Compositional Sentiment Analysis

Stephen Pulman

Dept. of Computer Science, Oxford University
and TheySay Ltd
stephen.pulman@cs.ox.ac.uk, stephen.pulman@theysay.io
@sgpulman

NYC Sentiment Symposium, March 5th 2014

UNIVERSITY OF OXFORD

TheySay

Building a sentiment analysis system

Version 1: cheap and cheerful

- collect lists of positive and negative words or phrases, from public domain lists or by mining them.
- given a text, count up the number of positives and negatives, and classify based on that.

Problems:

- if number of positive = number of negatives, do we say 'neutral'?
- **Compositional sentiment:** a phrase like 'not wonderfully interesting' is negative, even though 'wonderfully' and 'interesting' will be in the list of positive words.

Version 2: a bag-of-words classifier

- get a training corpus of texts human annotated for sentiment (e.g. pos/neg/neut).
- represent each text as a vector of counts of *n*-grams[1] of (normalised) words, and train your favourite classifier on these vectors.
- should capture some 'compositional' effects: e.g. 'very_interesting' likely signal for positivity, whereas 'not_very' a signal for negativity.
- will work for any language and domain where you can get accurately labelled training data.
- bag-of-words means structure is largely ignored:
  "Republican block on Democrat move"
  = "Democrat block on Republican move"

[1] *n* usually <= 3, and as *n* gets bigger, more training data is required

Problems:

- Equally balanced texts will still be problematic
- and richer compositional effects will still be missed:

  clever, too clever, not too clever
  bacteria, kill bacteria, fail to kill bacteria
  never fail to kill bacteria

- and difficult to give sentiment labels accurately to phrases.
- or to pick out mixed sentiment:

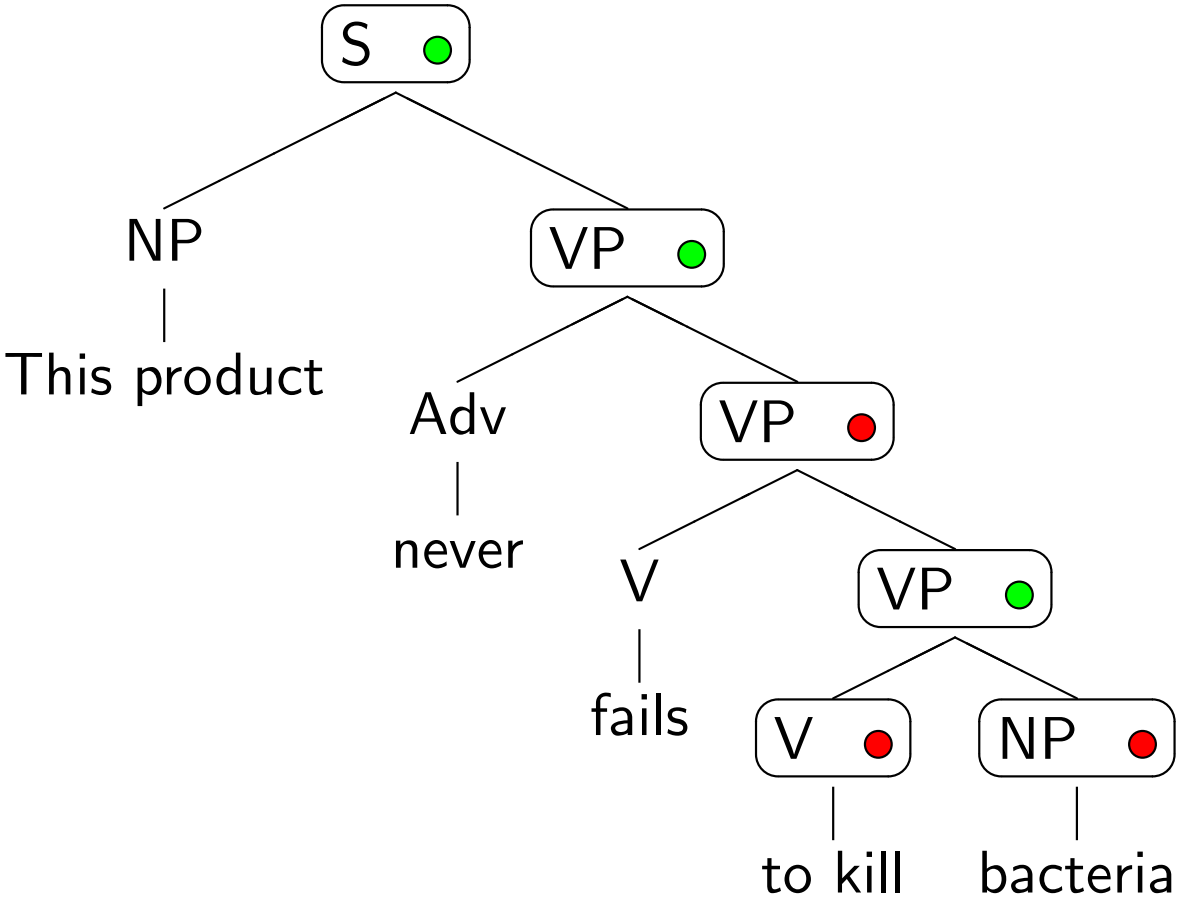
  "The display is amazingly sharp. However, the battery life is disappointing."

- such examples occur quite frequently in practice:

  The Trout Hotel: This newly refurbished hotel could not fail to impress
  it would not be possible to find a worse company to deal with

Version 3: best - use linguistic analysis

- do as full a parse as possible on input texts.
- use the syntax to do 'compositional' sentiment analysis:



Sentiment logic rules[2]

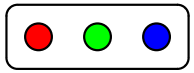
- kill + negative → positive (kill bacteria)
- kill + positive → negative (kill kittens)
- too + anything → negative (too clever, too red, too cheap)
- etc. In our system (`www.theysay.io`) we have 65,000+ of such rules...

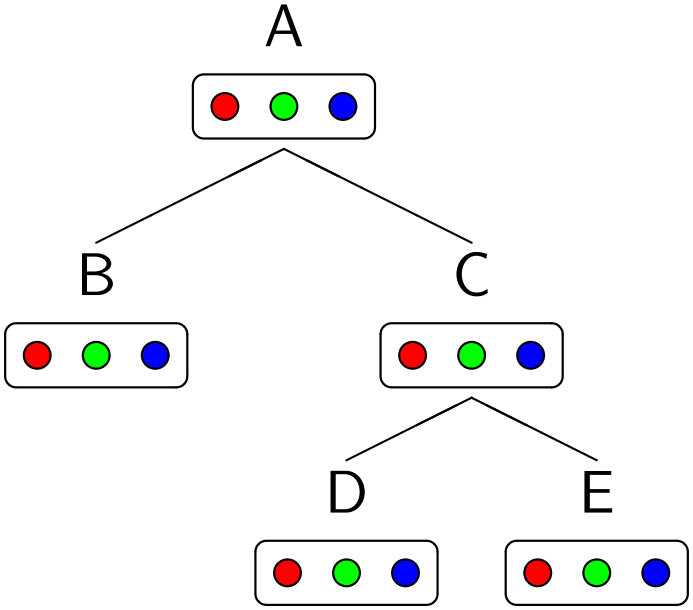
Problems:

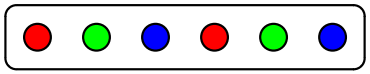
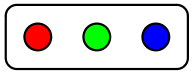
- still need extra work for context-dependence ('cold', 'wicked', 'sick'...)
- can't deal with reader perspective: "Oil prices are down" is good for me, not for Chevron or Shell investors.
- can't deal with sarcasm or irony: "Oh, great, they want it to run on Windows"

[2] Moilanen and Pulman, 2007

Machine learning for composition[3]

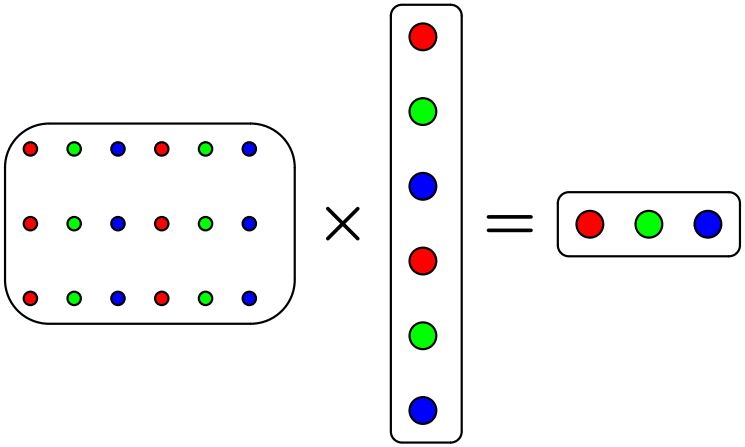
Assume we have a 'sentiment treebank'. Represent words as vectors  .



To compute C's vector, we concatenate those of D and E  , and learn from the training data a function which combines them in the 'right' way to form a  . Likewise we combine B and C to find A.

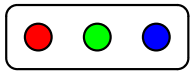
[3] Hermann and Blunsom, 2013; Socher et al 2013

Various composition functions

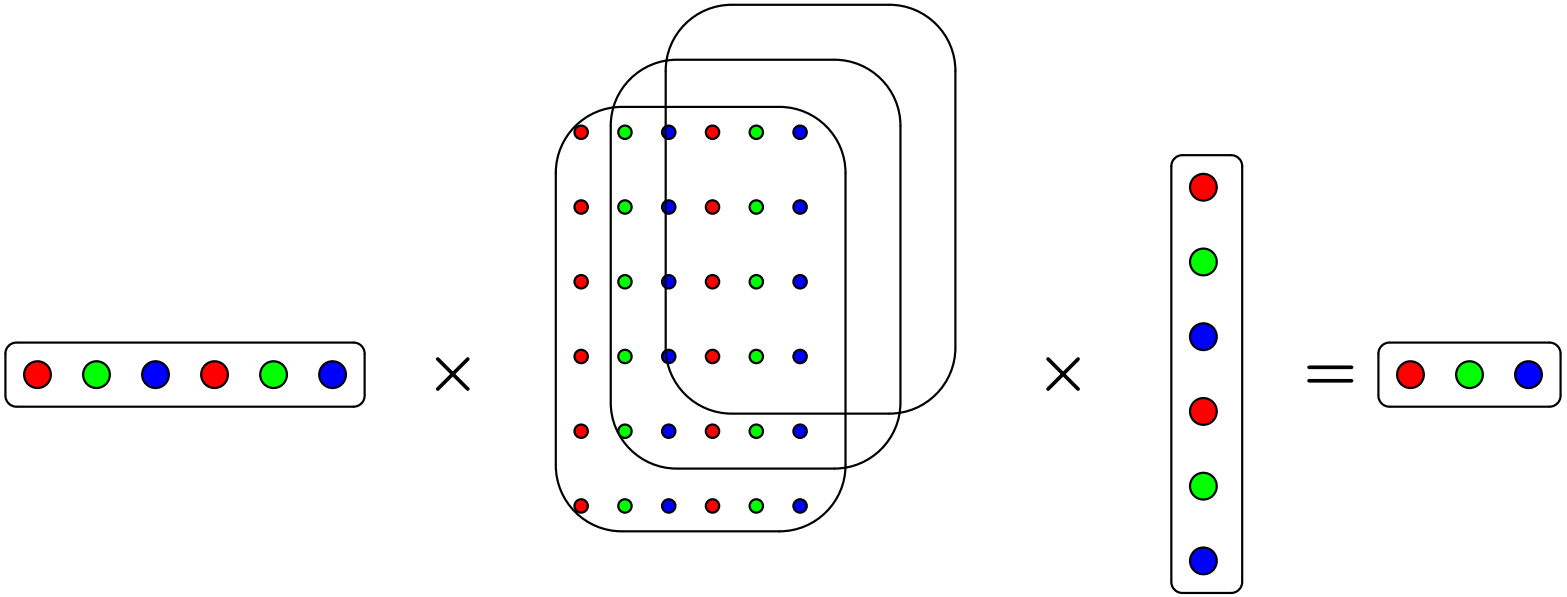


Here the weight matrix represents pairwise word/phrase combinations, perhaps also with syntactic info. We apply an element-wise non-linearity to the resulting vector. Weights can be learned via neural network methods.

