

Information Extraction

Lecture 9 – Sentiment Analysis

CIS, LMU München

Winter Semester 2017-2018

Dario Stojanovski, CIS

Today

- Today we will take a tangent and look at another problem in information extraction: sentiment analysis

Sentiment Analysis

- Determine if a sentence/document expresses positive/negative/neutral sentiment towards some object

Some Applications

- **Review classification:** Is a review positive or negative toward the movie?
- **Product review mining:** What features of the ThinkPad T43 do customers like/dislike?
- **Tracking sentiments toward topics over time:** Is anger ratcheting up or cooling down?
- **Prediction (election outcomes, market trends):** Will Romney or Obama win?

Social media

- Twitter most popular
- Short (140 characters) and very informal text
- Abbreviations, slang, spelling mistakes
- 500 million tweets per day
- Tons of applications

Level of Analysis

We can inquire about sentiment at various linguistic levels:

- Words – objective, **positive**, **negative**, **neutral**
- Clauses – “*going out of my mind*”
- Sentences – possibly multiple sentiments
- Documents

Words

- Adjectives

- objective: red, metallic

- positive: honest important mature large patient

- negative: harmful hypocritical inefficient

- subjective (but not positive or negative):
curious, peculiar, odd, likely, probable

Words

– Verbs

- positive: **praise, love**
- negative: **blame, criticize**
- subjective: **predict**

– Nouns

- positive: **pleasure, enjoyment**
- negative: **pain, criticism**
- subjective: **prediction, feeling**

Clauses

- Might flip word sentiment
 - “*not good at all*”
 - “*not all good*”
- Might express sentiment not in any word
 - “*convinced my watch had stopped*”
 - “*got up and walked out*”

Sentences/Documents

- Might express multiple sentiments
 - *“The acting was great but the story was a bore”*
- Problem even more severe at document level

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

Some Special Issues

- Whose opinion?

(Writer)

(writer, Xirao-Nima, US)

(writer, Xirao-Nima)

“The US **fears** a spill-over”, **said** Xirao-Nima, a professor of foreign affairs at the Central University for Nationalities.

Some Special Issues

- Whose opinion?
- Opinion about what?

Laptop Review

- I should say that I am a **normal** user and this laptop satisfied all my expectations, the screen size is **perfect**, its very **light, powerful, bright**, lighter, elegant, delicate... But the only think that I regret is the Battery life, barely 2 hours... some times less... it is **too short**... this laptop for a flight trip is **not good** companion...
Even the short battery life I can say that I am **very happy** with my Laptop VAIO and I consider that I did the best decision. I am sure that I did the **best** decision buying the SONY VAIO

Advanced

- Sentiment towards a specific entity
 - Person, product, company
- Identify expressed sentiment towards several aspects of the text
 - Different features of a laptop
- Emotion Analysis
 - Identify emotions in text (love, joy, anger, sadness ...)

Word Sentiment

Let's try something simple

- Choose a few seeds with known sentiment
- Mark synonyms of **good** seeds: **good**
- Mark synonyms of **bad** seeds: **bad**
- Iterate

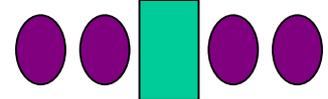
Word Sentiment

Let's try something simple

- Choose a few seeds with known sentiment
- Mark synonyms of **good** seeds: **good**
- Mark synonyms of **bad** seeds: **bad**
- Iterate

Not quite.

exceptional -> **unusual** -> **weird**



Better Idea

Hatzivassiloglou & McKeown 1997

1. Build training set: label all adj. with frequency > 20 ; test agreement with human annotators
2. Extract all **conjoined** adjectives

Web

Results 1 - 10 of about 762,000 for "was [very nice](#) and".

[The Homestay Experience - Cultural Kaleidoscope 2006](#)

My host's home **was very nice and** comfortable. I got to try all types of food; Malaysian, Chinese, Indonesian and I loved it all. My host's parents were very ...

www.gardenschool.edu.my/studentportal/aec/Kaleidoscope06/experience.asp - 10k -

[Cached](#) - [Similar pages](#) - [Note this](#)

[PriceGrabber User Rating for Watch Your Budget - PriceGrabber.com](#)

Reviews, Camera I purchased **was very nice and** a bargain. There was a problem with shipping, but was resolved quickly. Buy with confidence from this vendor. ...

www.pricegrabber.com/rating_getreview.php?retid=5821 - [Similar pages](#) - [Note this](#)

[Testimonials](#)

"Everybody **was very nice and** service was as fast as they possibly could. ... "Staff member who helped me **was very nice and** easy to talk to." ...

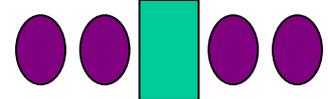
www.sa.psu.edu/uhs/news/testimonials.cfm - 22k - [Cached](#) - [Similar pages](#) - [Note this](#)

[Naxos Villages - Naxos Town or Chora Reviews: Very nice and very ...](#)

-Did you enjoy the trip to Naxos Town: Yes it **was very nice and** very scenic. -In order to get to the village were there enough signs in order to find it: It ...

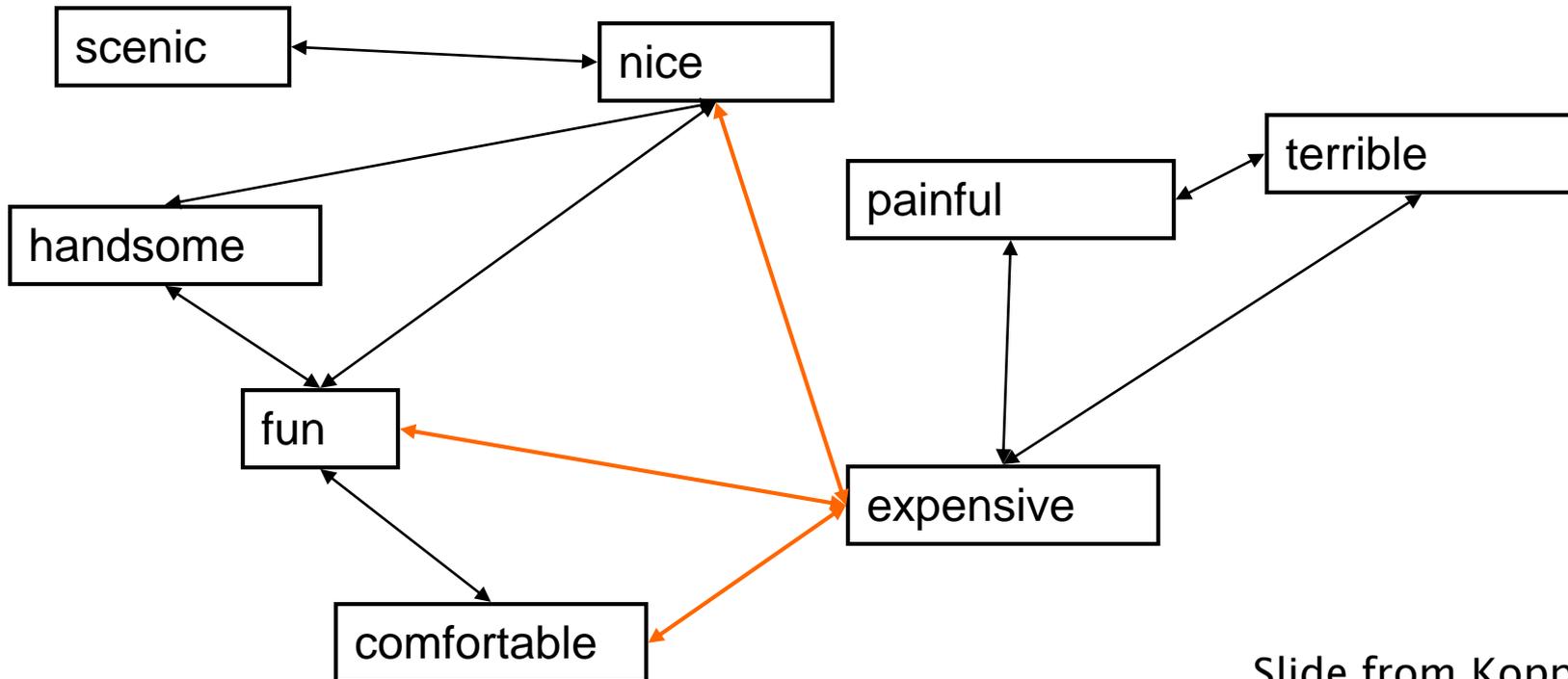


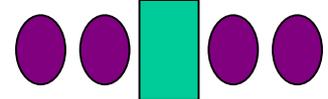
nice **and** comfortable
nice **and** scenic



