Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Richard Socher et al., 2013

Reporter: Dongyao Hu (胡东瑶)

Stanford Sentiment Treebank

- Traditional **bag of words classifiers**
  - Work well in longer documents.
  - Relying on a few words with **strong sentiment** like *'awesome'*
  - Accuracies even for **binary positive/negative classification** for single sentences **< 80%** for several years.
  - Ignoring **word order**
  - **hard examples of negation**

Stanford Sentiment Treebank

- **Homepage**: http://nlp.stanford.edu/sentiment
- **Complete analysis** of the **compositional effects of sentiment**.

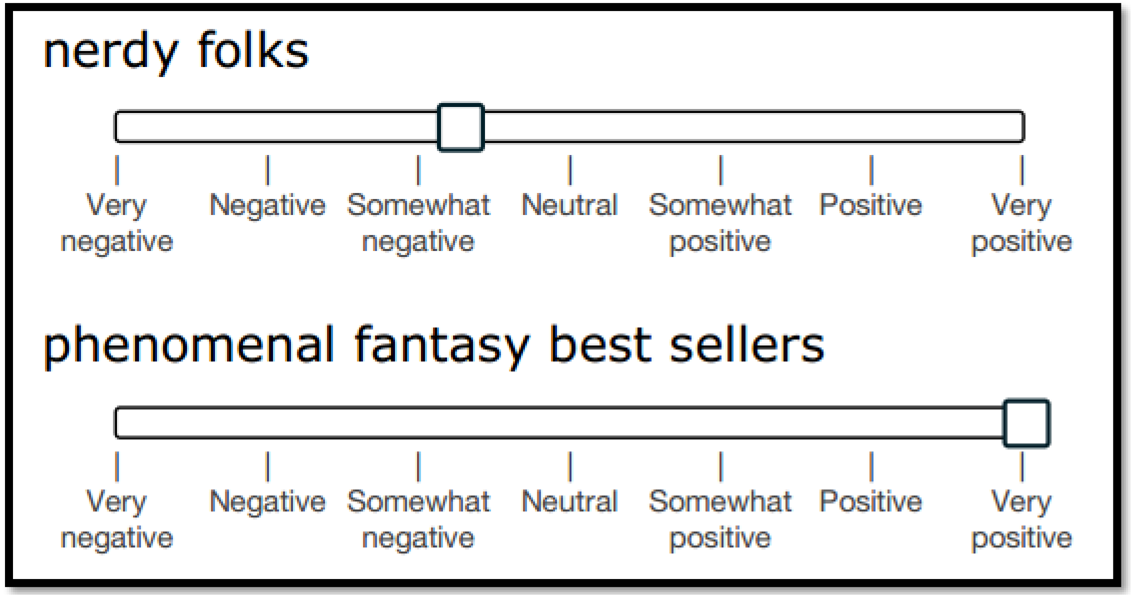
rottentomatoes.com (烂番茄)

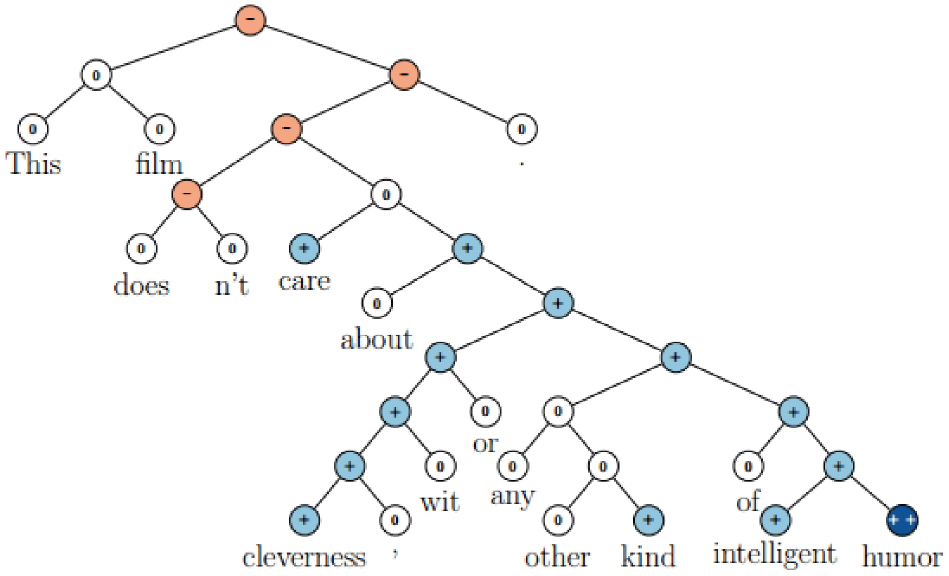
- **11,855 single sentences** extracted from **movie reviews**.
- **215,154 unique phrases** parsed with the **Stanford parser**.
- The **first corpus** with **fully labeled** (by **Amazon Mechanical Turk**).
- each **annotated by 3 human** judges.

