Sentiment Analysis

Ashwin ITTOO

26/2/2018

HEC Liège
Management School - University of Liège

About myself

Associate Prof. HEC Liège, ULiège

Research Associate, JAIST

Associate Editor, Elsevier (Computers in Industry)

Data Science Liège

PhD Team

Marc Jamoulle, MD, PhD (2014-2018)

- *Machine learning/NLP over medical abstracts*
- *OWL Ontologies (semantic web)*

Anne-Sophie Hoffait (2015-Present)

- *Predicting customer review helpfulness*
- *LASSO, RIDGE Regressions*

Cédric Gillain (2016-Present)

- *Detecting financial recommendations from texts*
- *Impact on financial markets*

Projects in the making...

Reinforcement Learning

- *Algorithmic Collusion*
- *Cooperation, Competition in MARL systems*

NLP (EU)

- *Processing of financial regulations*

General machine learning/stats (Multi-national)

- *Compliance, anti-trust detection*

- *IoT (large local company)*

Sentiment Analysis: Intro

- General class of **Affective Computing** problems

- Affective types of Scherer (2010)

Emotion: Relatively brief episode of response to the evaluation of an external or internal event as being of major significance.
(*angry, sad, joyful, fearful, ashamed, proud, elated, desperate*)

Mood: Diffuse affect state, most pronounced as change in subjective feeling, of low intensity but relatively long duration, often without apparent cause.
(*cheerful, gloomy, irritable, listless, depressed, buoyant*)

Interpersonal stance: Affective stance taken toward another person in a specific interaction, colouring the interpersonal exchange in that situation.
(*distant, cold, warm, supportive, contemptuous, friendly*)

sentiment

Attitude: Relatively enduring, affectively colored beliefs, preferences, and predispositions towards objects or persons.
(*liking, loving, hating, valuing, desiring*)

Personality traits: Emotionally laden, stable personality dispositions and behavior tendencies, typical for a person.
(*nervous, anxious, reckless, morose, hostile, jealous*)

Sentiment Analysis: Intro (cont)

- Known interchangeably as **Opinion Mining**
- Vast research area, many sub-fields/sub-tasks
  - *Opinion Mining/Sentiment Analysis*
  - *Document Sentiment Classification*
  - *Subjectivity Classification*
  - *Aspect-based Sentiment Analysis*
  - *Mining Comparative Opinions*
  - *Opinion Lexicon Generation*
  - *...*

Sentiment Analysis: Intro (cont)

- Example

> **Id: Abc123 on 5-1-2008** *"I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …"*

- Several associated dimensions
  - Opinion (sentiment) holder
  - Sentiment target
  - Sentiment orientation (positive, negative, neutal)
  - Time

Sentiment Analysis: Intro (cont)

- 2 main types
- 1. Regular opinions
  - Sentiment expressed on some target
  - Further sub-divided as direct, indirect
  - E.g. direct: “this phone is cool”
  - E.g. indirect: “the pill has relieved my headache”

- 2. Comparative opinions
  - Comparisons between targets
  - “Iphone is better than Samsung”

Formalizing an Opinion

- A quintuple (5-tuple)

$$(e_j,\ a_{jk},\ so_{ijkl},\ h_i,\ t_l),$$

- Where

e_j is a target entity.

a_{jk} is an aspect/feature of the entity e_j.

so_{ijkl} is the sentiment value of the opinion from the opinion holder h_i on feature a_{jk} of entity e_j at time t_l. so_{ijkl} is +ve, -ve, or neu, or more granular ratings.

h_i is an opinion holder.

t_l is the time when the opinion is expressed.

Formalizing an Opinion (cont)

- Additional entries in quintuple possible
  - Age, gender, ..

- Can lead to “overspecification”
- Ignored in practice
- But useful for generating **opinion summaries**

Opinion Summaries

- From Hu & Liu, 2004

"“I bought an iPhone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is clear too. It is much better than my old Blackberry, which was a terrible phone and so difficult to type with its tiny keys. However, my mother was mad with me as I did not tell her before I bought the phone. She also thought the phone was too expensive, …”

Feature Based Summary of iPhone:

Feature1: **Touch screen**

Positive: 212

- *The touch screen was really cool.*
- *The touch screen was so easy to use and can do amazing things.*

…

Negative: 6

- The screen is easily scratched.
- I have a lot of difficulty in removing finger marks from the touch screen.

