Information Extraction

Seminar – Sentiment Analysis (Part 2)

CIS, LMU München
Winter Semester 2023-2024

Alexander Fraser, CIS

Two Approaches to Classifying Documents

- Bottom-Up
  - Assign sentiment to words
  - Derive clause sentiment from word sentiment
  - Derive document sentiment from clause sentiment

- Top-Down
  - Get labeled documents
  - Use text categorization methods to learn models
  - Derive word/clause sentiment from models

Slide modified from Koppel/Wilson

Bottom-Up Sentiment Analysis

- We saw this in the first part of this lecture
- Key concepts:
  - Prior polarity (from sentiment lexicon)
  - Clause-level
    - Particularly negation
- Heavy emphasis on feature engineering

Top-Down Sentiment Analysis

- So far we've seen attempts to determine document sentiment from word/clause sentiment
- Now we'll look at the old-fashioned supervised method: get labeled documents and learn models

Slide from Koppel/Pang/Gamon

Finding Labeled Data

- Online reviews accompanied by star ratings provide a ready source of labeled data
  - movie reviews
  - book reviews
  - product reviews

Slide from Koppel/Pang/Gamon

Movie Reviews (Pang, Lee and V. 2002)

- Source: Internet Movie Database (IMDb)

- 4 or 5 stars = positive; 1 or 2 stars = negative
  - 700 negative reviews
  - 700 positive reviews

Slide from Koppel/Pang/Gamon

Evaluation

- Initial feature set:
  - 16,165 unigrams appearing at least 4 times in the 1400-document corpus
  - 16,165 most often occurring bigrams in the same data
  - Negated unigrams (when "not" appears to the left of a word)

- Test method: 3-fold cross-validation

  (so about 933 training examples)

Slide from Koppel/Pang/Gamon

Results

	Features	# of features	frequency or presence?	NB	ME	SVM
(1)	unigrams	16165	freq.	**78.7**	N/A	72.8
(2)	unigrams	"	pres.	81.0	80.4	**82.9**
(3)	unigrams+bigrams	32330	pres.	80.6	80.8	**82.7**
(4)	bigrams	16165	pres.	77.3	**77.4**	77.1
(5)	unigrams+POS	16695	pres.	81.5	80.4	**81.9**
(6)	adjectives	2633	pres.	77.0	**77.7**	75.1
(7)	top 2633 unigrams	2633	pres.	80.3	81.0	**81.4**
(8)	unigrams+position	22430	pres.	81.0	80.1	**81.6**

Figure 3: Average three-fold cross-validation accuracies, in percent. Boldface: best performance for a given setting (row). Recall that our baseline results ranged from 50% to 69%.

Slide from Koppel/Pang/Gamon

Observations

- In most cases, SVM slightly better than NB
- Binary features good enough
- Drastic feature filtering doesn’t hurt much
- Bigrams don’t help (others have found them useful)
- POS tagging doesn’t help
- Benchmark for future work: 80%+

Slide from Koppel/Pang/Gamon

Looking at Useful Features

- Many top features are unsurprising (e.g. *boring*)
- Some are very unexpected
  - *tv* is a negative word
  - *flaws* is a positive word
- That's why bottom-up methods are fighting an uphill battle

Slide from Koppel/Pang/Gamon

Other Genres

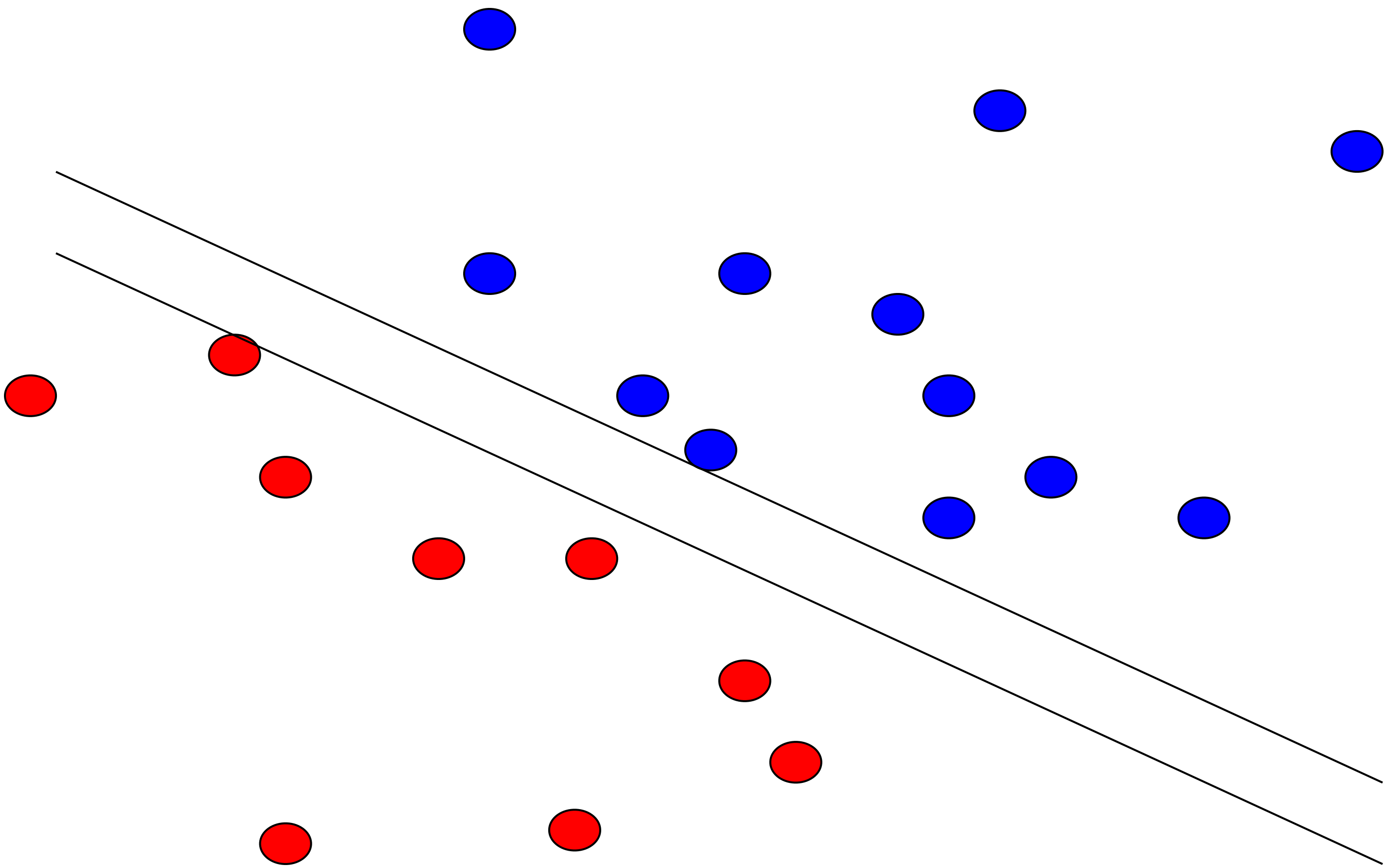
- The same method has been used in a variety of genres
- Results are better than using bottom-up methods
- Using a model learned on one genre for another genre does not work well

