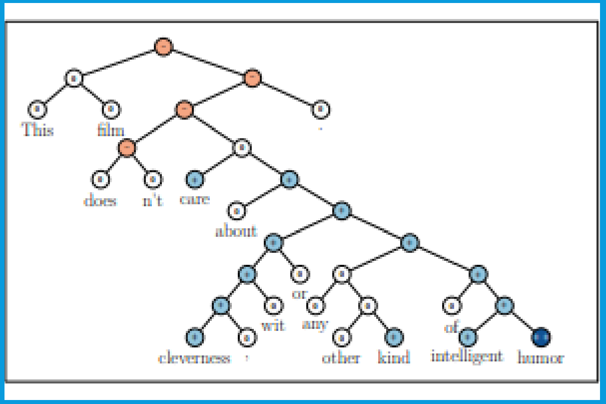
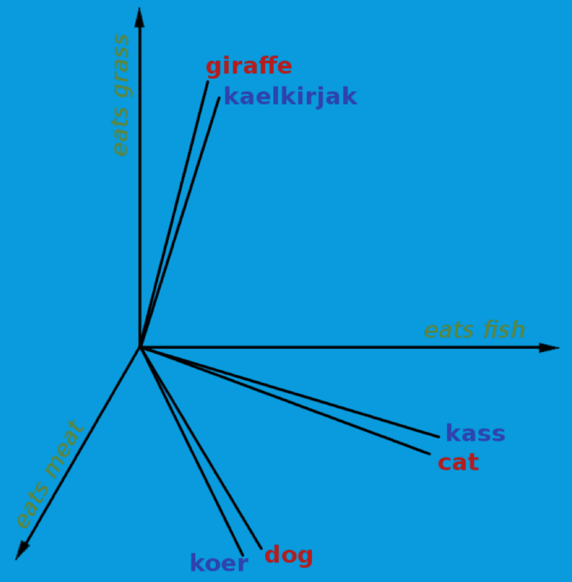
RECURSIVE DEEP MODELS FOR SEMANTIC COMPOSITIONALITY OVER A SENTIMENT TREEBANK

Gleb Alexeev





IMPORTANT DEFINITIONS

- Semantic Word Space
  - Aim to capture meaning in a phrase or text by providing representations of natural language. [1]
- Semantic Compositionality
  - The principle that the meaning of a (syntactically complex) whole is a function only of the meanings of its (syntactic) parts together with the manner in which these parts were combined. [2] In other words, describing a function using it's parts and the operations between them. For example, to describe $f(x,y,z)$, we set it equal to the operations based on its parts: $f(x,y,z) = y(x + z)/(x-y) + xyz$ (simple example)

INTRODUCTION

- Problems with meaning captured in longer phrase representation through semantic vector spaces used as features.
- Semantic Compositionality receiving a lot of attention, but likewise held back by absence of labeled data.
- Point of the paper:
  - Provide a new corpus, *The Stanford Sentiment Treebank*
  - Provide a powerful and accurate model using Recursive Neural Tensor Networks
  - Compare between other models on different aspects to the corpus

STANFORD SENTIMENT TREEBANK

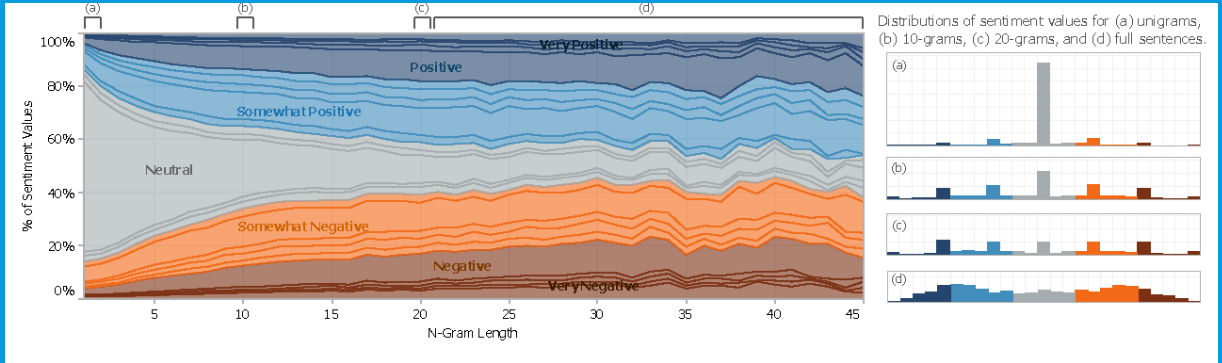
- Corpus based on dataset introduced by Pang and Lee (2005) and consists of 11,855 single sentences extracted from movie reviews. [3]
- 215,154 unique phrases, each annotated by 3 human judges.
- https://nlp.stanford.edu/sentiment/treebank.html
- There are other treebanks available, but not enough on short comments like Twitter. (less overall signal per document)