…

Feature2: **voice quality**

Document Level Sentiment Analysis

- Determine **overall** opinion of **entire document**

- Documents
  - Tweets
  - Online Reviews
  - Blogs, discussion forums
  - News articles

easy

More difficult

- Determining overall opinion
  - Challenging
  - Reformulated as a **text categorization problem**

Document Level Sentiment Analysis (cont)

- Given a corpus of documents
- Classify into 2 (or 3) classes
  1. Positive
  2. Negative
  3. (Neutral)

- Example

> *"I bought an iPhone a few days ago. It is such a nice phone, although a little large. The touch screen is cool. The voice quality is clear too. I simply love it!"*

- Classification: pos or neg?
- Various approaches
  - Supervised (incl. RNN)
  - Semi-supervised (lexicon-based)

Document Level Sentiment Analysis (cont)

- Classical approach of Turney (2002)
- Semi-supervised
- Data from eopinions.com
- Step 1:
  - Find Parts-of-Speech (POS) of words (more later)

	First word	Second word	Third word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Document Level Sentiment Analysis (cont)

- Step 2:
  - Estimate the sentiment of the selected phrases
  - How? Given phrase *p*
  - Measure the **association strengths** between
    - p and positive phrase, e.g. “excellent” (x)
    - p and negative phrase, e.g. “poor” (y)
    - Sentiment = x – y
- Step 3:
  - Average sentiment for phrases/document

Document Level Sentiment Analysis (cont)

- **Association strength** between phrases/words
- Turney (2002) *Pointwise Mutual Info. (PMI)*

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

- Chi-squared

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

  – H0: Assume words are not associated)
- Word counts/MLE estimates (probabilities) from corpus or search engine hits

Sentiment Classification

- Supervised approach
- Different types of classifiers
  - SVM
  - Neural networks
  - Logistic regression (maxent)
  - Naïve-Bayes
- Naïve-Bayes example
  - Reasonably good performance (strong baseline)
  - Straightforward (to work out)

Sentiment Classification (cont)

- Naïve-Bayes
  - Generative model (joint probability)
  - Constructs a model of each class
  - Given an observation, *x*
    - Return class likely to have generated *x*

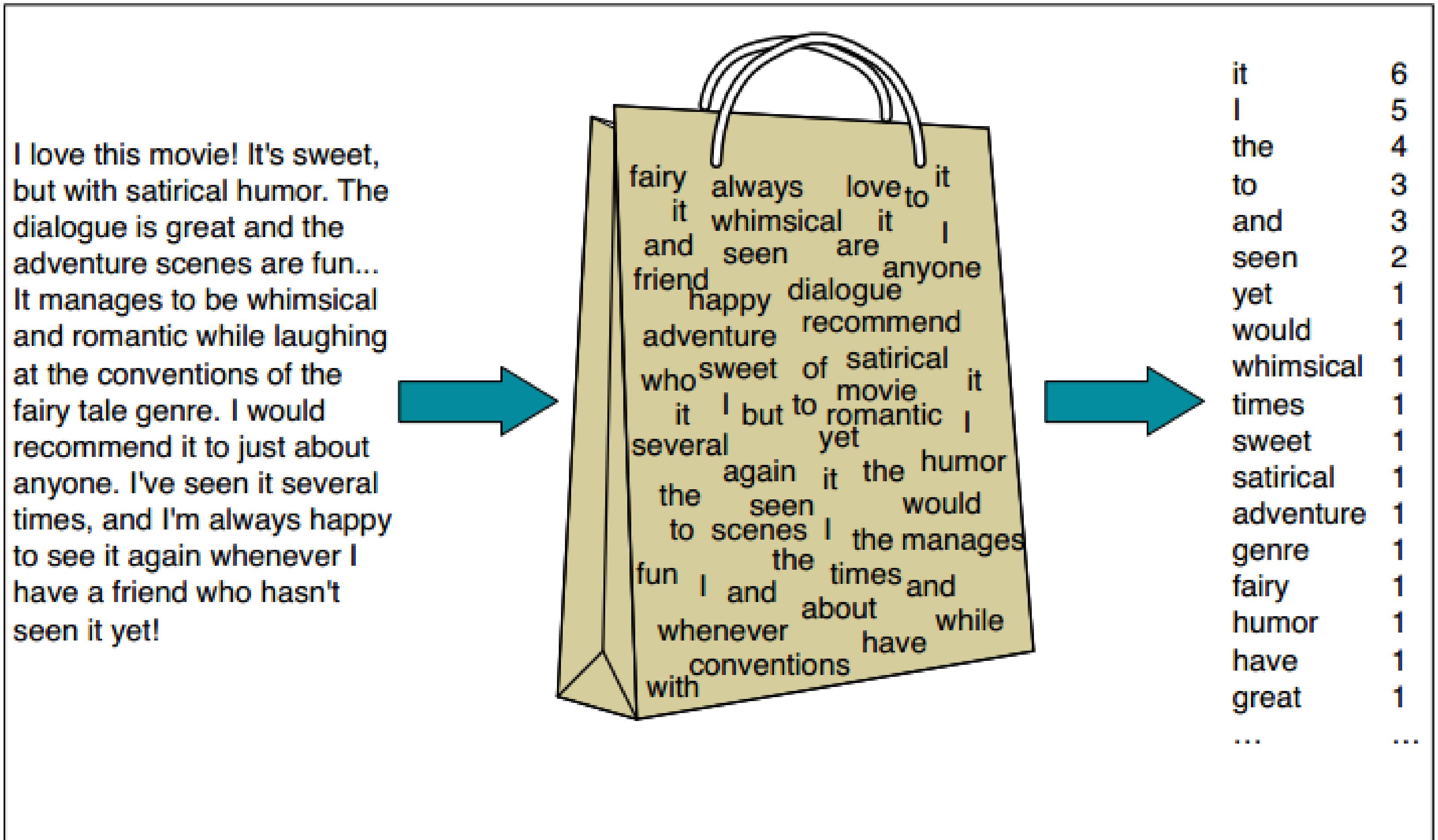
- MaxEnt
  - Discriminative model (conditional probability)
  - Learns discriminative features