Cheating (Ignoring Neutrals)

- One nasty trick that researchers use is to ignore neutral data (e.g. movies with three stars)
- Models learned this way won’t work in the real world where many documents are neutral
- The optimistic view is that neutral documents will lie near the negative/positive boundary in a learned model.

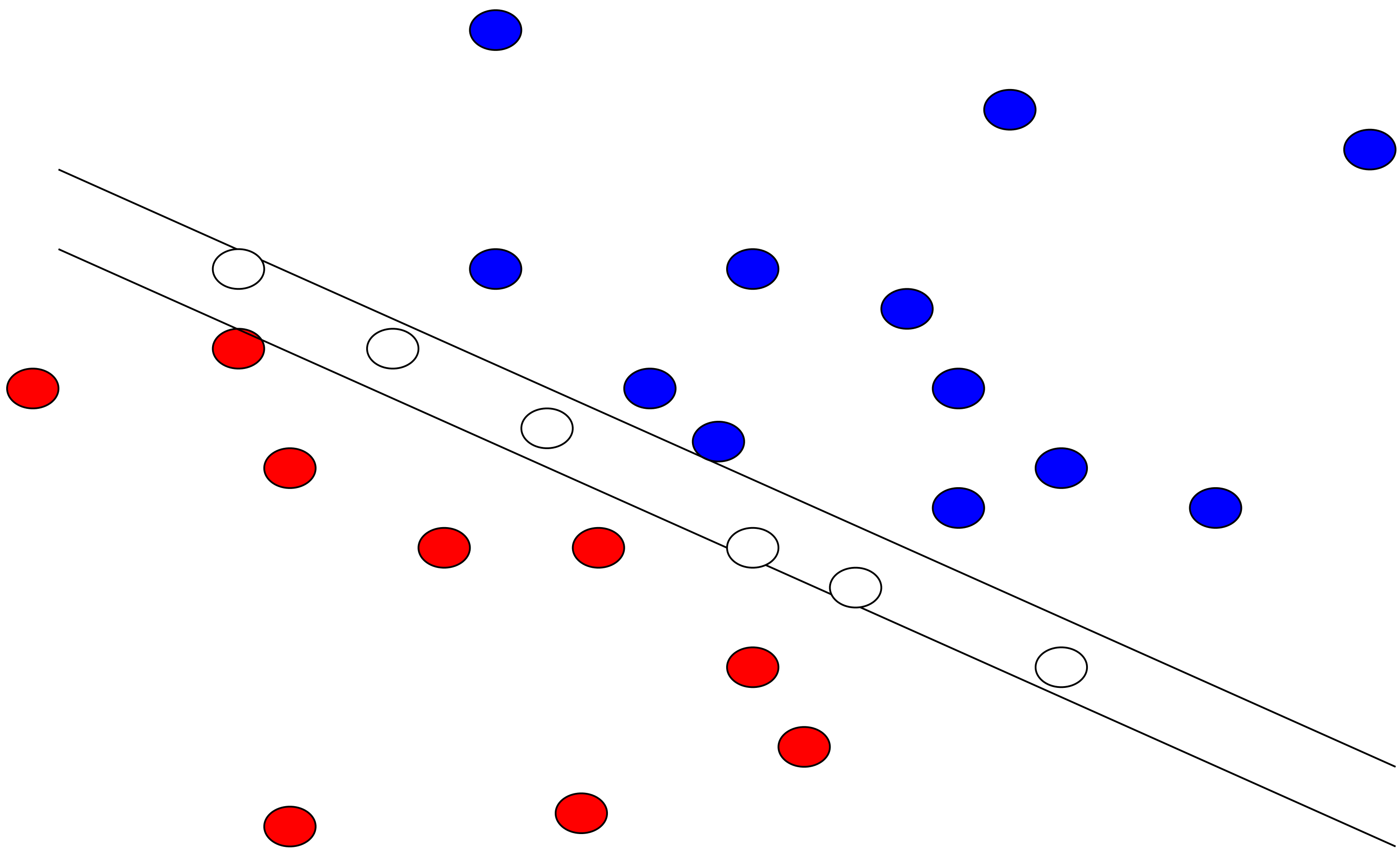
Slide modified from Koppel/Pang/Gamon

A Perfect World

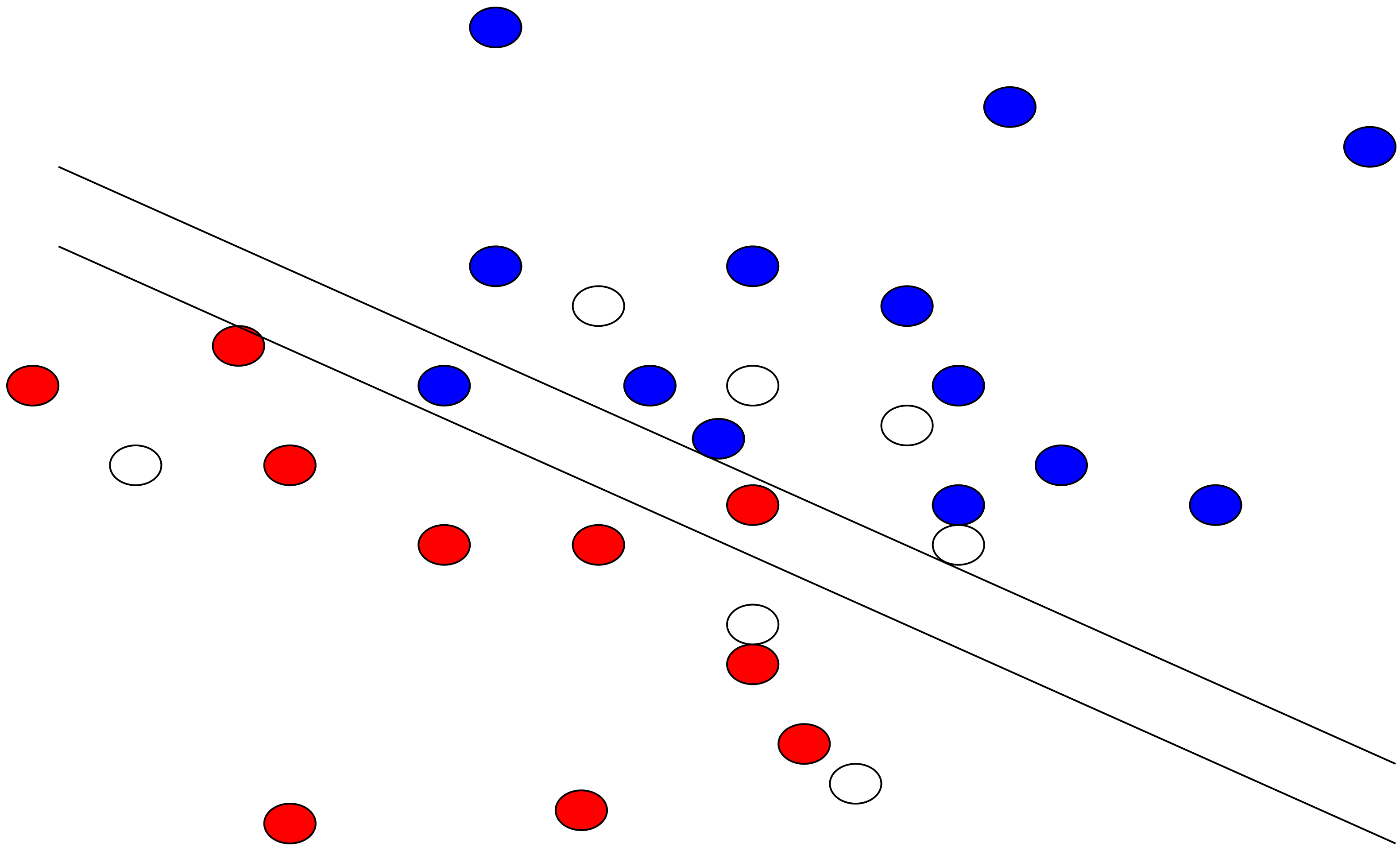


Slide from Koppel/Pang/Gamon

A Perfect World



The Real World



Slide from Koppel/Pang/Gamon

Some Obvious Tricks

- Learn separate models for each category or
- Use regression to score documents

But maybe with some ingenuity we can do even better.

Slide from Koppel/Pang/Gamon

Corpus

We have a corpus of 1974 reviews of TV shows, manually labeled as positive, negative or neutral

Note: neutrals means either no sentiment (most) or mixed (just a few)

For the time being, let's do what most people do and ignore the neutrals (both for training and for testing).

Slide from Koppel/Pang/Gamon

Basic Learning

- Feature set: 500 highest infogain unigrams
- Learning algorithm: SMO
- 5-fold CV Results: 67.3% correctly classed as positive/negative

OK, but bear in mind that this model won't class any neutral test documents as neutral – that's not one of its options.

Slide from Koppel/Pang/Gamon

So Far We Have Seen..

… that you need neutral training examples to classify neutral test examples

In fact, it turns out that neutral training examples are useful even when you know that all your test examples are positive or negative (not neutral).

Slide from Koppel/Pang/Gamon

Multiclass Results

OK, so let's consider the three class (positive, negative, neutral) sentiment classification problem.