STANFORD SENTIMENT TREEBANK CONT'D

- Bag of Words
  - Works well with strong sentiments, but still on average has achieved 80% accuracy for binary classification problems
  - 60% with multiclass.
- Ignoring word order is not plausible, especially in the case of negation, and Stanford sentiment deals with that, by providing an n-gram model.
- Used Amazon Mechanical Turk (interface) to label 215,154 phrases and n-grams taken from rottentomatoes.com corpus, which was basically a slider:

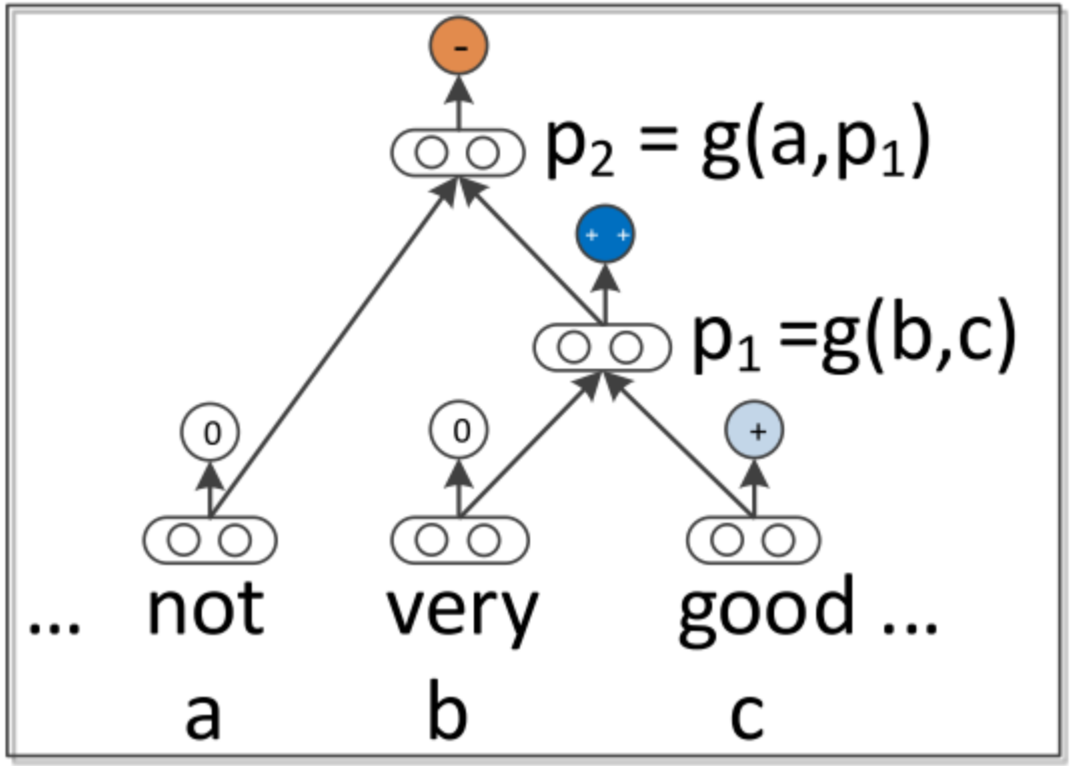
STANFORD SENTIMENT TREEBANK CONT'D

- Observations:
  1. Reader's perception is that many of the sentences could be neutral
  2. Stronger sentiment builds up in longer phrases, and the majority of shorter phrases are neutral
  3. Most annotators moved the slider to one of 5 options: negative, somewhat negative, neutral, somewhat positive, and positive
     - This forms a 5 class classification capturing most of the variance within the labels. (named fine-grained sentiment classification).
- The main point of the experiment was to recover these 5 labels for phrases of all lengths.



RECURSIVE NEURAL MODELS

- Parse a given n-gram into a binary tree and represent each word (corresponding to leaves in the tree) using a d-dimensional vector.
- Compute parent vectors using a bottom-up approach using different composition functions.
- To start with, word vectors are initialized randomly from a uniform distribution.
- For classification task, use the compositions word vectors as input for the softmax.
- Different models differ in terms of how word vectors are combined together as shown in the figure.