Naïve-Bayes Sentiment Classification

- Multinomial Naïve-Bayes (NB)
  - Word counts
- Bernoulli Naïve-Bayes
  - Absence/presence of words (binary)
- Multinomial works better
- Bag-of-Word (BOW) model
  - Document treated a collection of distinct words
  - Syntactic & semantic relations between words ignored
  - Works well for some problems in practice

Naïve-Bayes Sentiment Classification

- BOW
  - Each word represented as a vector dimension
  - Each vector element = word freq (or tf-idf)



Naïve-Bayes Sentiment Classification (cont)

- Tf-idf
  - Term frequency inverse document frequency
  - Measures importance of words in corpus
  - More frequent $\not\Leftrightarrow$ more important
  - More useful that raw frequency counts

$$tf - idf_{t,d} = tf_{t,d} \times idf_t = tf_{t,d} \times log \frac{N}{df_t}$$

Naïve-Bayes Sentiment Classification (cont)

- NB is a probabilistic classifier
- Given document *d*
- Classes $c \in C$
- Return that class $\hat{c}$
  - Maximum posterior probability given *d*

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d)$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

Naïve-Bayes Sentiment Classification (cont)

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} \frac{P(d|c)P(c)}{P(d)}$$

- Drop *p(d)*. **Why?**

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} P(c|d) = \underset{c \in C}{\operatorname{argmax}} P(d|c)P(c)$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

Naïve-Bayes Sentiment Classification (cont)

- d = collection of features (e.g. words), f

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(d|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}} \overbrace{P(f_1, f_2, \ldots., f_n|c)}^{\text{likelihood}} \overbrace{P(c)}^{\text{prior}}$$

- Hard to compute
  - Likelihood term expands recursively
  - Requires estimating every feature combination
  - E.g. each word occurring at each position
  - Too many parameters to estimate

Naïve-Bayes Sentiment Classification (cont)

- NB makes 2 "naïve" assumptions

1. Word position is irrelevant (BOW model)
2. Conditional independence assumption
   - $P(f_i|c)$ are independent given class

$$P(f_1, f_2, \ldots, f_n|c) = P(f_1|c) \cdot P(f_2|c) \cdot \ldots \cdot P(f_n|c)$$

- Class prediction by NB classifier

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}}\, P(c) \prod_{f \in F} P(f|c)$$

$$c_{NB} = \underset{c \in C}{\operatorname{argmax}} \log P(c) + \sum_{i \in positions} \log P(w_i|c)$$

NB Training

- Estimating $P(c)$ and $P(f_i|c)$

docs of class c

of docs (in total)

$$\hat{P}(c) = \frac{N_c}{N_{doc}}$$

$P(f_i|c)$.

  - Fraction of times feature f_i (or word w_i) appears in all docs of class C

$$\hat{P}(w_i|c) = \frac{count(w_i,c)}{\sum_{w \in V} count(w,c)}$$

  - Add 1 to prevent 0 probabilities (Laplace smoothing). **Why?**

$$\hat{P}(w_i|c) = \frac{count(w_i,c)+1}{\sum_{w \in V}(count(w,c)+1)} = \frac{count(w_i,c)+1}{\left(\sum_{w \in V} count(w,c)\right)+|V|}$$

HEC Liège Management School - University of Liège

NB Example (Sentiment Classification)

- Training corpus

	Cat	Documents
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

$$P(-) = \frac{3}{5} \qquad P(+) = \frac{2}{5}$$

- "with" does not occur in any class → dropped

NB Example (Sentiment Classification)

- Compute probability of test words appearing in + or - class

$$P(\text{``predictable''}|-) = \frac{1+1}{14+20} \qquad P(\text{``predictable''}|+) = \frac{0+1}{9+20}$$

$$P(\text{``no''}|-) = \frac{1+1}{14+20} \qquad P(\text{``no''}|+) = \frac{0+1}{9+20}$$

$$P(\text{``fun''}|-) = \frac{0+1}{14+20} \qquad P(\text{``fun''}|+) = \frac{1+1}{9+20}$$

- Calculate max. posterior probability for + and – class

$$P(-)P(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$P(+)P(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Predicted class: - (negative)

NB Example (cont)