On the same corpus as above (but this time not ignoring neutral examples in training and testing), we obtain accuracy (5-fold CV) of:

- **56.4%** using multi-class SVM
- **69.0%** using linear regression

Slide from Koppel/Pang/Gamon

Can We Do Better?

But actually we can do much better by combining pairwise (pos/neg, pos/neut, neg/neut) classifiers in clever ways.

When we do this, we discover that pos/neg is the least useful of these classifiers (even when all test examples are known to not be neutral).

Let’s go to the videotape…

Slide from Koppel/Pang/Gamon

Optimal Stack

Pos Vs Neg	**Pos Vs Neut**	**Neut Vs neg**	**Actual category**		
			neg	**neut**	**pos**
Neg	Neut	Neg	**354**	52	
Neg	Neut	Neut	117	**154**	148
Neg	Pos	Neg		**47**	
Neg	Pos	Neut		9	**108**
Pos	Neut	Neg	**145**	69	
Pos	Neut	Neut	42	**225**	46
Pos	Pos	Neg		**90**	
Pos	Pos	Neut		12	**356**

Slide from Koppel/Pang/Gamon

Optimal Stack

Here's the best way to combine pairwise classifiers for the 3-class problem:

- *IF positive > neutral > negative THEN class is positive*
- *IF negative > neutral > positive THEN class is negative*
- *ELSE class is neutral*

Using this rule, we get accuracy of 74.9%

(OK, so we cheated a bit by using test data to find the best rule. If, we hold out some training data to find the best rule, we get accuracy of 74.1%)

Slide from Koppel/Pang/Gamon

Key Point

Best method does not use the positive/negative model at all – only the positive/neutral and negative/neutral models.

This suggests that we might even be better off learning to distinguish positives from negatives by comparing each to neutrals rather than by comparing each to each other.

Slide from Koppel/Pang/Gamon

Positive /Negative models

So now let's address our original question. Suppose I know that all test examples are not neutral. Am I still better off using neutral training examples?