-
$p_2 = g(a,p_1)$
+ +
$p_1 = g(b,c)$
0
0
+
... not
very
good ...
a
b
c

RNN

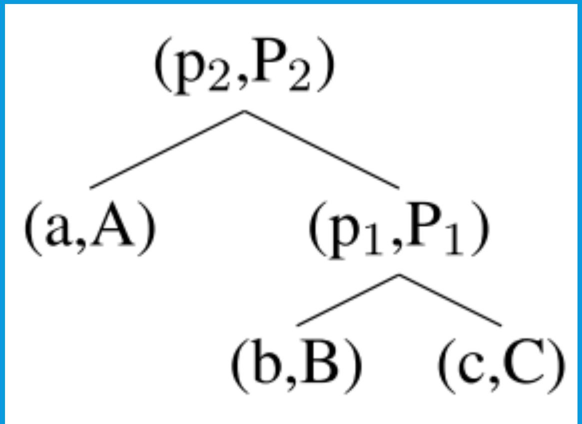
- Neural Network
  - Words represented as d-dimensional vectors, used to optimize parameters
  - Compute posterior via softmax
- RNN
  - Model uses parent and child vector inter-relations
  - f = tanh
  - *W* is the weight matrix to be learnt.

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

MV-RNN: MATRIX-VECTOR RNN

- Main idea
  - Represent every word and phrase as both a vector and a matrix.
  - Matrix for each word is initialized as identity matrix plus a small Gaussian noise.
- Example parse tree, with example equation used:

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

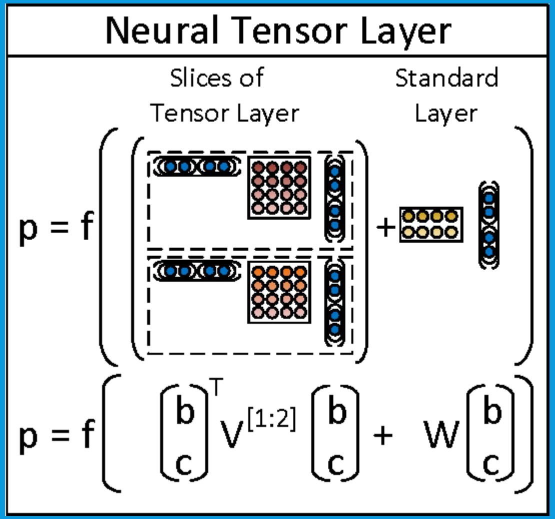


Disadvantage:

Number of parameters of MV-RNN becomes extremely large for even slightly larger relations (as each word is represented as a dxd matrix)

RNTN: RECURSIVE NEURAL TENSOR NETWORK

- Asks the question: can a single composition function form and compose better aggregate meaning from smaller constituents more accurately than many input specific ones?
    - The answer is yes, thanks RNTN!
    - Picture shows as single layer of the recursive neural tensor network. Each dashed box represents one of d-many slices and can capture a type of influence a child can have on its parent.
  - *V* is the tensor that defines multiple bilinear forms.
  - Each slice of the tensor *V* can be interpreted as capturing a specific type of composition



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

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

EXPERIMENTS

- Fine-grained Sentiment For All Phrases
- Full Sentence Binary Sentiment
- Contrastive Conjunction
  - Sentences of the form *X but Y*
- High Level Negation
  - Negating Positive Sentences
  - Negating Negative Sentences

RESULTS

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**

- Models compared with
  - Naive Bayes - NB
  - SVMs
  - Naive Bayes with bag of bigram features - biNB
  - Average neural word vectors (ignoring word order) – vecAvg

- RNTN outperforms other models in special cases like where a positive sentence is negated or where a negative sentence is negated to make it less negative (not positive though). This suggests that RNTN could capture the effect of negative words in both positive and negative sentiment sentences.

BIBLIOGRAPHY

[1] Baroni, Marco; Lenci, Alessandro. "Distributional Memory: A General Framework for Corpus-Based Semantics". Computational Linguistics. 36 (4): 673–721. doi:10.1162/coli_a_00016.

[2] Pelletier, F.J. Topoi (1994) 13: 11. https://doi.org/10.1007/BF00763644

[3] http://www.cs.cornell.edu/people/pabo/movie-review-data/

[4] The paper itself (see syllabus, but here’s the link anyway) https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf