idea: represent each mode in the parse tree with vector *and* matrix to create explicit interactions
- augment map to $\tilde{\phi} : \mathbb{R}^{2(d+d^2)} \to \mathbb{R}^{d+d^2}$
- additional $|V|d^2$ parameters on word level (recall: V is huge!)
- $W, W_M \in \mathbb{R}^{2d \times d}$, so $|\Theta| = 2 \times 2d^2$

Recursive Neural Tensor Network

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

- restrict to manageable number of parameters keeping the explicit non-linearities
- $W \in \mathbb{R}^{2d \times d}$, $V^{[1:d]} \in \mathbb{R}^{2d \times 2d \times d}$, so $|\Theta| = 2d^2 + 4d^3$
- instead of modeling the interactions with V, one could use another neural network. However, the paper reports optimization problems due to the increased complexity needed to achieve similar interactions as for the explicit ones in the RNTN.

Experiments: overview

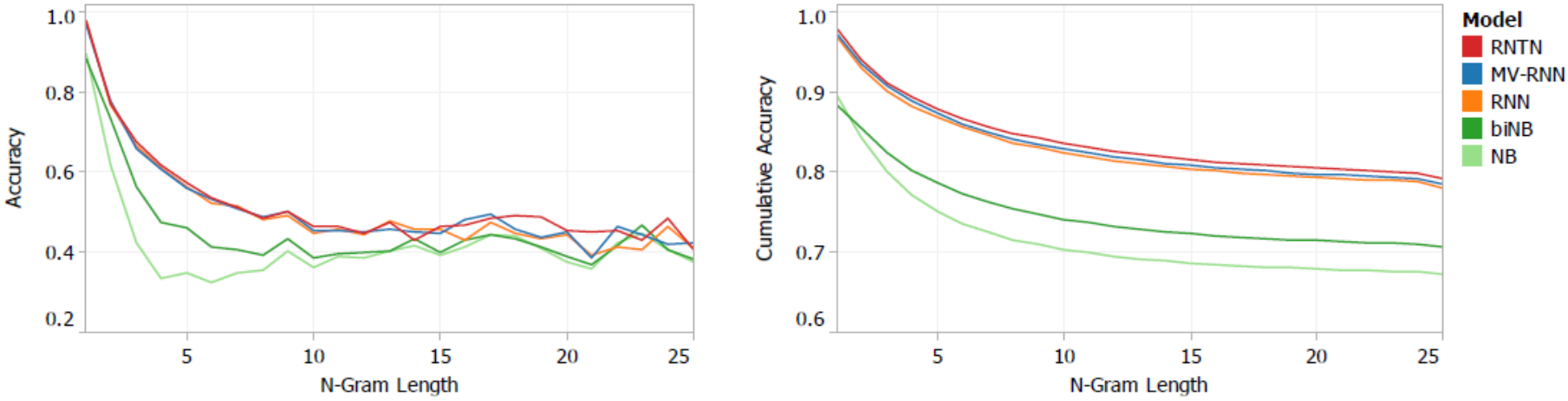
- optimal performance at word vector sizes between 25-35
- optimal batch size between 20-30
- data set split into train, development and test sets with 8544, 1101 and 2210 sentences respectively
- when assessing performance on positive/negative sentences only, the sets have 6920/872/1821 elements

Sentiment analysis I

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**

- RNTN performs always best
- prediction in long sentences for multiple classes is increasingly difficult

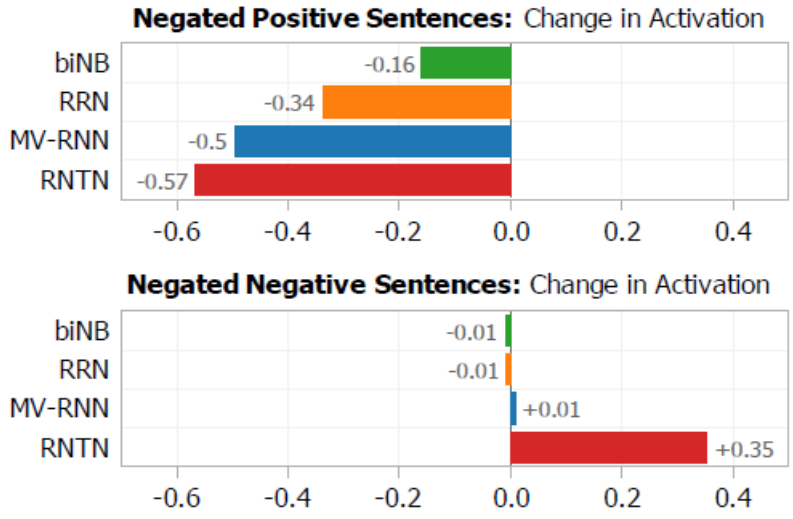
Sentiment analysis II



- prediction for large N/long sentences difficult
- RNNs able to model some structure in sentences
- the three RNN descendants perform similarly

Model Analysis

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**



- negated positive is measured as correct sentiment inversions
- negated negative correct classified if positive activations increase
- RNTN outperforms all other models significantly

Discussion

- are there better loss/performance metrics for overall classification as the dramatical differences of the different RNNs regarding negations are not reflected in the sentiment results
- why cross entropy loss?
- how sensitive is the performance to the chosen tree (a comparison for the cases when different trees are available would have been interesting)
- Are there new approaches increasing the accuracy for long sentences?