Hatzivassiloglou & McKeown 1997

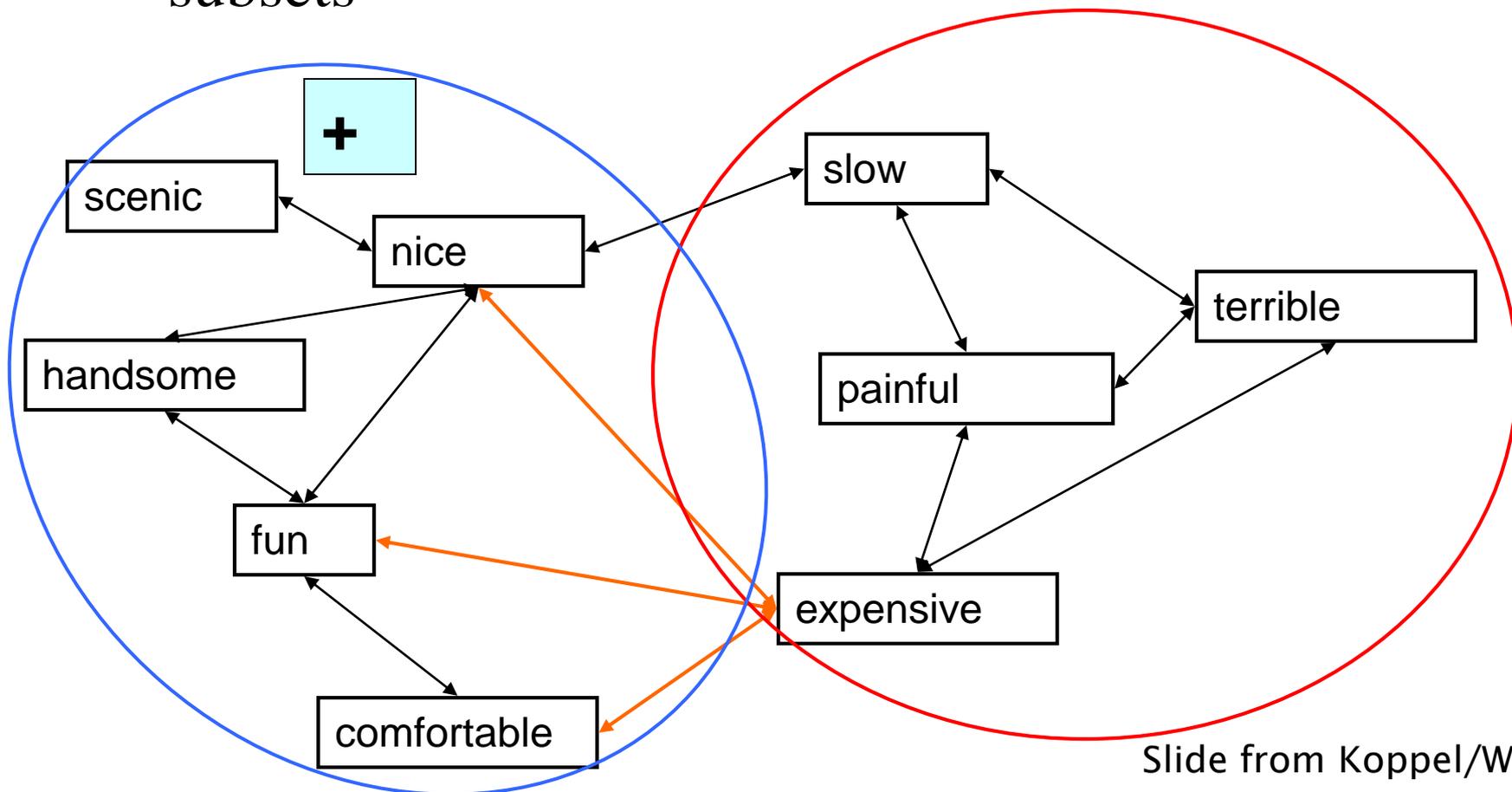
3. A supervised learning algorithm builds a **graph** of adjectives linked by the same or different semantic orientation





Hatzivassiloglou & McKeown 1997

4. A clustering algorithm partitions the adjectives into two subsets



Even Better Idea Turney 2001

- Pointwise Mutual Information (Church and Hanks, 1989):

$$\text{PMI}(word_1, word_2) = \log_2 \left(\frac{p(word_1 \wedge word_2)}{p(word_1)p(word_2)} \right)$$

Even Better Idea Turney 2001

- Pointwise Mutual Information (Church and Hanks, 1989):

$$\text{PMI}(word_1, word_2) = \log_2 \left(\frac{p(word_1 \wedge word_2)}{p(word_1)p(word_2)} \right)$$

- Semantic Orientation:

$$\text{SO}(phrase) = \text{PMI}(phrase, "excellent") - \text{PMI}(phrase, "poor")$$

Even Better Idea Turney 2001

- Pointwise Mutual Information (Church and Hanks, 1989):

$$\text{PMI}(word_1, word_2) = \log_2 \left(\frac{p(word_1 \wedge word_2)}{p(word_1)p(word_2)} \right)$$

- Semantic Orientation:

$$\text{SO}(phrase) = \text{PMI}(phrase, "excellent") - \text{PMI}(phrase, "poor")$$

- PMI-IR estimates PMI by issuing queries to a search engine

$$\text{SO}(phrase) = \log_2 \left(\frac{\text{hits}(phrase \text{ NEAR } "excellent")\text{hits}("poor")}{\text{hits}(phrase \text{ NEAR } "poor")\text{hits}("excellent")} \right)$$

Resources

These -- and related -- methods have been used to generate sentiment dictionaries

- Sentinet
- General Enquirer
- ...

Bottom-Up: Words to Clauses

- Assume we know the “polarity” of a word
- Does its context flip its polarity?

Prior Polarity versus Contextual Polarity

Wilson et al 2005

- **Prior polarity**: out of context, positive or negative
 - beautiful* → positive
 - horrid* → negative
- A word may appear in a phrase that expresses a different polarity in context

“Cheers to Timothy Whitfield for the **wonderfully horrid** visuals.”

Contextual polarity

Example

Philip Clap, President of the National Environment Trust, sums up well the general thrust of the reaction of environmental movements: there is no reason at all to believe that the polluters are suddenly going to become reasonable.

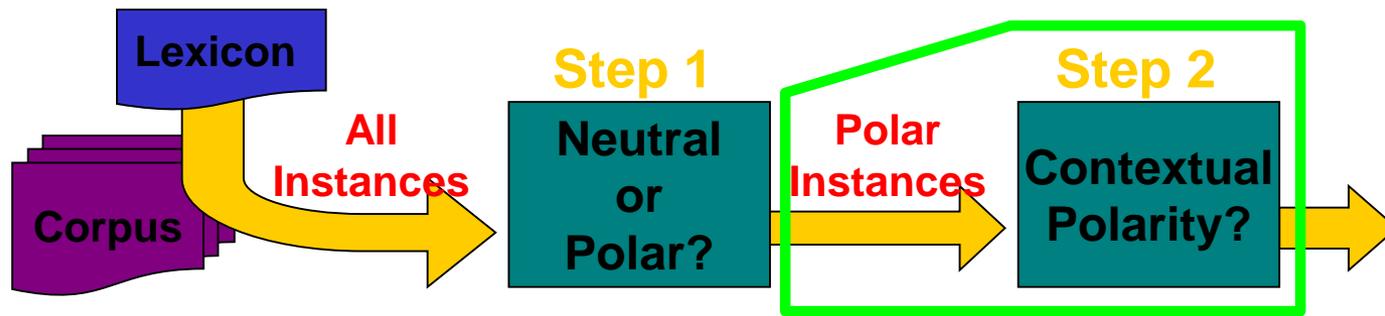
Example

Philip Clap, President of the National Environment **Trust**, sums up **well** the general thrust of the reaction of environmental movements: there is no **reason** at all to believe that the **polluters** are suddenly going to become **reasonable**.

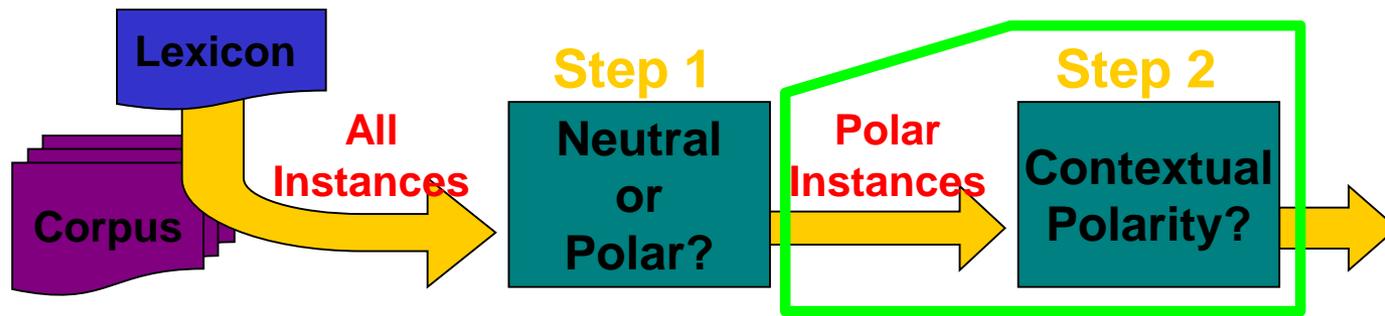
Example

Philip Clap, President of the National Environment ~~Trust~~, sums up well the general thrust of the reaction of environmental movements: there is no reason at all to believe that the ~~polluters~~ are suddenly going to become reasonable.





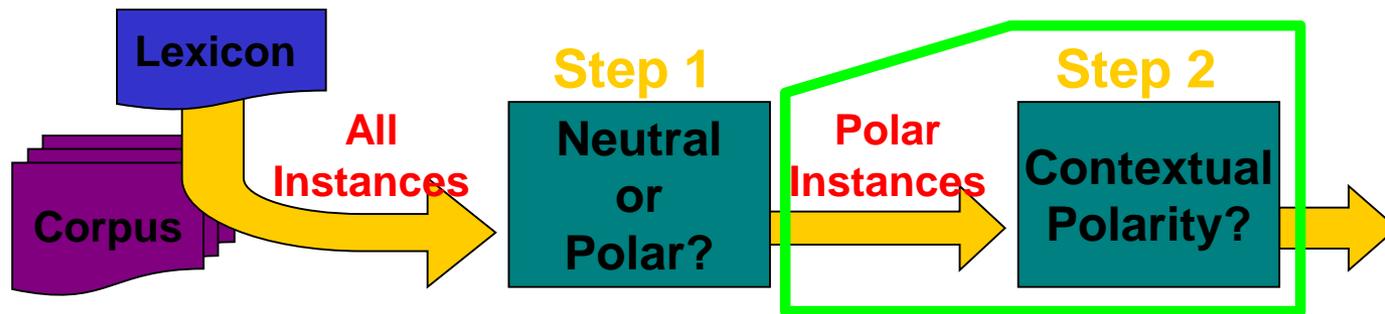
- Word token
- Word prior polarity
- Negated
- Negated subject
- Modifies polarity
- Modified by polarity
- Conjunction polarity
- General polarity shifter
- Negative polarity shifter
- Positive polarity shifter



- **Word token**
- **Word prior polarity**
- Negated
- Negated subject
- Modifies polarity
- Modified by polarity
- Conjunction polarity
- General polarity shifter
- Negative polarity shifter
- Positive polarity shifter

Word token
terrifies

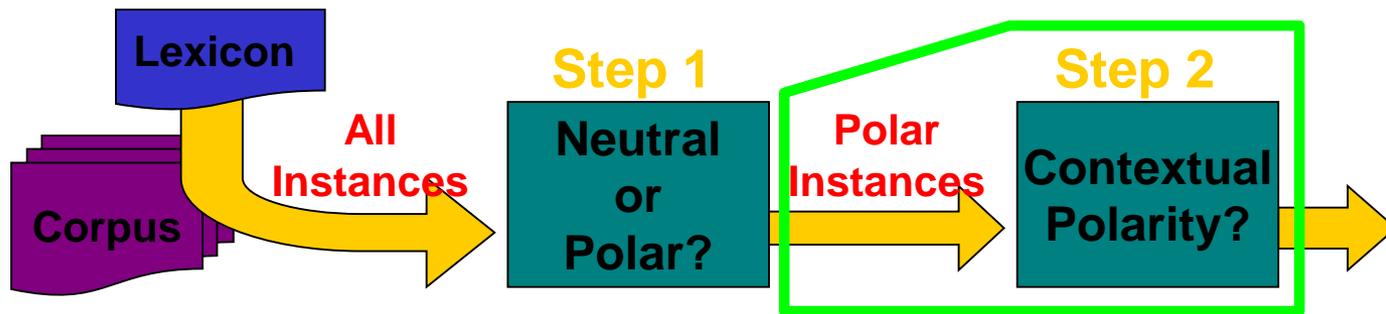
Word prior polarity
negative



- Word token
- Word prior polarity
- **Negated**
- **Negated subject**
- Modifies polarity
- Modified by polarity
- Conjunction polarity
- General polarity shifter
- Negative polarity shifter
- Positive polarity shifter

Binary features:

- **Negated**
For example:
 - *not good*
 - *does not look very good*
 - ❖ *not only good but amazing*
- **Negated subject**
No politically prudent Israeli could support either of them.

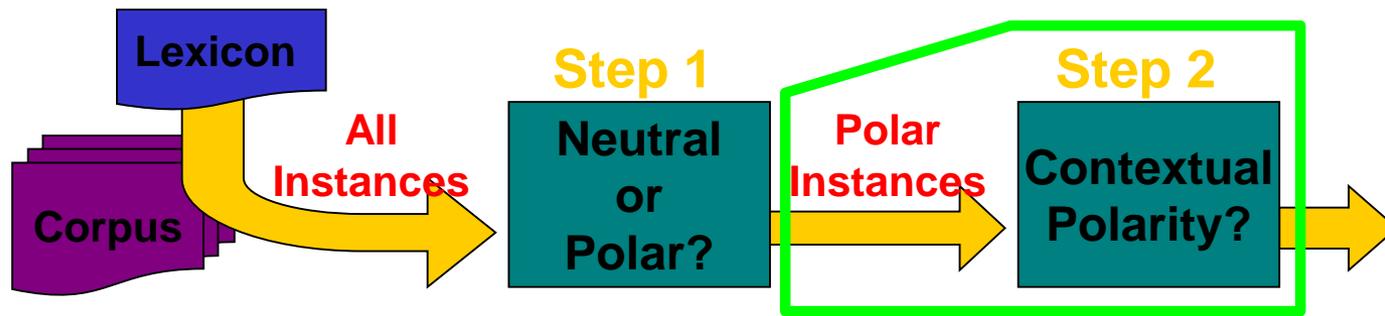


- Word token
- Word prior polarity
- Negated
- Negated subject
- **Modifies polarity**
- **Modified by polarity**
- Conjunction polarity
- General polarity shifter
- Negative polarity shifter
- Positive polarity shifter

- **Modifies polarity**
5 values: positive, negative, neutral, both, not mod
substantial: negative
- **Modified by polarity**
5 values: positive, negative, neutral, both, not mod
challenge: positive

substantial (pos) *challenge* (neg)





- Word token
- Word prior polarity
- Negated
- Negated subject
- Modifies polarity
- Modified by polarity
- **Conjunction polarity**
- General polarity shifter
- Negative polarity shifter
- Positive polarity shifter

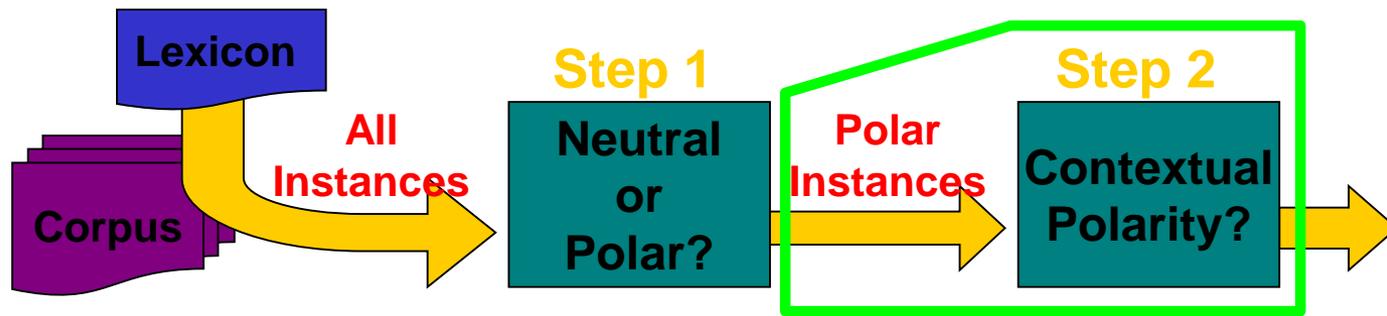
- **Conjunction polarity**

5 values: positive, negative, neutral, both, not mod

good: negative

good (pos) and *evil* (neg)





- Word token
- Word prior polarity
- Negated
- Negated subject
- Modifies polarity
- Modified by polarity
- Conjunction polarity
- **General polarity shifter**
- **Negative polarity shifter**
- **Positive polarity shifter**

- General polarity shifter

*pose **little** threat*

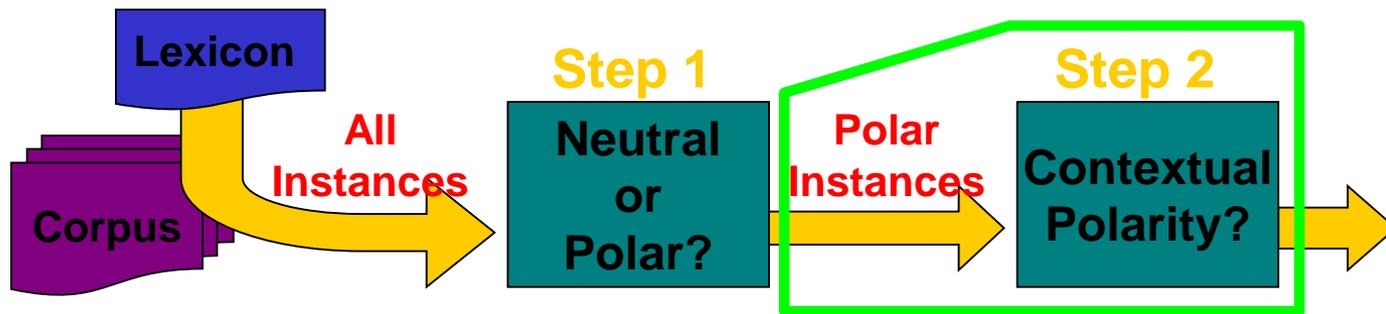
*contains **little** truth*

- Negative polarity shifter

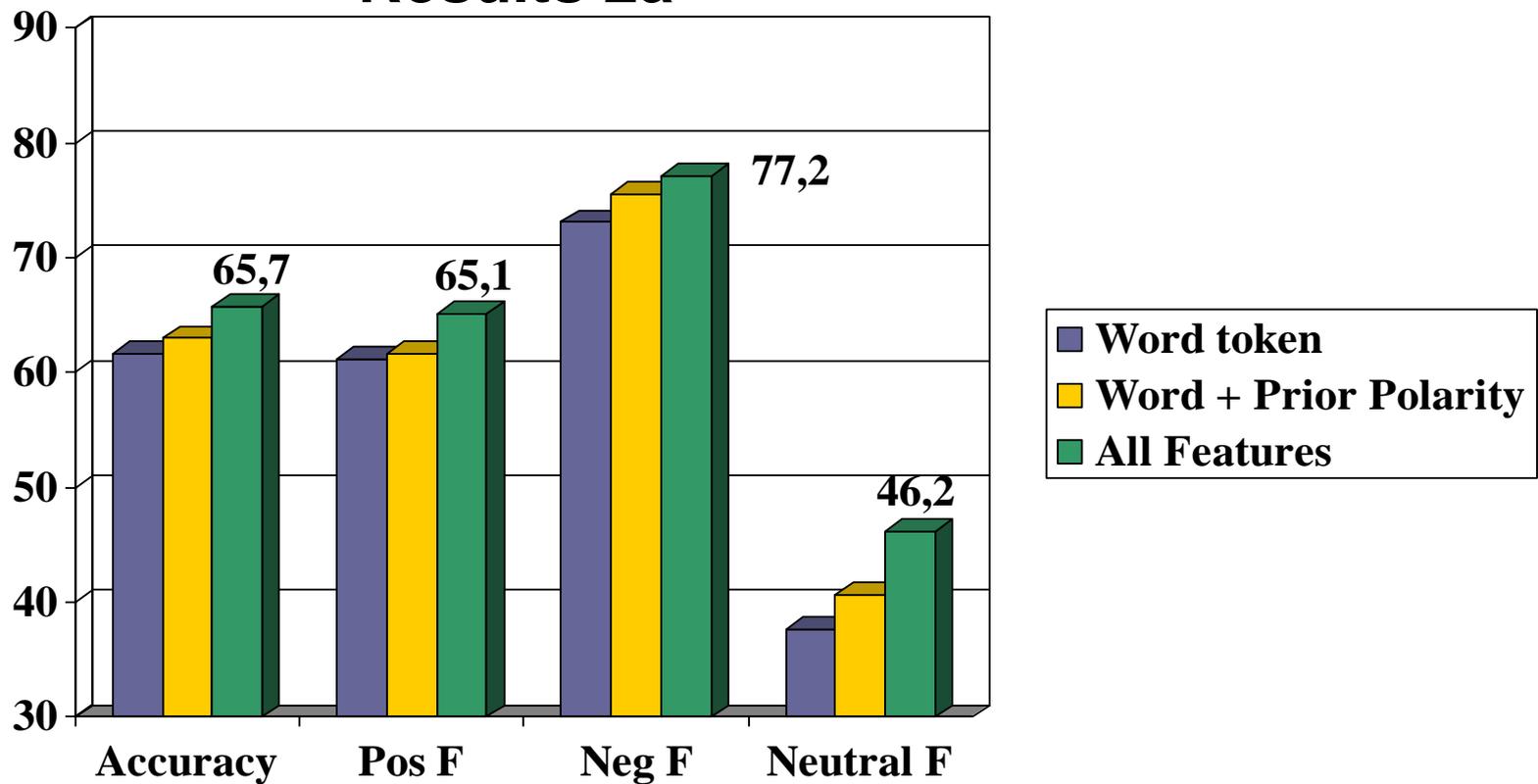
***lack** of understanding*

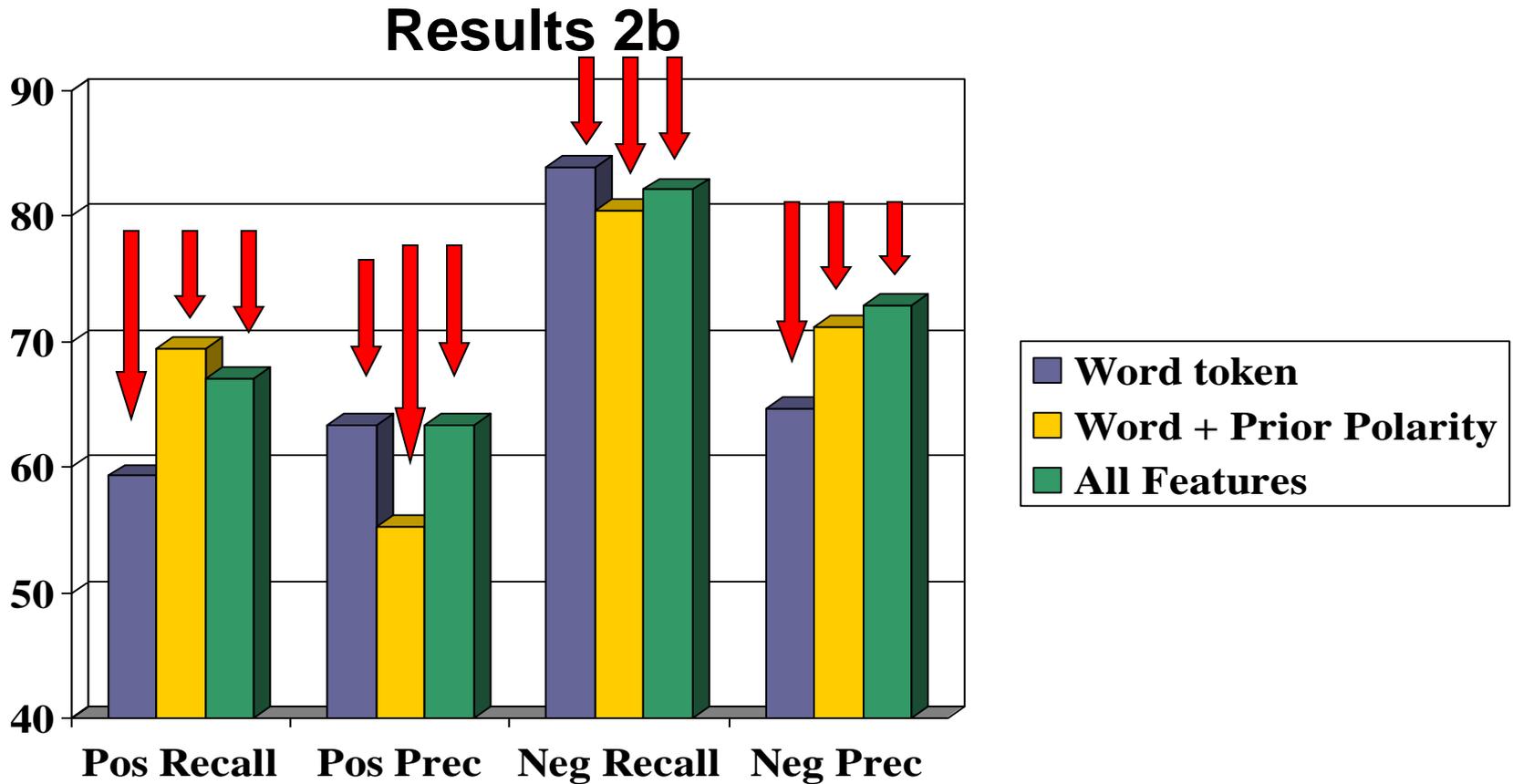
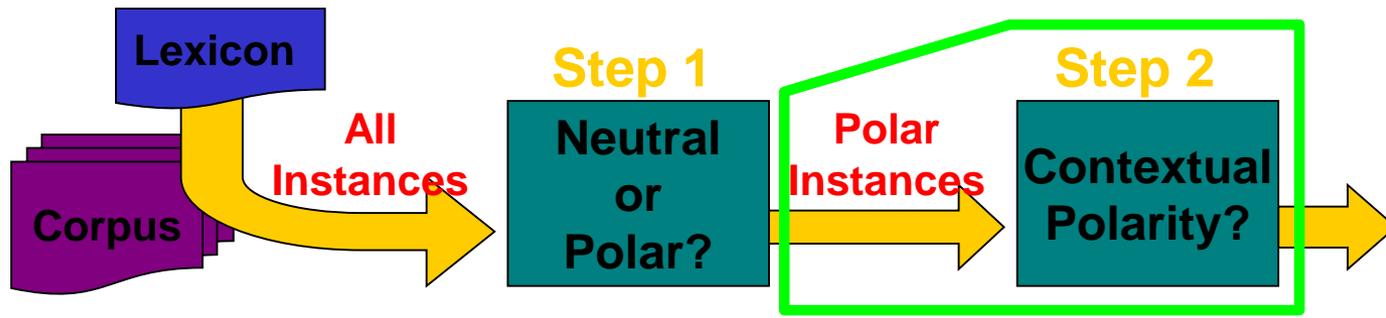
- Positive polarity shifter

***abate** the damage*



Results 2a





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

Finding Labeled Data

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

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

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)

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%.