- NB algorithm (training and testing phases)

function TRAIN NAIVE BAYES(D, C) **returns** log $P(c)$ and log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $$loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$$
return *logprior*, *loglikelihood*, V

function TEST NAIVE BAYES(*testdoc*, *logprior*, *loglikelihood*, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in *testdoc*
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c \; sum[c]$

Slightly Improving NB: Shifters

- Sentiment shifters (valence shifters)
  - Words, phrases shift sentiment (e.g. from pos to neg)
  - Known as polarity inverters

- Most common polarity inverters
  - Negation words
  - Not, never, don't, can't, …

- Other sentiment shifters
  - Modal auxiliary verbs: would, should, could
  - Presuppositional markers: barely, hardly ("hardly works)

Dealing with Negation/Polarity Inverters

- 1. Syntactic parsing
  - Find items in a sentence syntactically related to negation (more later)
  - Syntactic parsing is computationally expensive, challenging and can be brittle
  - Not applicable to social media texts (e.g. tweets), syntactically ill-formed

- 2. Prefix every word following negation with a negation marker
  - E.g. NOT_<followed by word>

```
didn't like this movie , but I
```

```
didn't NOT_like NOT_this NOT_movie , but I
```

- Newly formed words likelier to appear in “negative” class documents

Sentiment Classification: Lexicon-Based Approaches

- Recall goal
  - Classify document according to sentiment categories (+, -)
- Approaches so far:
  - Semi-supervised, Turney, 2002
  - Standard supervised learning/classification approach (Naïve-Bayes or any other classifier)

- Lexicon-based approaches
  - Relatively straightforward
  - Involves use of sentiment lexicons (dictionaries)

Lexicon-Based Approaches (cont)

- Input
  - Sentences, s, from document d
  - Positive lexicon, Negative lexicon (pos_lex and neg_lex)

- Output
  - Sentiment class: +, - (or neutral)

- Steps
  - Splits s on “but” words (“but”, “except that”, “however”)
  - Indicative of polarity change
  - For each $w \in s$
    - If $w \in$ *pos_lex, pos_score ++*
    - *Else if* $w \in$ *neg_lex, neg_score ++*
  - S_score = pos_score – neg_score
  - Average for all s ∈ d

HEC Liège
Management School - University of Liège

Lexicon-Based Approaches

- Input
  - Sentences, s, from document d
  - Positive lexicon, Negative lexicon (pos_lex and neg_lex)

- Output
  - Sentiment class: +, - (or neutral)

- Steps
  - Splits s on “but” words (“but”, “except that”, “however”)
  - Indicative of polarity change
  - For each $w \in s$
    - If $w \in$ *pos_lex, pos_score ++*
    - *Else if* $w \in$ *neg_lex, neg_score ++*
  - S_score = pos_score – neg_score
  - Average for all s ∈ d

Lexicons and Other Resources

- General Inquirer
  - Draws on early studies on cognition psychology of word meaning
  - http://www.wjh.harvard.edu/~inquirer/
- MPQA Subjectivity Lexicon
  - List of positive and negative words drawn from various sources
  - http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
- Liu Bing's Lexicon
  - List of positive and negative words for product reviewss
  - http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
- SentiWordNet
  - http://sentiwordnet.isti.cnr.it/

Semi-supervised Learning of Lexicons

- Turney's approach (2002)
  – Presented earlier
  – Discussed again (for revision)

- Neighborhood co-occurrence (collocation) as proxy for polarity similarity
  – If w1 and w2 are associated frequently → they share the same polarity
  – PMI for measuring association (collocation strength)

- Select parts-of-speech patterns/phrases

	First word	Second word	Third word (Not Extracted)
1.	JJ	NN or NNS	anything
2.	RB, RBR, or RBS	JJ	not NN nor NNS
3.	JJ	JJ	not NN nor NNS
4.	NN or NNS	JJ	not NN nor NNS
5.	RB, RBR, or RBS	VB, VBD, VBN, or VBG	anything

Semi-supervised Learning of Lexicons (cont)

- Choose positive and negative seeds (pos_seed, neg_seed)
  - Words that always express positive , negative sentiments
  - “Excellent”, “poor” as seeds

- Measure association of selected patterns with pos_seed, neg_seed
- PMI measure

$$PMI(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- Counts/probabilities estimated from search engine hits
- Estimate PMI as

k = distance between phrases

$$\mathrm{PMI}(w,s) = \log_2 \frac{\frac{1}{kN}\,\mathrm{hits}(w\ \mathrm{NEAR}\ s)}{\frac{1}{N}\,\mathrm{hits}(w)\,\frac{1}{N}\,\mathrm{hits}(s)}$$

Semi-supervised Learning of Lexicons (cont)

- For a given pattern, *w*

$$\text{Polarity}(w) = \text{PMI}(w, \text{“excellent”}) - \text{PMI}(w, \text{“poor”})$$

- Sample results

Extracted Phrase	Polarity
online experience	2.3
very handy	1.4
low fees	0.3
inconveniently located	-1.5
other problems	-2.8
unethical practices	-8.5