We then use a 'softmax' function to map from the phrase vectors to a distribution over sentiment labels:

 $\text{Softmax}_{\Rightarrow}$ $\{\text{Pos} = 0.5, \text{Neg} = 0.3, \text{Neut} = 0.2\}$

More complex



Here a bilinear tensor operation combines the concatenated child vectors with their transpose: each slice of the tensor represents a different composition operation.

Socher's system, trained and tested on a treebank derived from a standard movie review corpus gives better results on both multi-label and binary sentiment analysis.

Does compositional approach work better in applications?[4]

NFP: a monthly economic index that measures job growth or decay:
- a 'market mover'.

Questions:

- Can we predict the direction of the NFP from financial indicators?
- Can we predict the direction of the NFP from sentiment in text?
- If so, does compositional sentiment perform better than BOW classifier?

[4] Joint work with Oxford Man Institute of Quantitative Finance: Levenberg et al. 2014

Back tested over data from 2000-2012

Almost 10m words of text containing relevant keys:

Source	Sentences
Associated Press	54K
Dow Jones	236K
Reuters	169K
Market News	385K
Wall Street Journal	76K

- and financial time-series data from many different sources, including:

Consumer Price Index (CPI)
Institute of Supply Management index (ISM)
Job Openings and Labor Turnover Survey (JOLTS)

Process text using TheySay's API:

"The Governor noted that despite jobs being down, there was a surprising bright spot: construction added 1,900 jobs in November - its largest gain in 22 months."

pos: 0.925, neg: 0.0, neut: 0.075, conf: 0.69

"When I drive down the main street of my little Kansas City suburb I see several dark empty storefronts that didn't used to be that way."

pos: 0.0, neg: 0.973, neut: 0.027, conf: 0.674

"We continue to fare better than the nation - our rate has been at or below the national rate for 82 out of the past 83 months - but we must also recognize that there were 10200 jobs lost at the same time."

pos: 0.372, neg: 0.591, neut: 0.037, conf: 0.723

Bag of words vs. compositional sentiment

Method:

- Train a bigram SVM classifier for comparison.
- Use output sentiment distributions as feature for ML classifiers.
- Train individual logistic regression classifiers on text and numerical streams.
- Use a novel 'Bayesian classifier combination' method to get best combination of individual classifiers.
- **Summary of results:**

  Classifier combination beats individual classifiers
  BOW classifier combo peaks at 67% (AUC)
  Compositional sentiment combo 85%
  Compositional + financial time series **94%**

Conclusions

- Compositional sentiment methods give a substantial improvement in accuracy
- . . . and finer grained analyses
- . . . and in the financial domain at least can yield accurate predictions, especially when combined with numerical data.

Try it out!

`www.theysay.io`

`http://nlp.stanford.edu:8080/sentiment/rntnDemo.html`

References

Karo Moilanen and Stephen Pulman, 2007, *Sentiment composition.* In Proceedings of the Recent Advances in Natural Language Processing International Conference (RANLP-2007), pages 378382, Borovets, Bulgaria, September 27-29 2007.
`http://www.clg.ox.ac.uk/_media/people:karo:sentcompranlp07final.pdf`

Karl Moritz Hermann and Phil Blunsom, 2013, *The Role of Syntax in Vector Space Models of Compositional Semantics.* In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 894904. Sofia, Bulgaria.
`http://aclweb.org/anthology//P/P13/P13-1088.pdf`

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Chris Manning, Andrew Ng and Chris Potts, 2013, *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank* Conference on Empirical Methods in Natural Language Processing (EMNLP 2013), Seattle, Washington, USA; ACL, pp. 1631-1642.
`http://aclweb.org/anthology//D/D13/D13-1170.pdf`

Abby Levenberg, Stephen Pulman, Karo Moilanen, Edwin Simpson and Stephen Roberts, 2014, *Predicting Economic Indicators from Web Text Using Sentiment Composition*, to appear in International Journal of Computer and Communication Engineering. Barcelona, Spain. February, 2014. IACSIT Press.