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%+

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

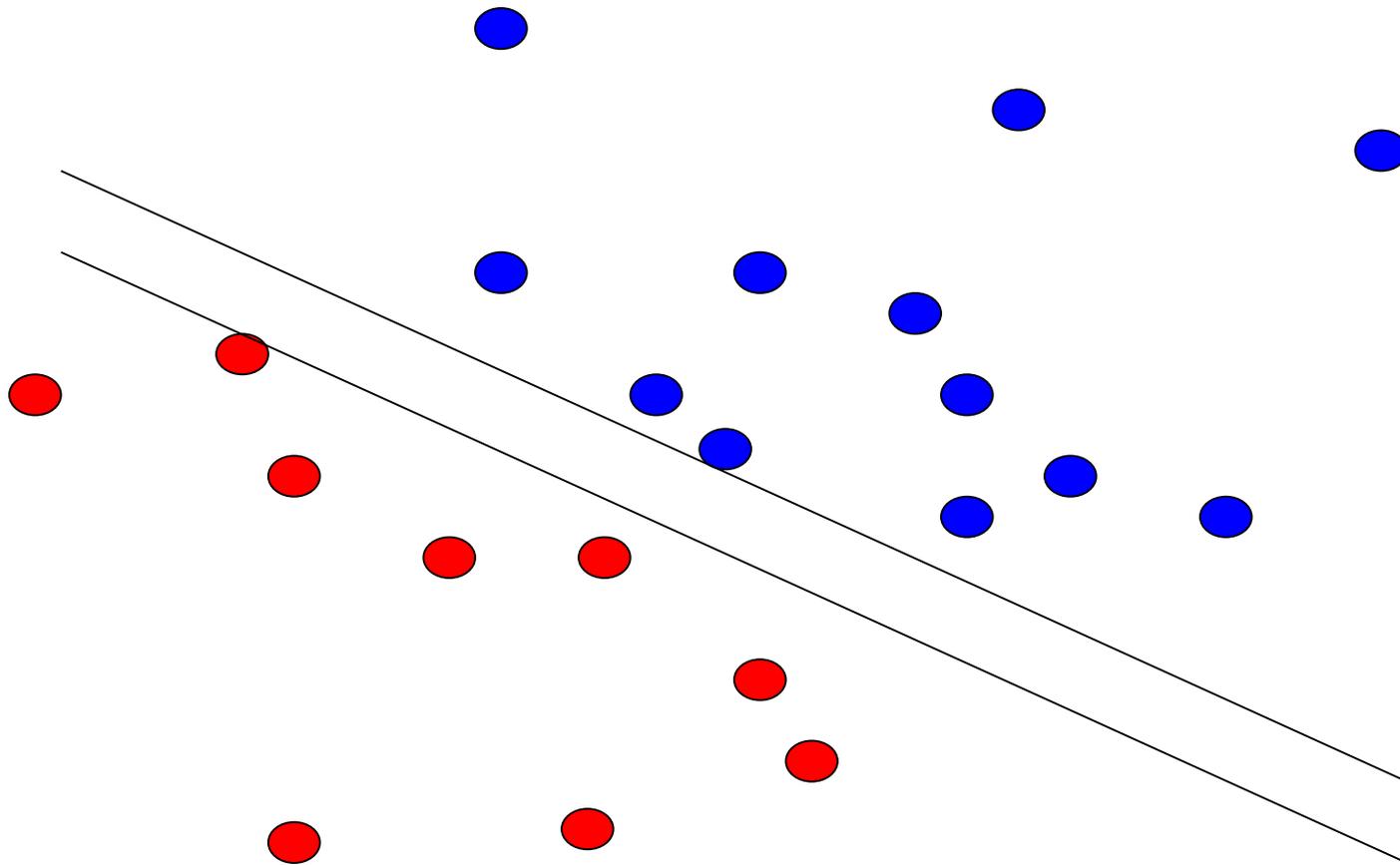
Other Genres

- 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

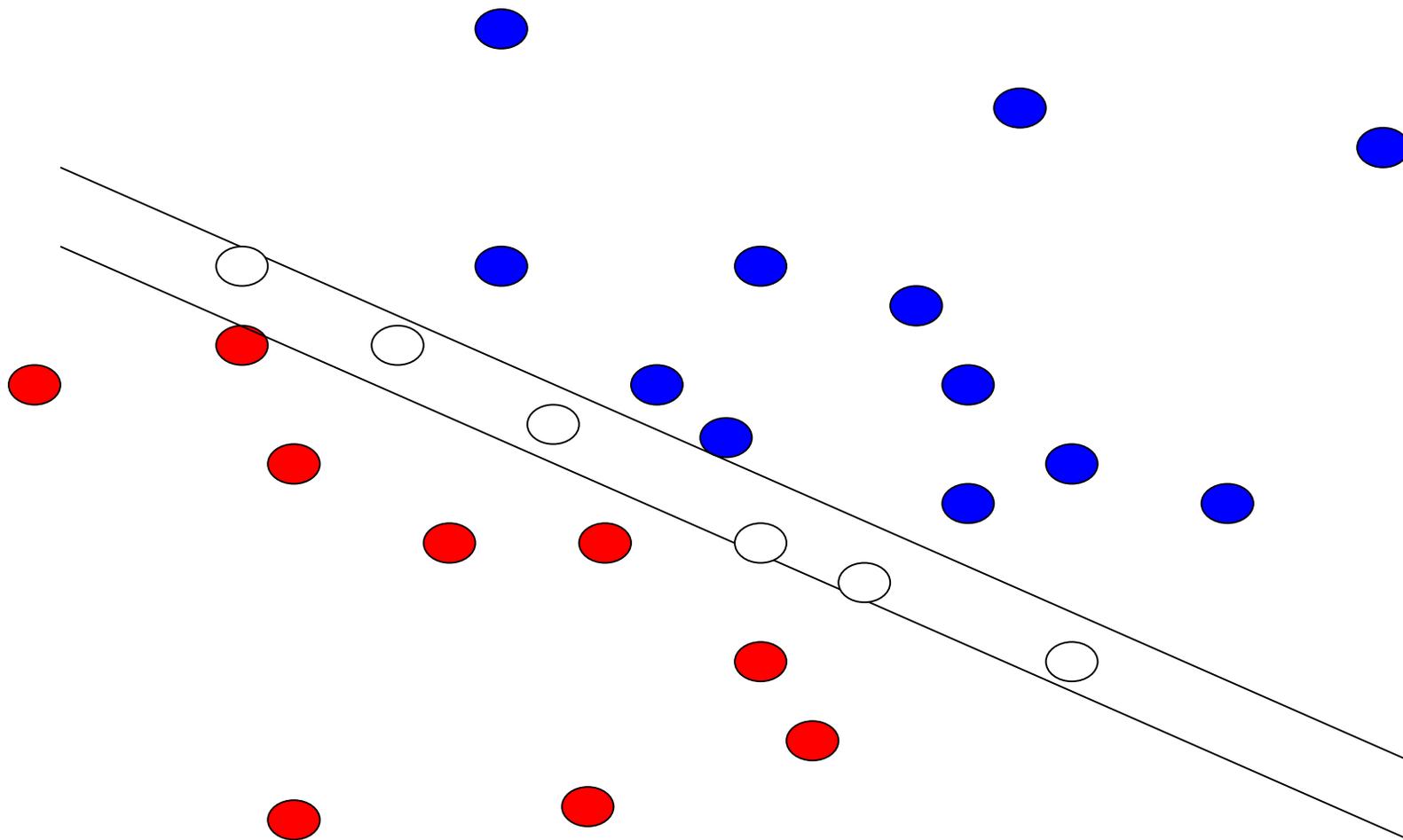
Cheating (Ignoring Neutrals)

- 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.

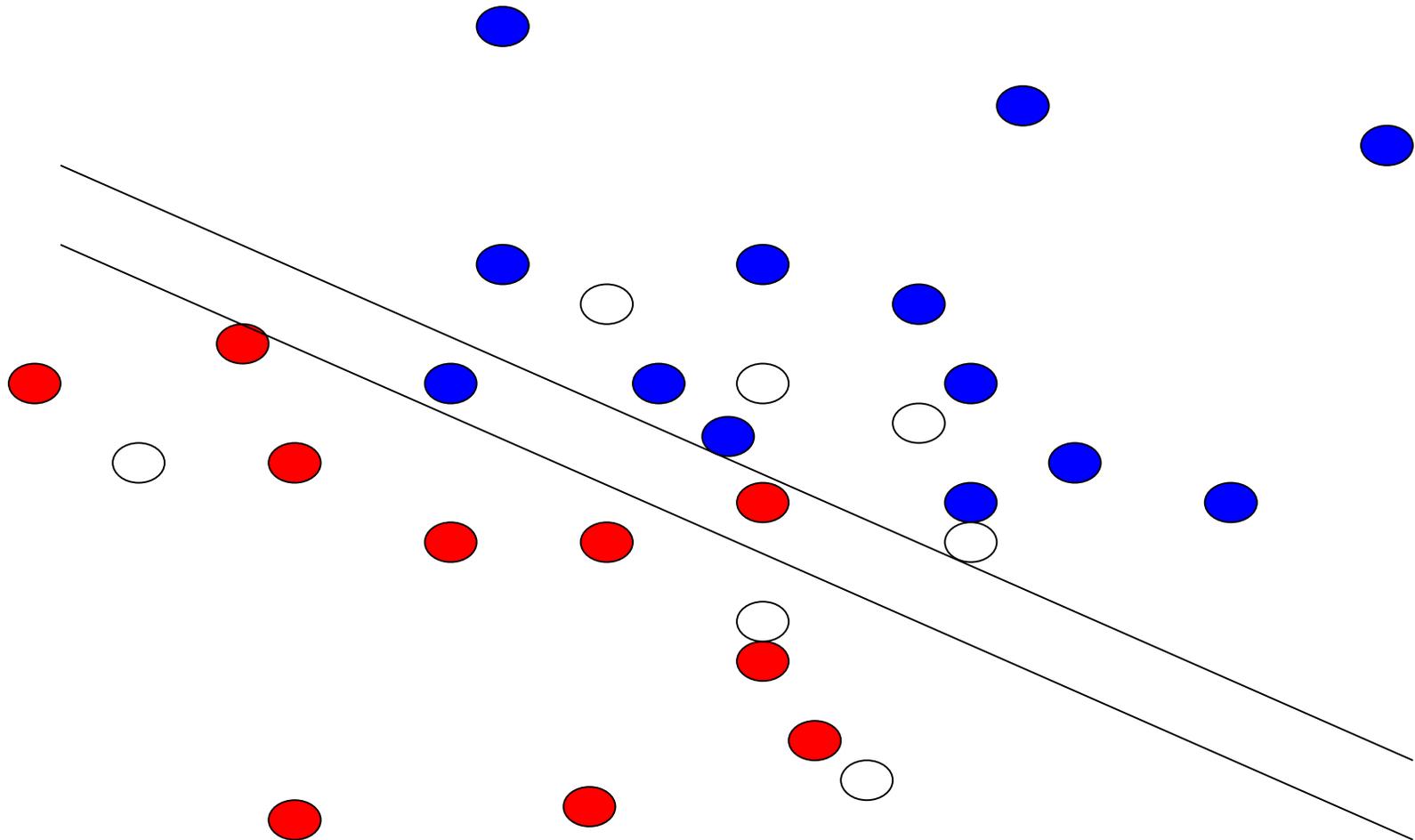
A Perfect World



A Perfect World



The Real World



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.

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).

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.

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).

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

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...

Optimal Stack

	Pos Vs	Pos Vs	Neut Vs	Actual category		
	Neg	Neut	neg	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

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%)

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.

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.)

Slide from Koppel/Pang/Gamon

Can Sentiment Analysis Make Me Rich?

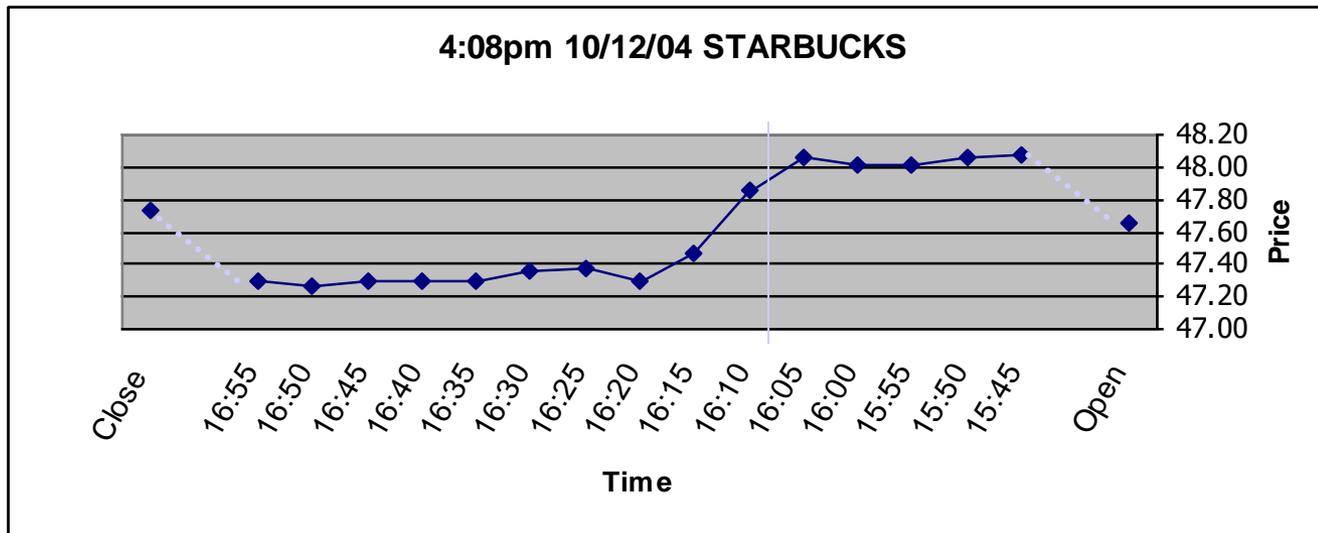
Can Sentiment Analysis Make Me Rich?