Semi-supervised Learning of Lexicons (cont)

- WordNet-based approaches
  - Synonyms share polarity
  - Antonyms have opposite polarity

- WordNet (http://wordnetweb.princeton.edu/perl/webwn)
  - Lexico-semantic dictionary
  - Words and semantic relations (e.g. synonymy, part-of)

- WordNet extended to SentiWordNet (http://sentiwordnet.isti.cnr.it/)

Synset	Pos	Neg	Obj
good#6 'agreeable or pleasing'	1	0	0
respectable#2 honorable#4 good#4 estimable#2 'deserving of esteem'	0.75	0	0.25
estimable#3 computable#1 'may be computed or estimated'	0	0	1
sting#1 burn#4 bite#2 'cause a sharp or stinging pain'	0	0.875	.125
acute#6 'of critical importance and consequence'	0.625	0.125	.250
acute#4 'of an angle; less than 90 degrees'	0	0	1
acute#1 'having or experiencing a rapid onset and short but severe course'	0	0.5	0.5

Semi-supervised Learning of Lexicons (cont)

- WordNet-based approaches
  - Synonyms share polarity
  - Antonyms have opposite polarity

- WordNet (http://wordnetweb.princeton.edu/perl/webwn)
  - Lexico-semantic dictionary
  - Words and semantic relations (e.g. synonymy, part-of)

- WordNet extended to SentiWordNet (http://sentiwordnet.isti.cnr.it/)

Synset	Pos	Neg	Obj
good#6 ‘agreeable or pleasing’	1	0	0
respectable#2 honorable#4 good#4 estimable#2 ‘deserving of esteem’	0.75	0	0.25
estimable#3 computable#1 ‘may be computed or estimated’	0	0	1
sting#1 burn#4 bite#2 ‘cause a sharp or stinging pain’	0	0.875	.125
acute#6 ‘of critical importance and consequence’	0.625	0.125	.250
acute#4 ‘of an angle; less than 90 degrees’	0	0	1
acute#1 ‘having or experiencing a rapid onset and short but severe course’	0	0.5	0.5

State-of-the-Art Sentiment Classification

- Longstanding challenge
  - Capturing semantic compositionality
  - E.g. each clause of a long sentence can express a different sentiment
  - Need to “compose” individual sentiments
  - Generate overall sentiment

This movie doesn't care about cleverness, wit or any other kind of intelligent humor. Those who find ugly meanings in beautiful things are corrupt without being charming.

State-of-the-Art Sentiment Classification (cont)

- Sentiment Treebank
  - Fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences
- RNN trained over sentiment treebank
  - capture contrastive conjunctions, negation and scope
- Read more at https://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf
- Java demo
- Visual demo

State-of-the-Art Sentiment Classification (cont)

- *Learning to Generate Reviews and Discovering Sentiment*
  - *Alec Radford, Rafal Jozefowicz, Ilya Sutskever*

- Touches upon many novel, interesting topics of machine learning

- Unsupervised pre-training with word representations

State-of-the-Art Sentiment Classification (cont)

Learning word representations

- No semantic information in classical word vectors
  - *D1= “banking finance banking”*
  - *D2 = “insurance finance insurance”*
  - *D3 = “medical”*

Banking

1	0	0	0

Finance

0	1	0	0

Insurance

0	0	1	0

Medical

0	0	0	1

State-of-the-Art Sentiment Classification (cont)

Learning word representations (cont)

- Some improvements with freq. or tf-idf information

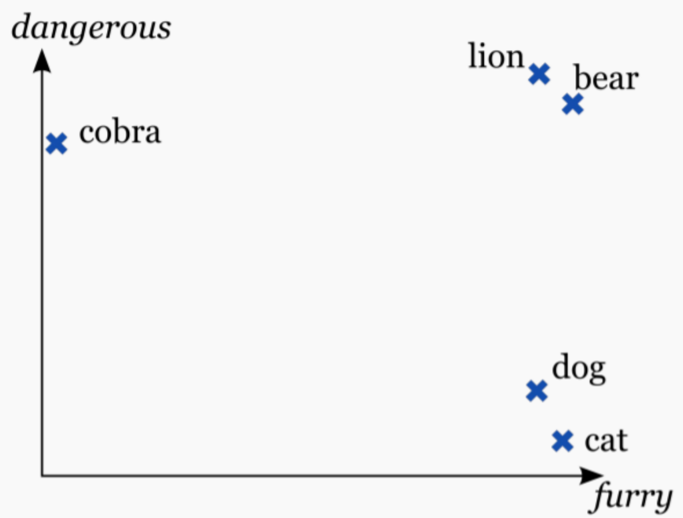
	D1	D2	D3
Banking	2	0	0
Finance	1	1	0
Insurance	0	2	0
Medical	0	0	1