Stanford Sentiment Treebank

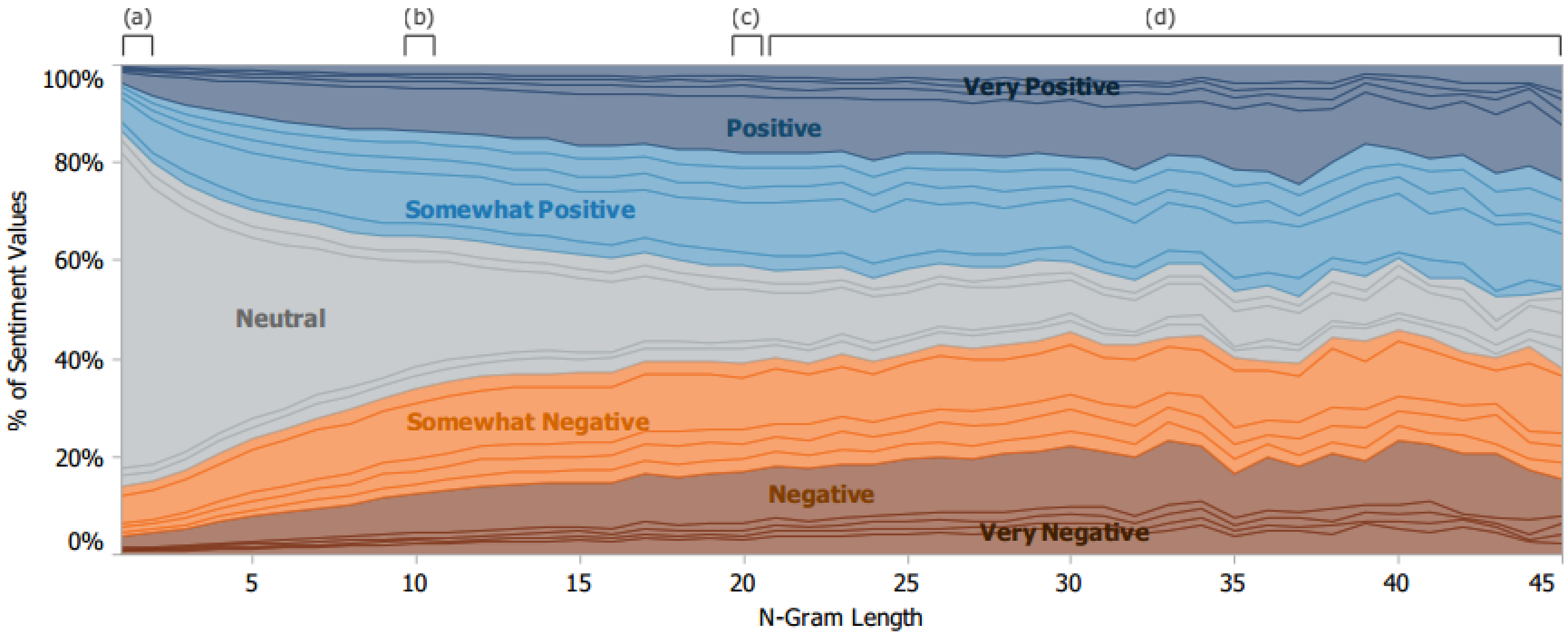
Amazon Mechanical Turk

The labeling interface





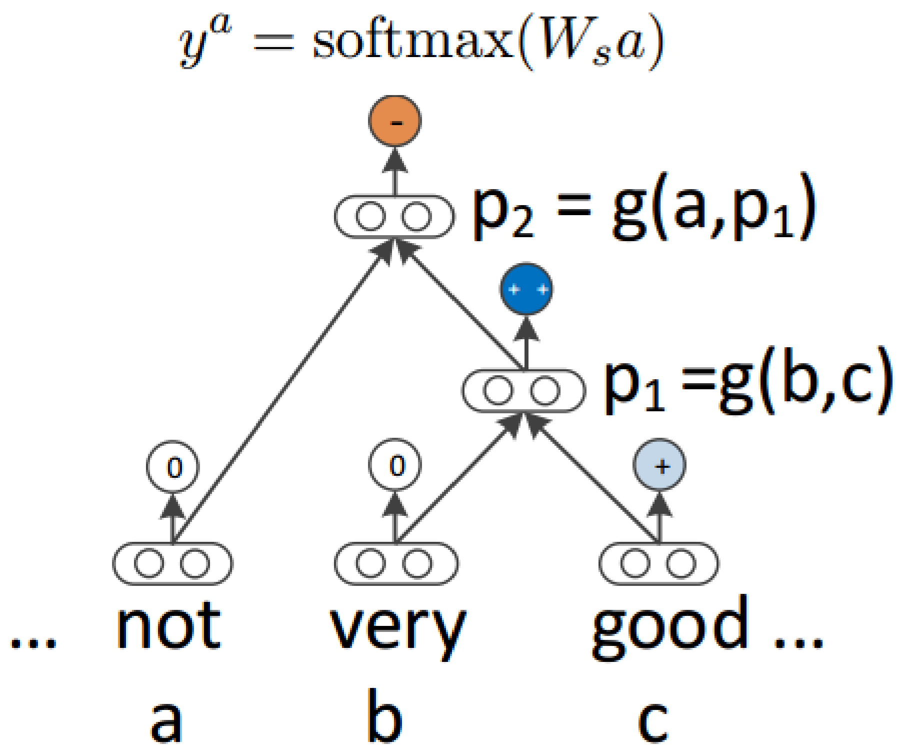
Stanford Sentiment Treebank



- **Longer phrases – stronger sentiment**, vice versa.
- Extreme options were **rarely used** by AMT annotators.