NEWSWIRE 4:08PM 10/12/04

STARBUCKS SAYS CEO ORIN SMITH TO RETIRE IN MARCH 2005

- How will this messages affect Starbucks stock prices?

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?

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

Experiment

- MSN corpus
 - 5000 headlines for 500 leading stocks
September 2004 – March 2005.
- Price data
 - Stock prices in 5 minute intervals

Feature set

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

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.

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.
- We exit when stock price moves Δ in either direction or after 40 minutes, whatever comes first.

Do we earn a profit?

Do we earn a profit?

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

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)

Machine learning

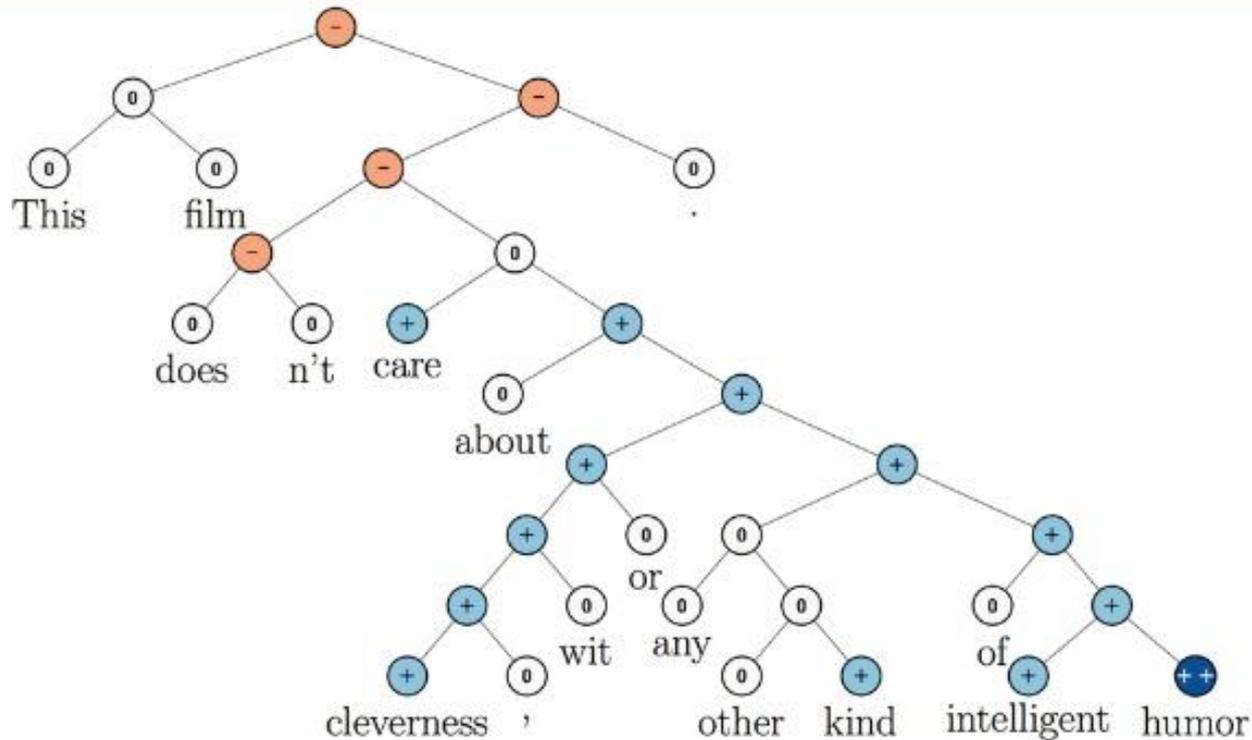
- Hand crafted features
 - In addition to unigrams: number of uppercase words, number of exclamation marks, number of positive and negative words ...
- In social media domain:
 - emoticons, hashtags (#happy), elongated words (haaaapy)

Deep learning

- Automatic feature extraction
 - Learn feature representation jointly
- Little to no preprocessing required
- General approaches:
 - Recursive Neural Networks
 - Convolutional Neural Networks
 - Recurrent Neural Networks

Recursive Neural Networks

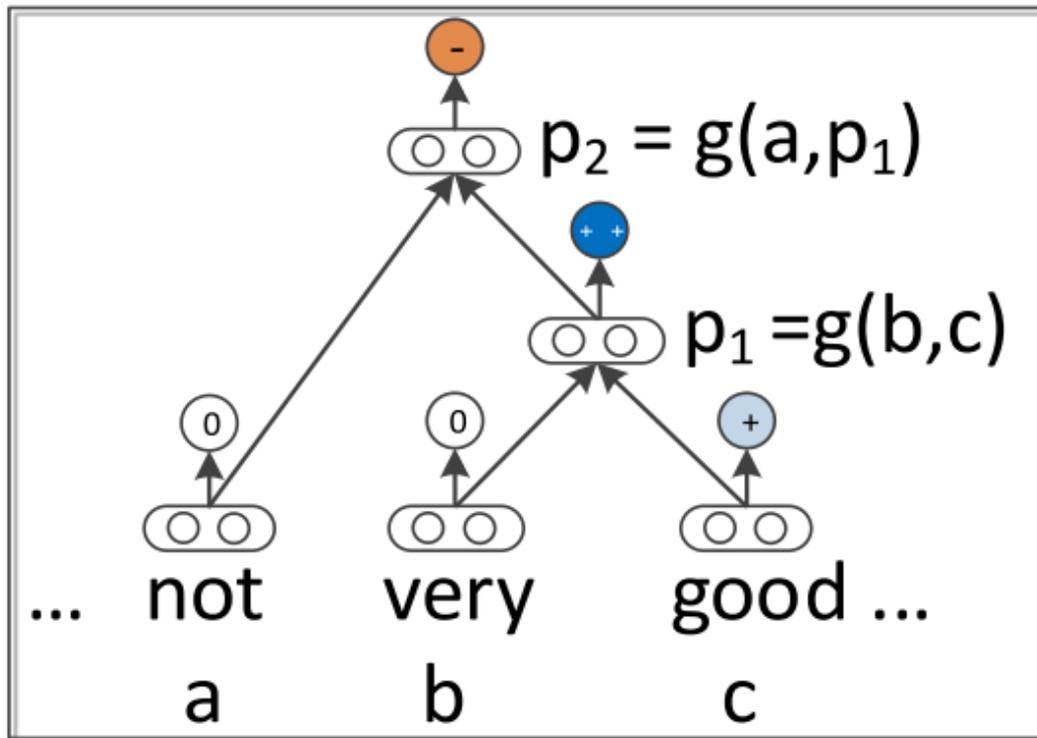
Recursive Deep Models & Sentiment: Socher (2013)