- Aim
  - Vectors containing **semantic information**
  - **Semantic: meaning**
  - Vectors known as **word embeddings**

State-of-the-Art Sentiment Classification (cont)

Learning word representations (cont)

- Example of semantic vectors (plotted)



	furry	*dangerous*
bear	0.9	0.85
cat	0.85	0.15
cobra	0.0	0.8
lion	0.85	0.9
dog	0.8	0.15

- ***How to learn these vectors?***

State-of-the-Art Sentiment Classification (cont)

Learning word representations (cont)

- Seminal article of Mikolov et al.(2013)

- 2 shallow-NN models
  - Skip-gram
  - CBOW

- Input
  - Large text collection, e.g. Google News

State-of-the-Art Sentiment Classification (cont)

- Skip-gram output
  - Words appearing in the context of a source word
  - E.g. Phone **works** very well

Context (to be predicted)

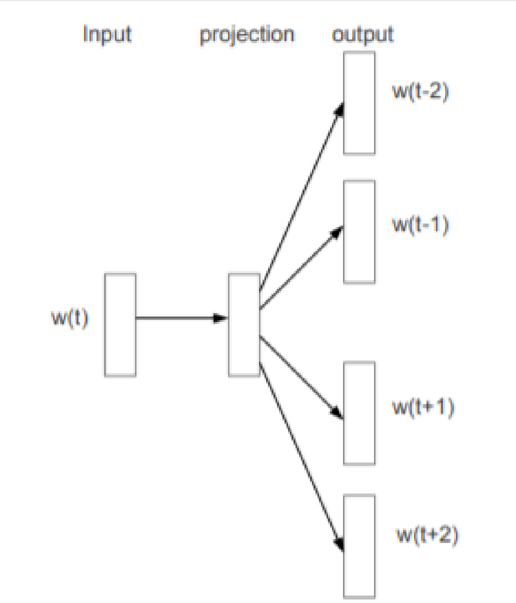
Context (to be predicted)

- CBOW output
  - Given context, predict target (middle word)

- Skip-gram + CBOW: Word2Vec

State-of-the-Art Sentiment Classification (cont)

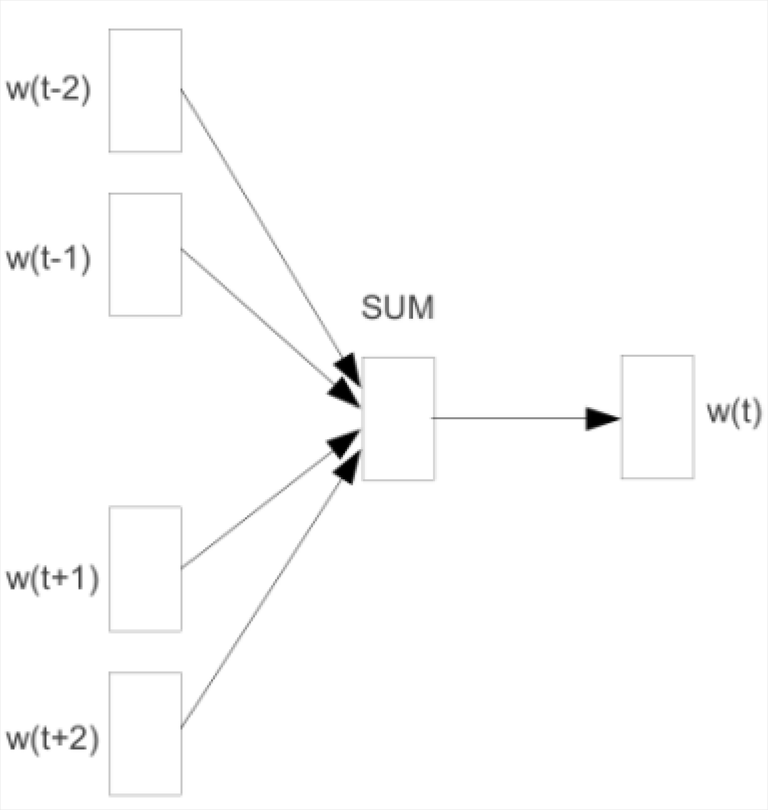
- Skip-gram



$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^\top v_{w_I}\right)}$$

State-of-the-Art Sentiment Classification (cont)

- CBOW



State-of-the-Art Sentiment Classification (cont)

- Recall our aim
  - Predict target word given context or
  - Context given “middle” word
- However
  - When training NN for these tasks
  - Vectors capturing semantic information generated