Yes.

Above we saw that using (equally distributed) positive and negative training examples, we got **67.3%**

Using our optimal stack method with (equally distributed) positive, negative and neutral training examples we get **74.3%**

(The total number of training examples is equal in each case.)

Can Sentiment Analysis Make Me Rich?

Slide from Koppel/Pang/Gamon

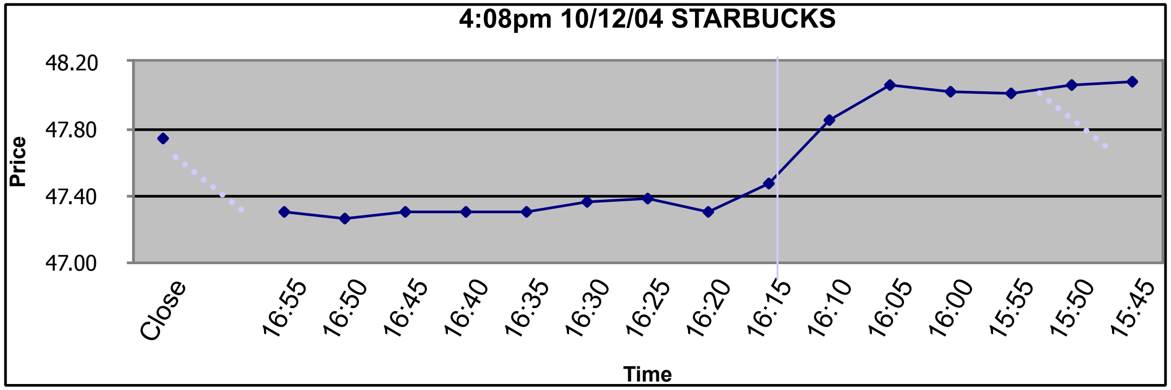
Can Sentiment Analysis Make Me Rich?

NEWSWIRE 4:08PM 10/12/04
STARBUCKS SAYS CEO ORIN SMITH TO RETIRE IN MARCH 2005

- How will these messages affect Starbucks stock prices?

Slide from Koppel/Pang/Gamon

Impact of Story on Stock Price



- Are price moves such as these predictable?
- What are the critical text features?
- What is the relevant time scale?

Slide from Koppel/Pang/Gamon

General Idea

- Gather news stories
- Gather historical stock prices
- Match stories about company X with price movements of stock X
- Learn which story features have positive/ negative impact on stock price

Slide from Koppel/Pang/Gamon

Experiment

- MSN corpus
  - 5000 headlines for 500 leading stocks September 2004 – March 2005.

- Price data
  - Stock prices in 5 minute intervals

Slide from Koppel/Pang/Gamon

Feature set

- Word unigrams and bigrams.
- 800 features with highest infogain
- Binary vector

Slide from Koppel/Pang/Gamon

Defining a headline as positive/negative

- If stock price rises more than Δ during interval T, message classified as positive.
- If stock price declines more than Δ during interval T, message is classified as negative.
- Otherwise it is classified as neutral.

With larger delta, the number of positive and negative messages is smaller but classification is more robust.

Slide from Koppel/Pang/Gamon

Trading Strategy

- Assume we buy a stock upon appearance of "positive" news story about company.
- Assume we short a stock upon appearance of "negative" news story about company.

Slide from Koppel/Pang/Gamon

Do we earn a profit?

Slide from Koppel/Pang/Gamon

Do we earn a profit?

- If this worked, I'd be driving a red convertible. (I'm not.)

Slide from Koppel/Pang/Gamon

Predicting the Future

- If you are interested in this problem in general, take a look at:

  Nate Silver

  The Signal and the Noise: Why So Many Predictions Fail - but Some Don't

  2012

  (Penguin Publishers)

Text Categorization
Deep Learning

(These deep learning slides are from Dr. Dario Stojanovski)

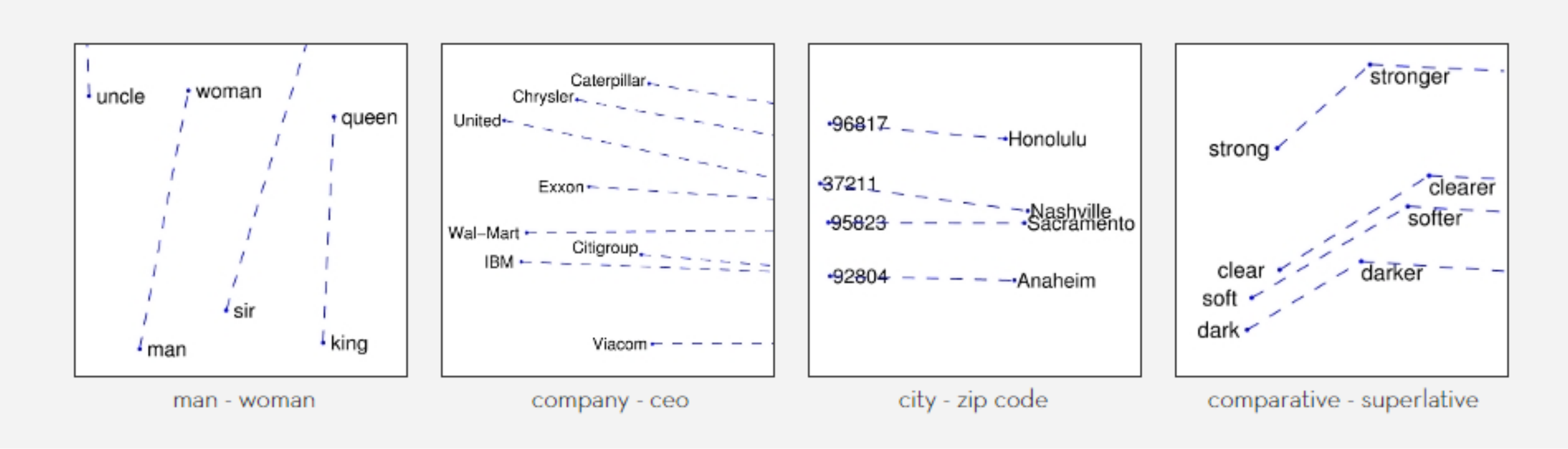
Machine learning

- Hand crafted features
  - In addition to unigrams: number of uppercase words, number of exclamation marks, number of positive and negative words …
- In social media domain:
  - emoticons, hashtags (#happy), elongated words (haaaapy)