- **5-class** labels are enough.

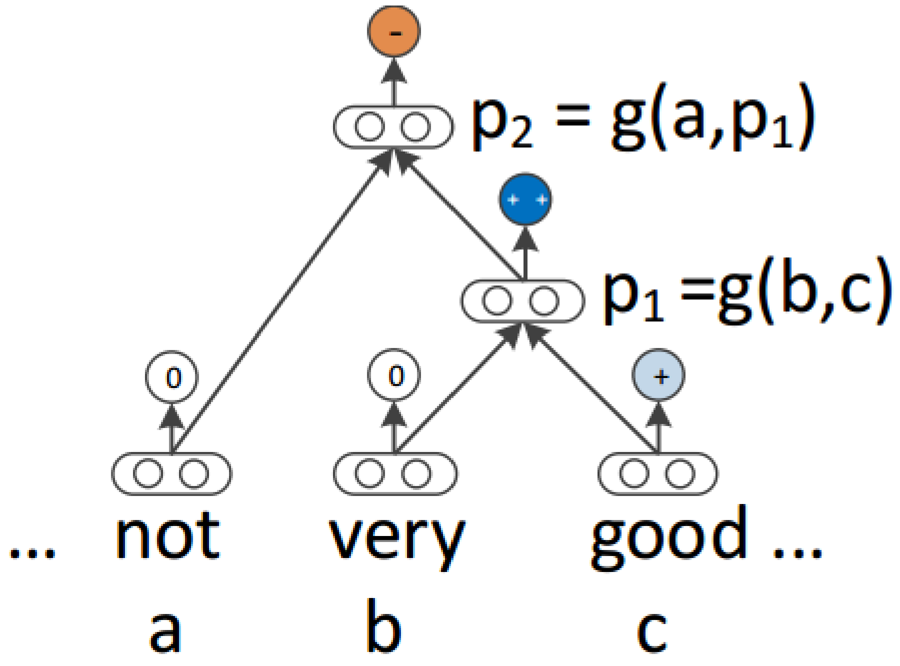
Recursive Neural Models

- Each word is represented as a **d-dimensional vector**.
- **tri-gram** as example.
- Initialize~ **uniform distribution**.
- **Compositionality functions** *g*.
- Classification into five classes using **softmax**, like [0,0,1,0,0].