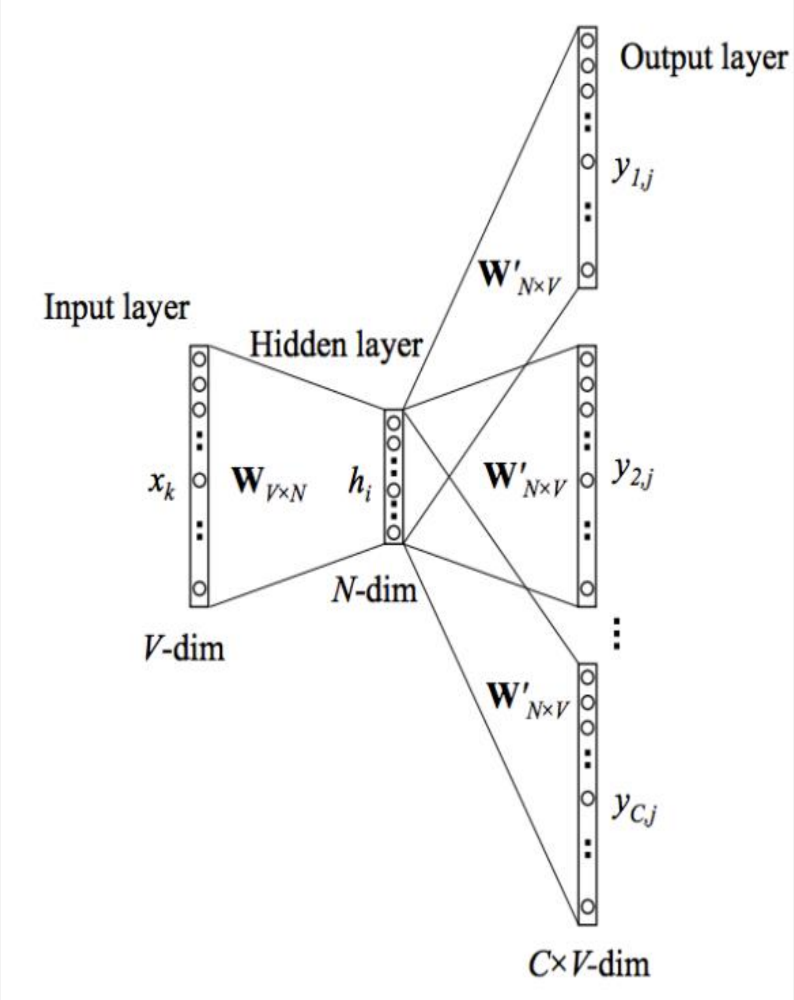
- Not so much interested in end task
- But in intermediate results
- Similar idea in SentimentNeuron (Radford et al).
  - Train for next char. Prediction
  - Intermediate results “encode” sentiment information

State-of-the-Art Sentiment Classification (cont)