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 $10 \times d$

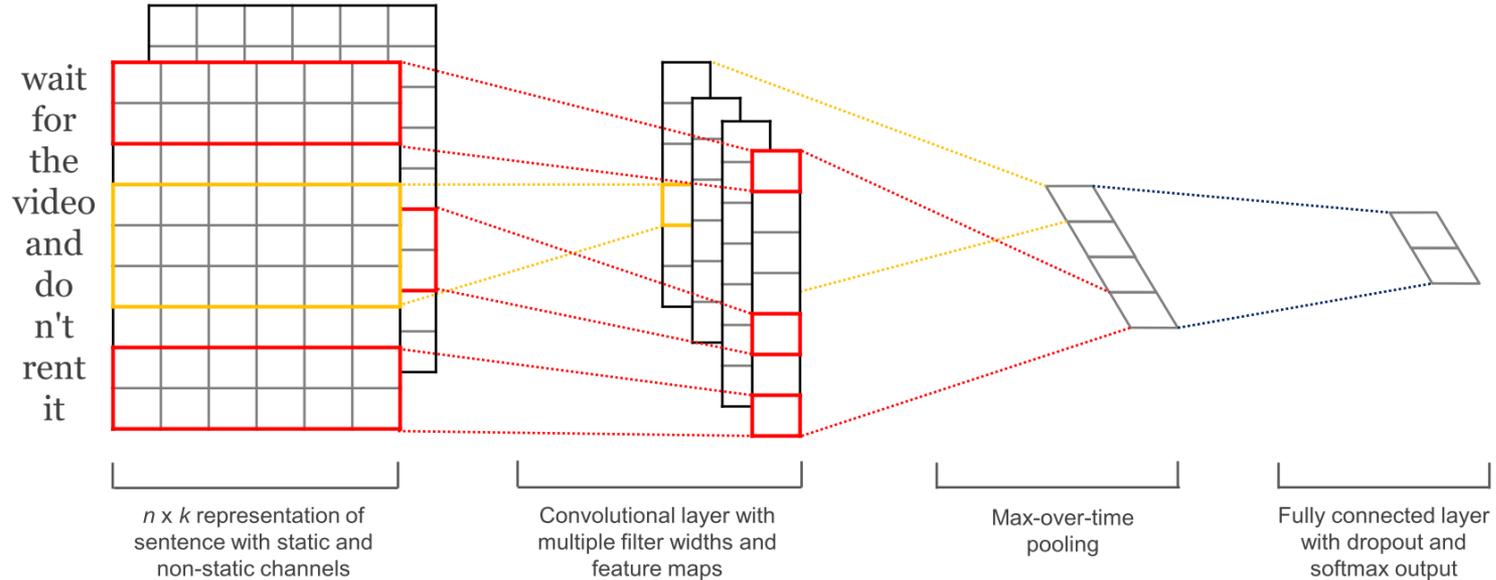
1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved Feature

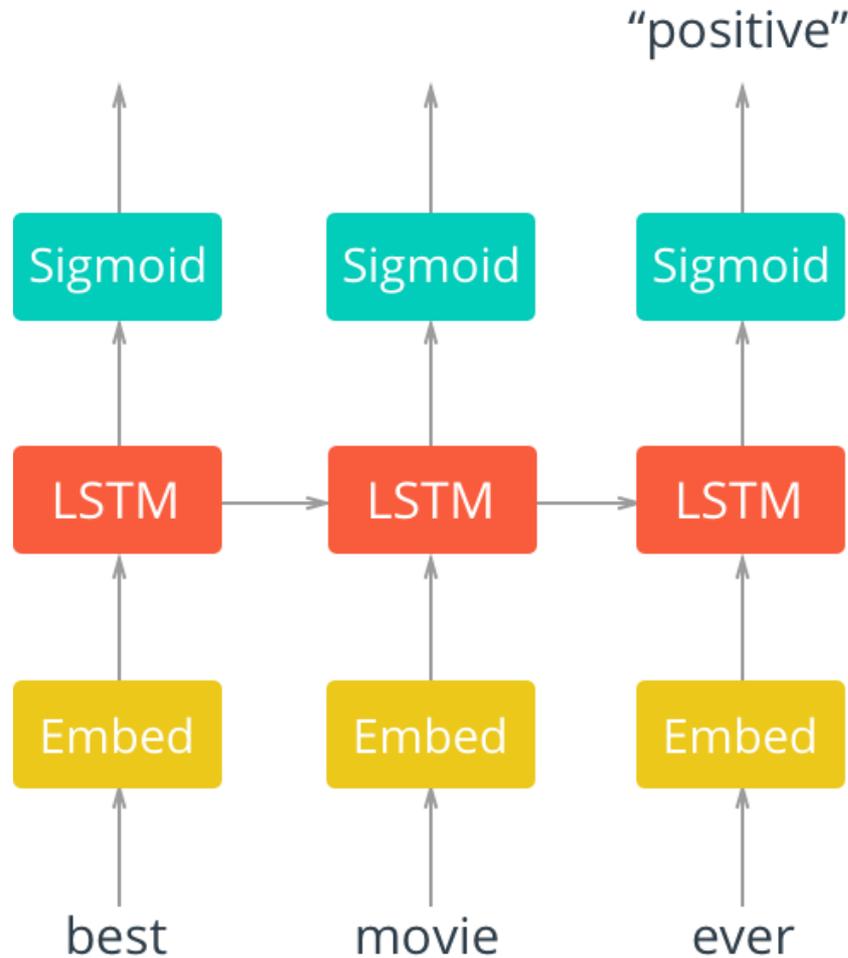
Convolutional Neural Networks



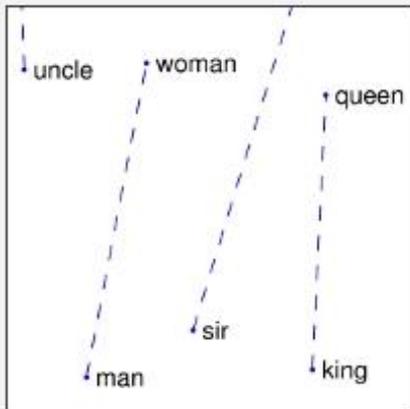
Feedforward Neural Networks

- Feedforward Neural Network with:
 - Bag of words
 - Averaged embeddings
- Feedforward Neural Network Language Model

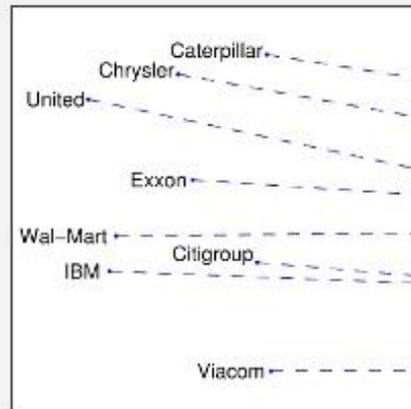
Recurrent Neural Networks



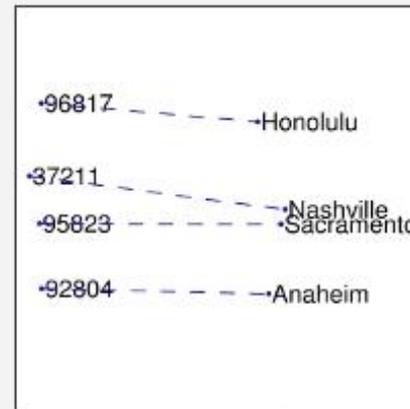
Word embeddings



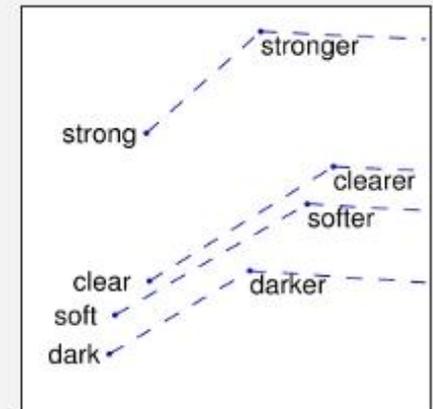
man - woman



company - ceo



city - zip code



comparative - superlative

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

Word embeddings

- Update word embeddings by back-propagation
- Most similar words before and after training

<i>bad</i>	<i>good</i> <i>terrible</i> <i>horrible</i> <i>lousy</i>	<i>terrible</i> <i>horrible</i> <i>lousy</i> <i>stupid</i>
<i>good</i>	<i>great</i> <i>bad</i> <i>terrific</i> <i>decent</i>	<i>nice</i> <i>decent</i> <i>solid</i> <i>terrific</i>

Sentiment Analysis in Keras

- Keras is a high-level neural networks API
- written in Python

```
model = Sequential()  
model.add(Embedding(vocab_size, emb_dim))  
model.add(LSTM(lstm_dim))  
model.add(Dense(3, activation="softmax"))
```

Sentiment neuron

- Character-level language model trained on Amazon reviews
- Linear model on this representation achieves state-of-the-art on Stanford Sentiment Treebank using 30-100x fewer labeled examples
- Representation contains a distinct “sentiment neuron” which contains almost all of the sentiment signal

Sentiment - Other Issues

- Somehow exploit NLP to improve accuracy
- Identify which specific product features sentiment refers to (fine-grained)
- “Transfer” sentiment classifiers from one domain to another (domain adaptation)
- Summarize individual reviews and also collections of reviews

- Slide sources
 - Nearly all of the slides today are from Prof. Moshe Koppel (Bar-Ilan University)
- Further reading on traditional sentiment approaches
 - 2011 AAIL 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." *arXiv preprint arXiv:1408.5882* (2014).
 - Socher, Perelygin, Wu, Chuang, Manning, Ng, Potts. **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).

- Thank you for your attention!