RNN: Recursive Neural Network

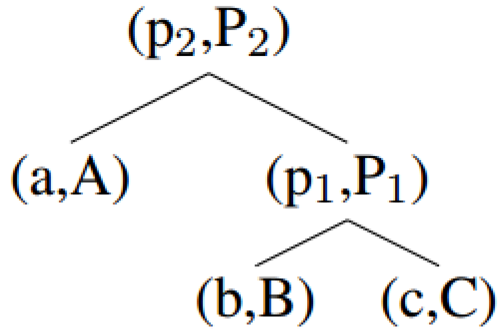
- *f = tanh*
- we omit the bias for simplicity.
- **Parent vectors** must be of the **same dimensionality** to be **recursively compatible** and be used as input to the next composition.



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

MV-RNN: Matrix-Vector RNN

- Represent word / longer phrase as *Vector & Matrix*.



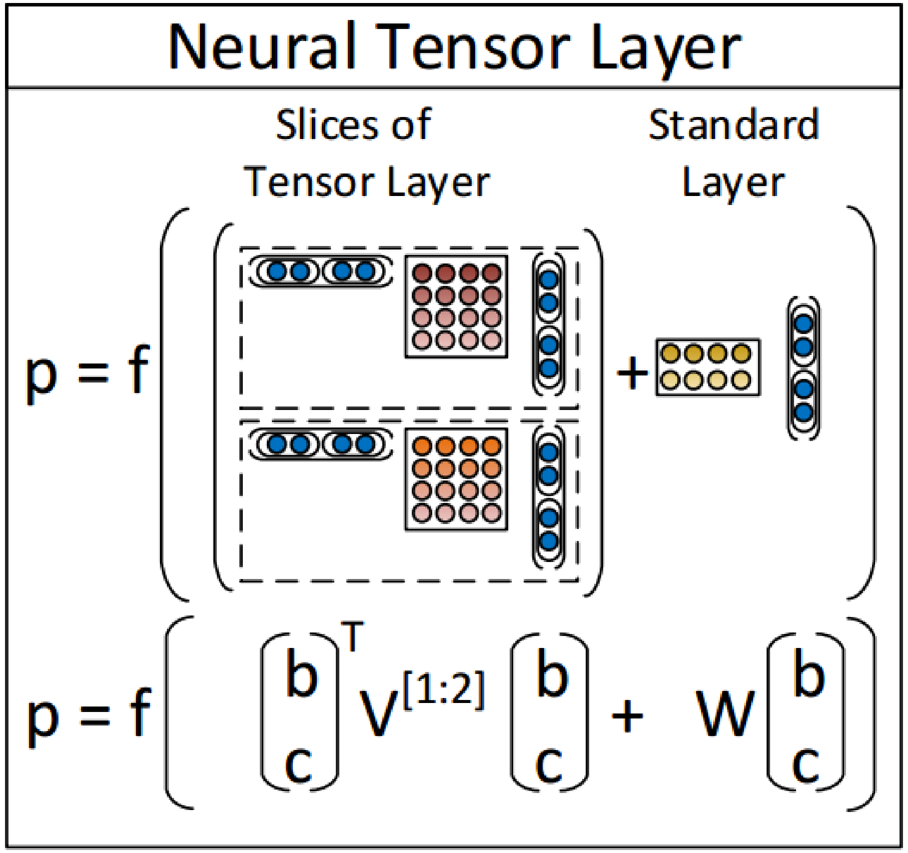
- When two constituents (成分) are combined the **matrix of one** is **multiplied with** the **vector of the other** and **vice versa.**

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right)$$

$$P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right)$$

RNTN: Recursive Neural Tensor Network

- MV-RNN: the **number of parameters becomes very large** and **depends on the size of the vocabulary**.
- The main idea of RNTN is to use the **same**, **tensor-based composition function** for all nodes.
- Refer to the **Lecture Slide**.