- Back to learning word embeddings

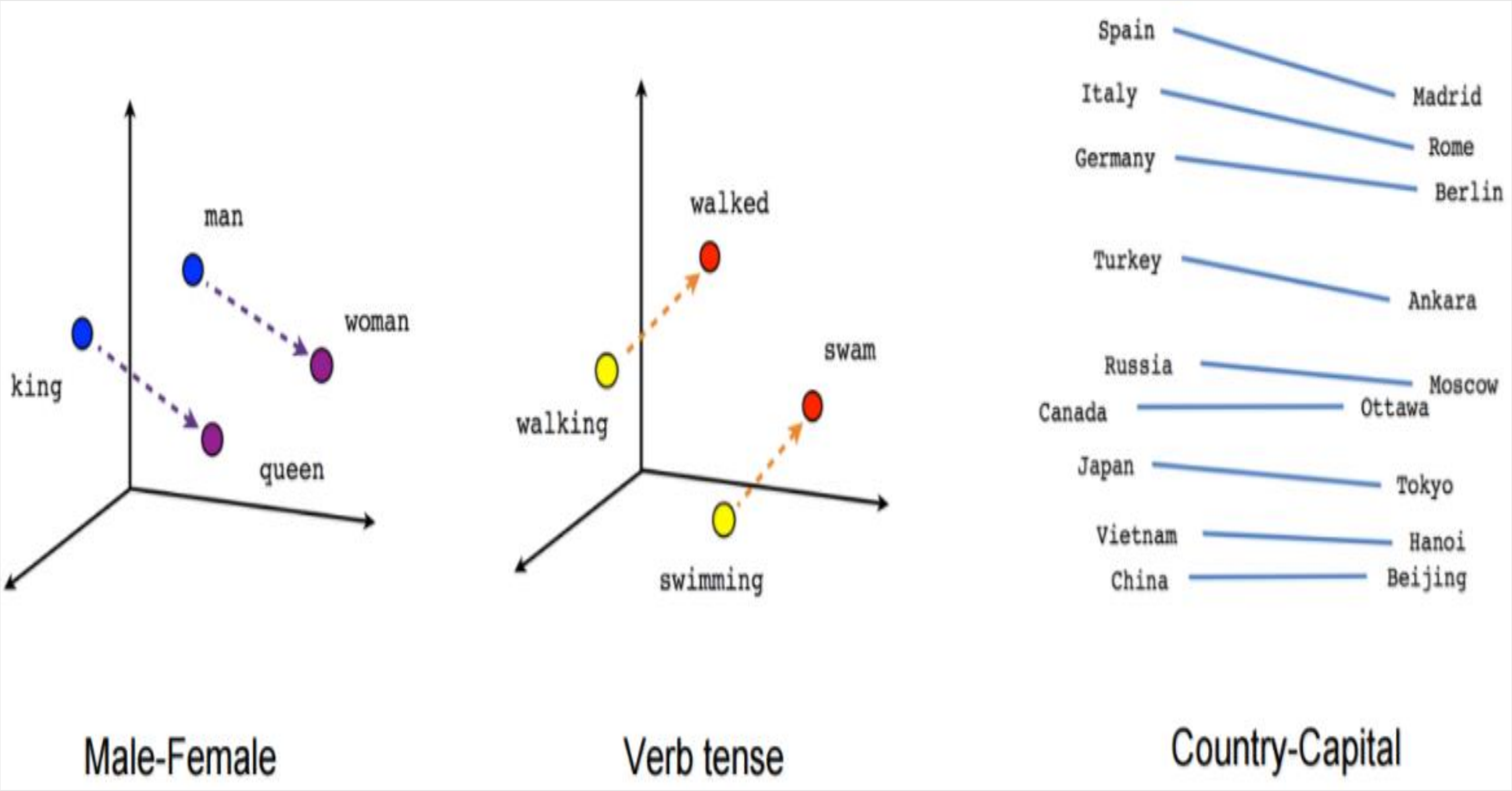
- Voc size = V
- N << V

- Each W_{VxN} row
  - A word vector
  - N dimensions
  - Dense representation of words
  - (Word embeddings)
  - Capture semantic info.



State-of-the-Art Sentiment Classification (cont)

- Plotting word embeddings



State-of-the-Art Sentiment Classification (cont)

- Word Embeddings generated by
  - Word2Vec
  - Glove

- Used in downstream NLP tasks (Unsupervised pre-training)
  - Machine Translation
  - Sentiment Analysis

- Radford et al. use **Skip-Thought vectors**
  - Similar to word embeddings
  - But at the sentence level

State-of-the-Art Sentiment Classification (cont)

- Generating word Embeddings computationally expensive

- Pre-trained embeddings
  - Available from various sources
  - Useful for domain adaption

- List of pre-trained word embeddings

- Python Word embedding example

Sentiment & Stock Movement

- Large area of research
- Seminal study of Bollen et al. (2010)
  - https://arxiv.org/pdf/1010.3003&

- Data
  - Public tweets, from Feb 28th – Dec 19th 2008
  - ~10 million tweets in all

- Sentiment of tweets using OpinionFinder
  - http://mpqa.cs.pitt.edu/opinionfinder/
- 6 types of moods inferred from tweets
  - Calm, Alert, Sure, Kind, Vital, Happy

Sentiment & Stock Movement (cont)

- Investigate whether sentiment, mood correlate with stock movements (DJIA)
- Granger causality (time series)
  - If variable X causes Y then changes in X will systematically occur before changes in Y

Lagged DJIA time series

DJIA time series

$$D_t = \alpha + \sum_{i=1}^{n} \beta_i D_{t-i} + \sum_{i=1}^{n} \gamma_i X_{t-i} + \epsilon_t$$

Opinion or mood

Challenges: Going Beyond Sentiment

- Sentiments not sufficient for some application
- Require more subtle information

- E.g. detecting whether a financial text recommends growth or value stock
  - *there has been a structural change in many industries resulting in lower profitability for small independent firms, Ritter argues*. → Growth stock recommendation

  - spreads have never stayed this wide for long, and when they normalize, patient value investors are richly rewarded. → value

Challenges: Going Beyond Sentiment (cont)

- Stance detection
- Detecting authors' standpoint
- Often with argument mining
  - Detecting argumentative structures from text
  - E.g. premise and conclusion
  - Applied for analyzing political debates

Some Resources

- Lexicons , Dictionaries
  - WordNet (http://wordnetweb.princeton.edu/perl/webwn)
  - WordNet extended to SentiWordNet (http://sentiwordnet.isti.cnr.it/)
  - General Inquirer http://www.wjh.harvard.edu/~inquirer/
  - MPQA http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
  - Liu Bing's http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
  - Emolex http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

- General NLP tools
  - https://nlp.stanford.edu/software/
  - Features (PoS, Syntactic dependencies)

- NLP Tools for Tweets
  - http://www.cs.cmu.edu/~ark/TweetNLP/

- "Happy" Corpus
  - https://arxiv.org/abs/1801.07746
  - https://www.kaggle.com/ritresearch/happydb
  - ***Distant-supervision? Transfer learning?***

References

- Turney, P. D. (2002). Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In ACL-02
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In KDD, pp. 168–177.
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In SIGKDD-04
- Bollen, Johan, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market." Journal of computational science2.1 (2011): 1-8.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)
- Radford, A., Jozefowicz, R., & Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. arXiv preprint arXiv:1704.01444.