Deep learning

- Automatic feature extraction
  - Learn feature representation jointly
- Little to no preprocessing required
- Takes into account word order
- General approaches:
  - Recursive Neural Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Self attention (Transformer)

Word embeddings



- Word embeddings capture syntactic and semantic regularities – no sentiment information encoded
- Good and bad are neighboring words

Pennington et al. 2014. GloVe: Global Vectors for Word Representation

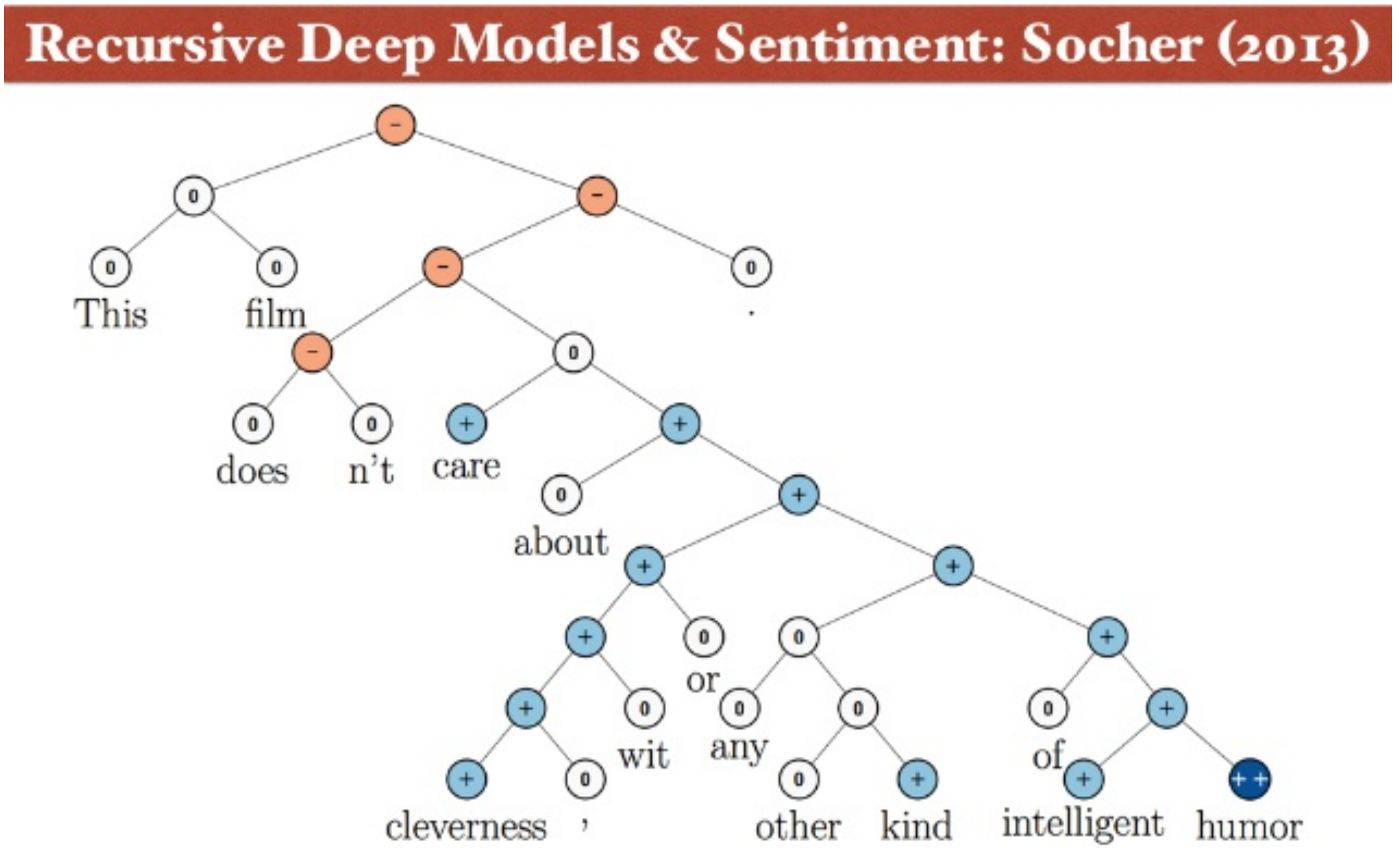
Word embeddings

- Update word embeddings by back-propagation
- Most similar words before (column 2) and after training (column 3)

bad	*good*	*terrible*
	terrible	*horrible*
	horrible	*lousy*
	lousy	*stupid*
good	*great*	*nice*
	bad	*decent*
	terrific	*solid*
	decent	*terrific*

Kim (2014)