How to train RNTN
(backpropagation algorithm omitted)

- 5 classes with 0-1 encoding.

- Minimize the **cross-entropy error** (equivalent to minimizing KLD)

- For this **nonconvex optimization** we use **AdaGrad**.

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

Cross-entropy error cost func.

Regulation

Experiments

Fine-grained Sentiment (细粒度情感) For All Phrases
Full Sentence Binary Sentiment

- The **previous state of the art** was **below 80%** (Socher et al., 2012)
- Pushes by **5.4%** for P/N classification.

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**

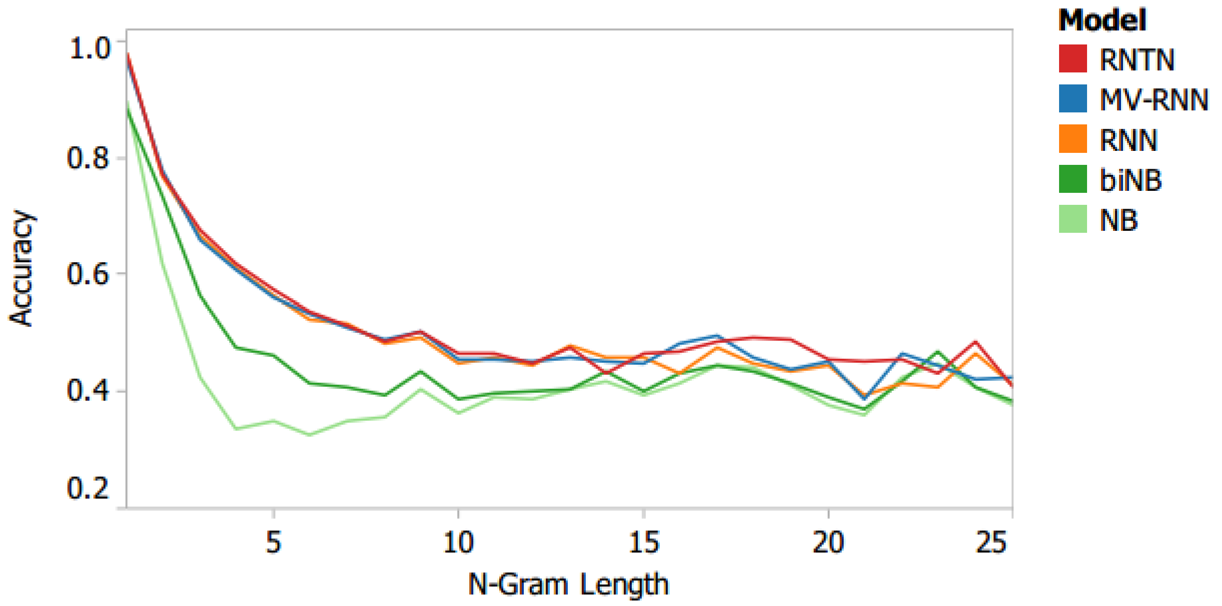
Fine-grained: 细粒度，这里指 5 级细粒度（--,-,0,+,++）
P/N: 正/负，这里指简单的二分类（-,+）
All: All phrases
Root: Full sentence
BiNB: Naive Bayes with bag of bigram
VecAvg: 简单地把词向量平均

Experiments

Fine-grained Sentiment (细粒度情感) For All Phrases

Full Sentence Binary Sentiment

- Bag of features **baselines** perform well **only with longer sentences.**
- The **recursive models** work very well on **shorter phrases**.



Experiments

Contrastive Conjunction (转折、对比类连词)

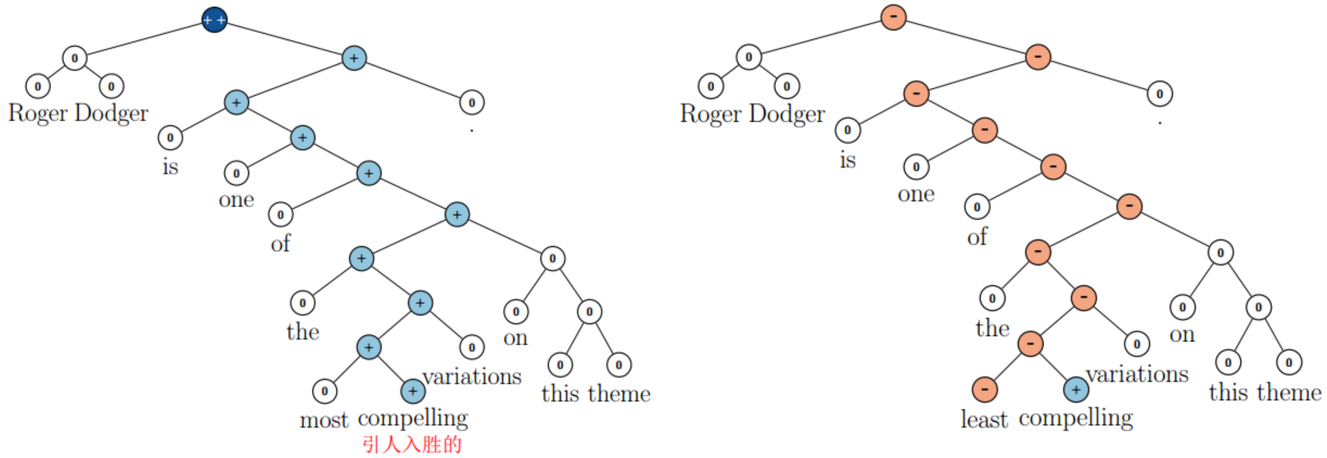
- '**X but Y** ' structure

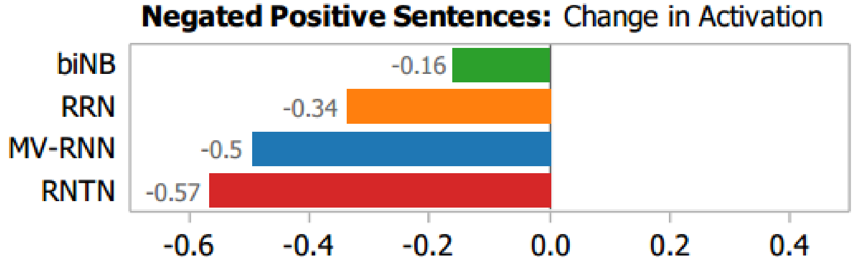
- The example is counted as correct, if the classifications for **both phrases X and Y are correct**.

- For the resulting **131** cases, the **RNTN obtains an accuracy of 41%** compared to **MV-RNN (37)**, **RNN (36)** and **biNB (27)**

Experiments

Negating Positive Sentences (否定肯定句)