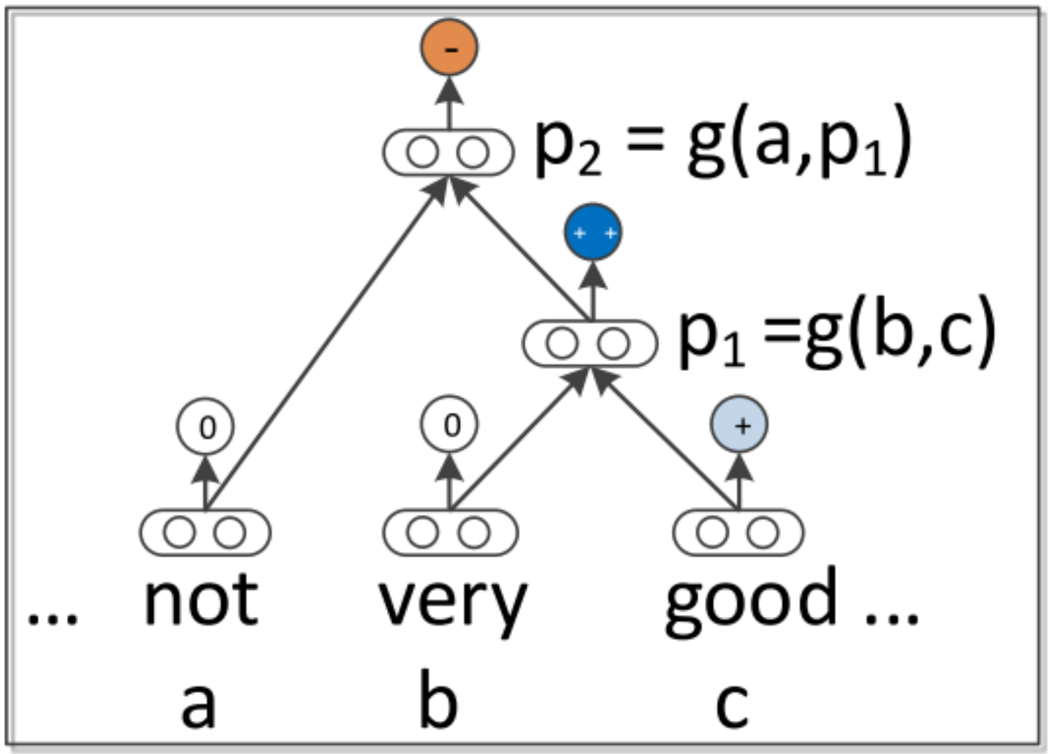
Recursive Neural Networks



Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., Potts, C. (2013)
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.

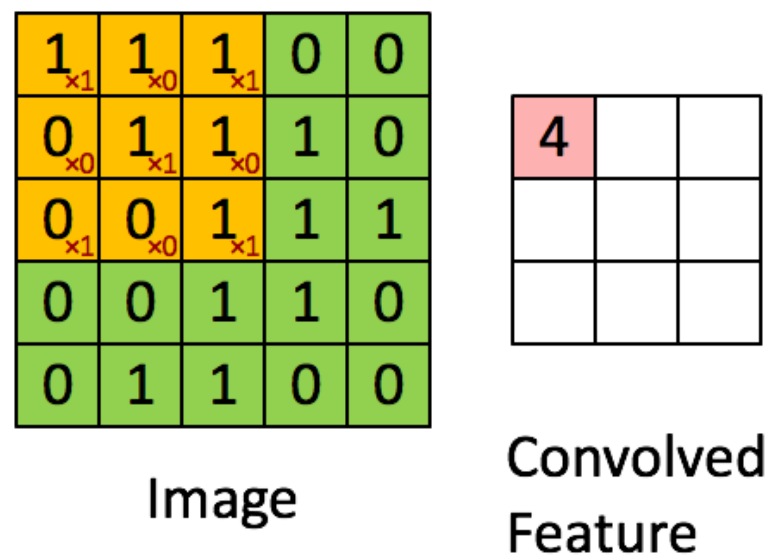
code & demo: http://nlp.stanford.edu/sentiment/index.html

Recursive Neural Networks

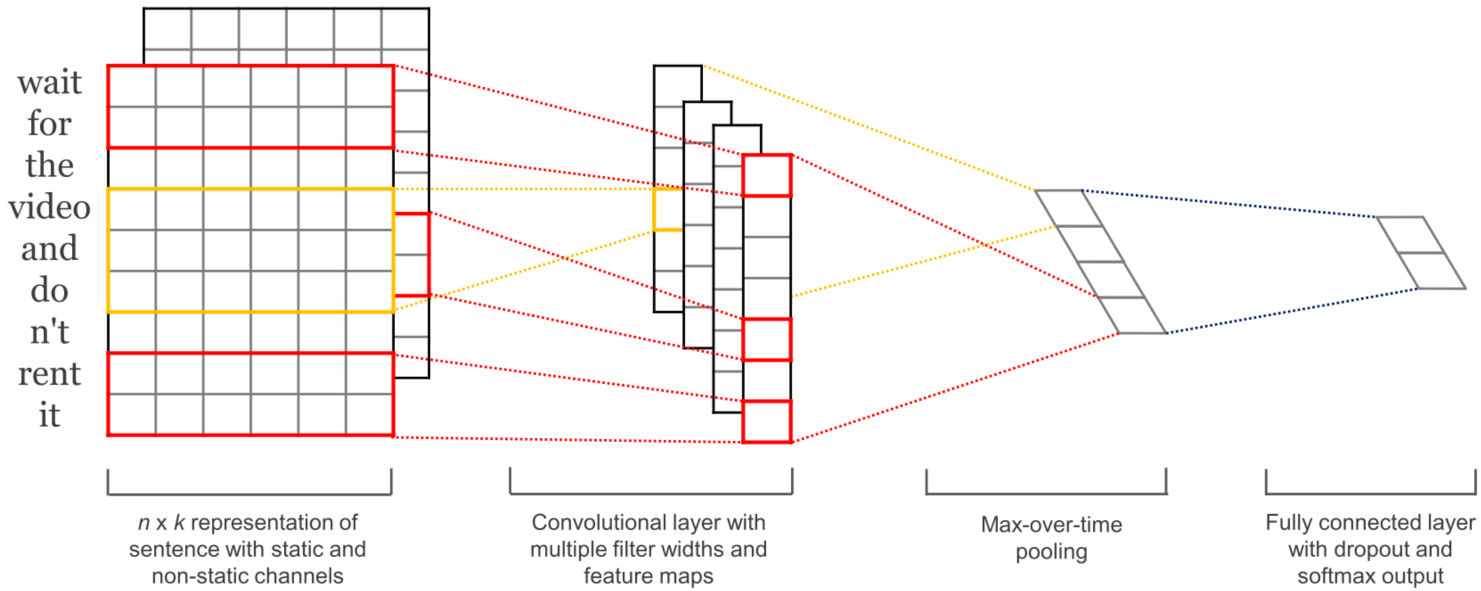


Convolutional Neural Networks

- Each row represents a word given by a word embedding with dimensionality *d*
- For a 10 word sentence, our “image” is a matrix of 10x*d*
- *(graphic from Ujjwal Karn)*

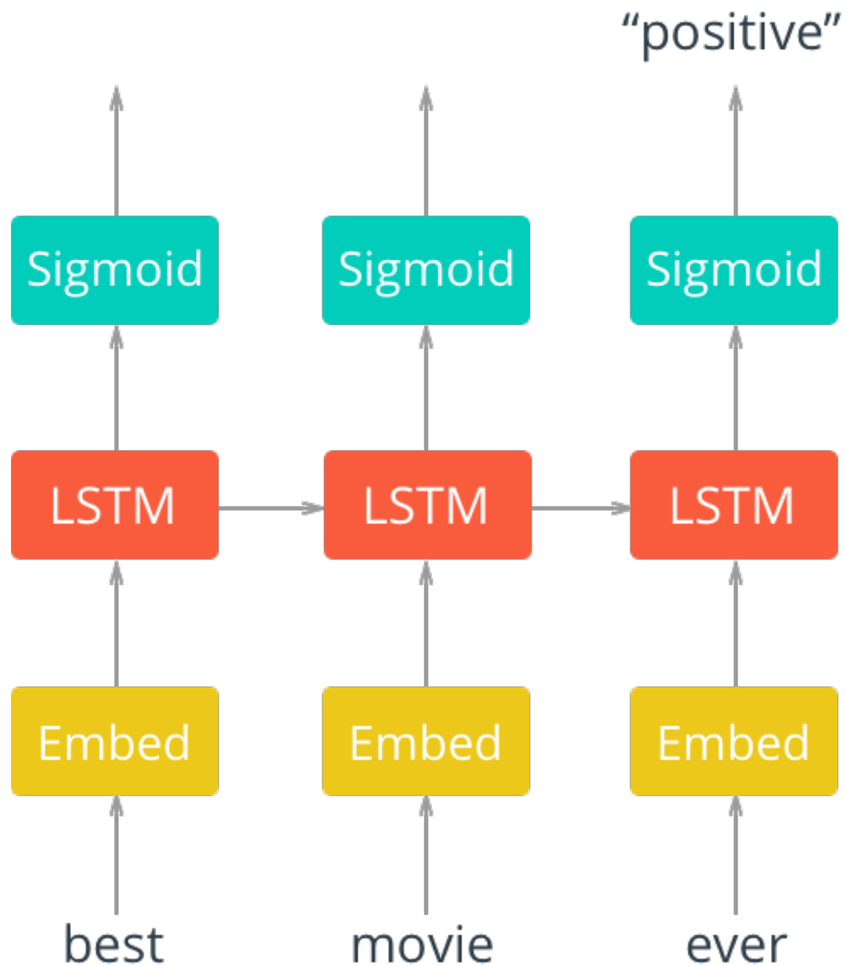


Convolutional Neural Networks



Kim (2014)

Recurrent Neural Networks

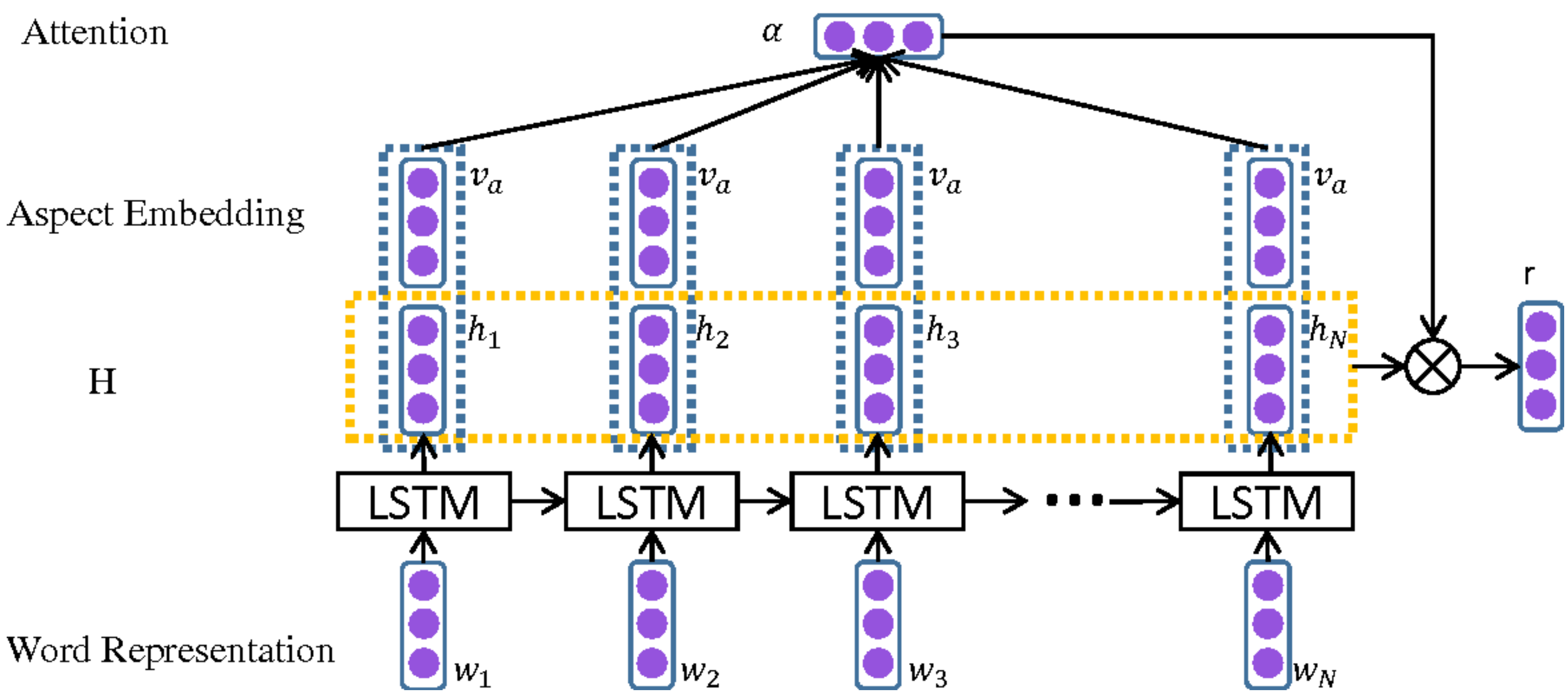


Sentiment Analysis using RNNs. Manish Chablani. 2017
https://towardsdatascience.com/sentiment-analysis-using-rnns-lstm-60871fa6aeba