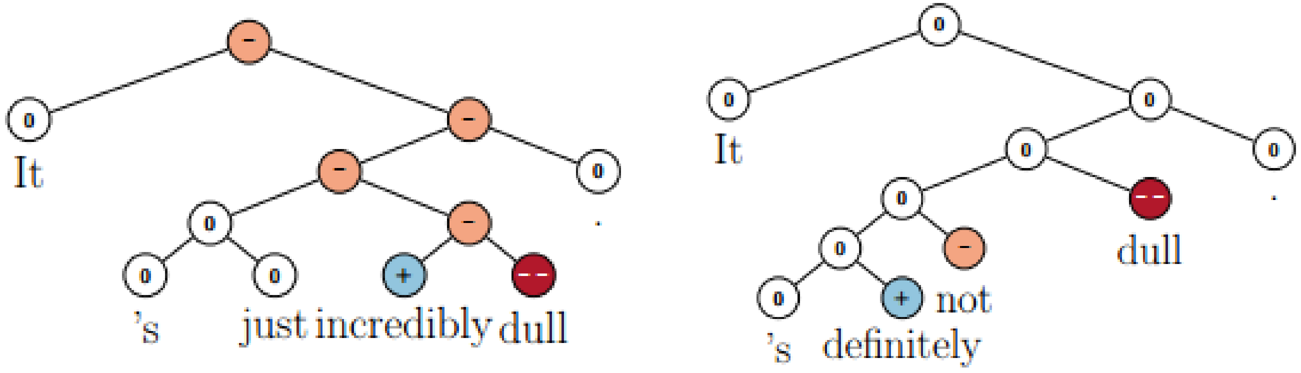


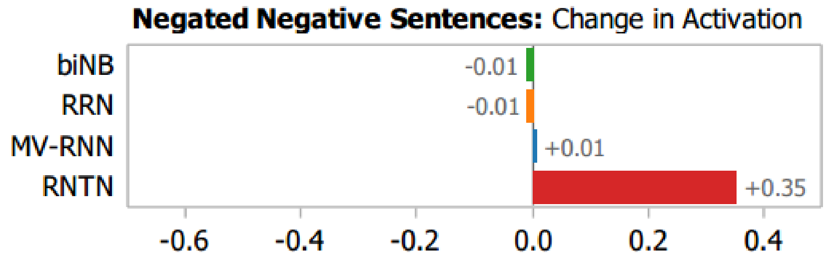
Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

- the negation changes the **overall** sentiment of a sentence **from positive to negative**.
- The **RNTN has the highest reversal accuracy**, showing its ability to structurally learn negation of positive sentences.

Experiments

Negating Negative Sentences (否定否定句)





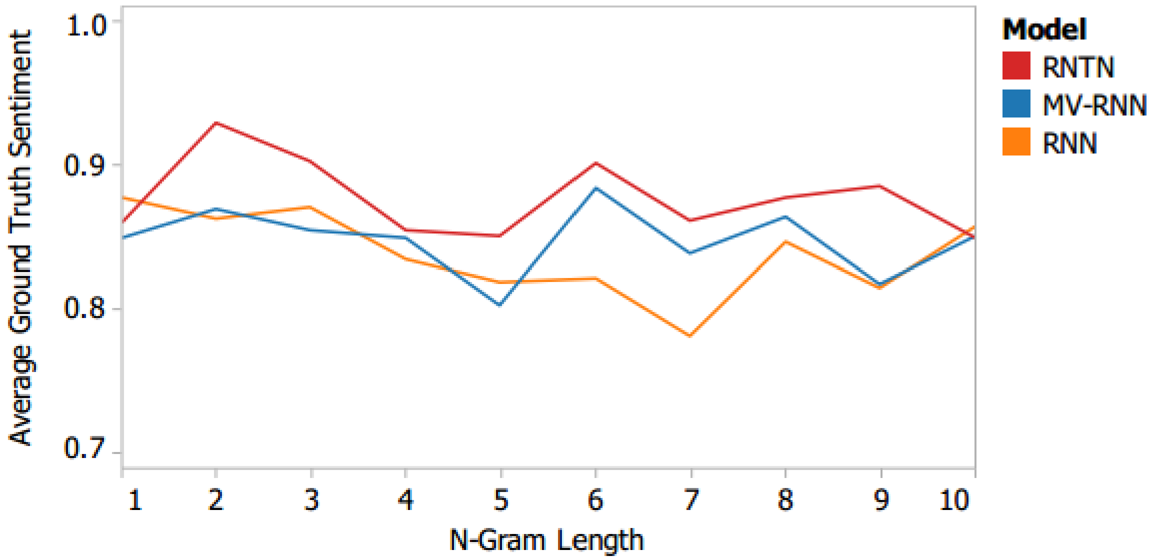
Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**81.8**

- When negative sentences are negated, overall sentiment should become **less negative**, but **not necessarily positive**.
- The **RNTN has the largest shifts** in the correct directions. **Only the RNTN** correctly captures **both types**.

Experiments

Most Positive and Negative Phrases

n	Most positive n-grams
1	engaging; best; powerful; love; beautiful
2	excellent performances; A masterpiece; masterful film; wonderful movie; marvelous performances
3	an amazing performance; wonderful all-ages triumph; a wonderful movie; most visually stunning
5	nicely acted and beautifully shot; gorgeous imagery, effective performances; the best of the year; a terrific American sports movie; refreshingly honest and ultimately touching
8	one of the best films of the year; A love for films shines through each frame; created a masterful piece of artistry right here; A masterful film from a master filmmaker,



- **RNTN selects more strongly** positive phrases **at most n-gram lengths** compared to other models.

Thank you