Aspect-based Sentiment

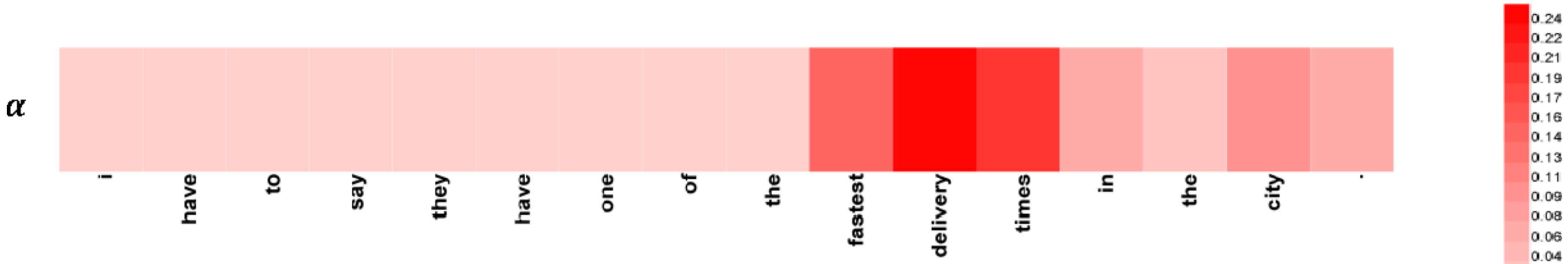
- What about aspect-based SA?
  - Interested in opinions towards multiple aspects
  - E.g. laptop: battery life, performance, screen …
  - We need a fine-grained way of getting the sentiment
- Attention-based models

Aspect-based model

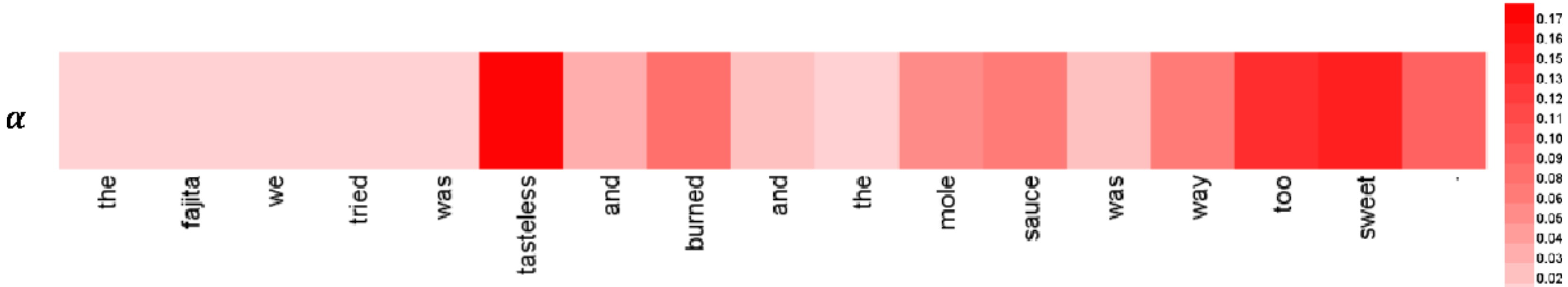


Wang et al. (2016)

Aspect-based model



(a) the aspect of this sentence: *service*



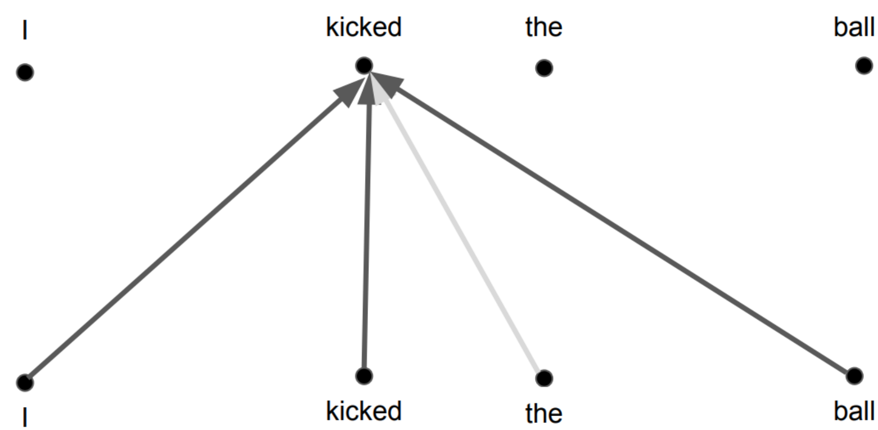
(b) the aspect of this sentence: *food*

Wang et al. (2016)

Transformer

- Self-attention model
  - Attention is all you need (Vaswani et al. 2017)
- Most work on NLP uses Transformers nowadays

Self-Attention



Taken and modified from: How Transformers Work. G Giacaglia. 2019
https://towardsdatascience.com/transformers-141e32e69591

BERT Pretraining

- Use very large monolingual data and train a Transformer language model
- Fine-tune your language model on sentiment analysis
- Takes advantage of huge monolingual data
- Probably all future work on sentiment analysis will use BERT (or variants of BERT) in one way or another

- Slide sources
  - Most slides before deep learnng are from Prof. Moshe Koppel (Bar-Ilan University)
  - Deep learning slides from Dr. Dario Stojanovski (CIS)
- Further reading on traditional sentiment approaches
  - 2011 AAAI tutorial on sentiment analysis from Bing Liu (quite technical)
- Deep learning for sentiment
  - See Stanford Deep Learning Sentiment Demo page
  - Kim, Yoon. "Convolutional neural networks for sentence classification." *EMNLP 2014*.
  - Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." EMNLP 2013.
  - Radford, Alec, Rafal Jozefowicz, and Ilya Sutskever. "Learning to generate reviews and discovering sentiment." *arXiv preprint arXiv:1704.01444* (2017).
  - Wang, Yequan, Minlie Huang, and Li Zhao. "Attention-based lstm for aspect-level sentiment classification." EMNLP 2016.

- Thank you for your attention!