CS 481

Artificial Intelligence Language Understanding

March 23, 2023

Announcements / Reminders

- Please follow the Week 20 To Do List instructions
- Written Assignment #03 is due on Sunday 03/26/23 at 11:59 PM CST
- Programming Assignment #02 is due on Sunday 04/02/23 at 11:59 PM CST

- Final Exam date:

  Thursday 04/27/2023 (last week of classes!)

  - Ignore the date provided by the Registrar
  - Section 02 [Online]: contact Mr. Charles Scott (scott@iit.edu) to arrange your exam

Plan for Today

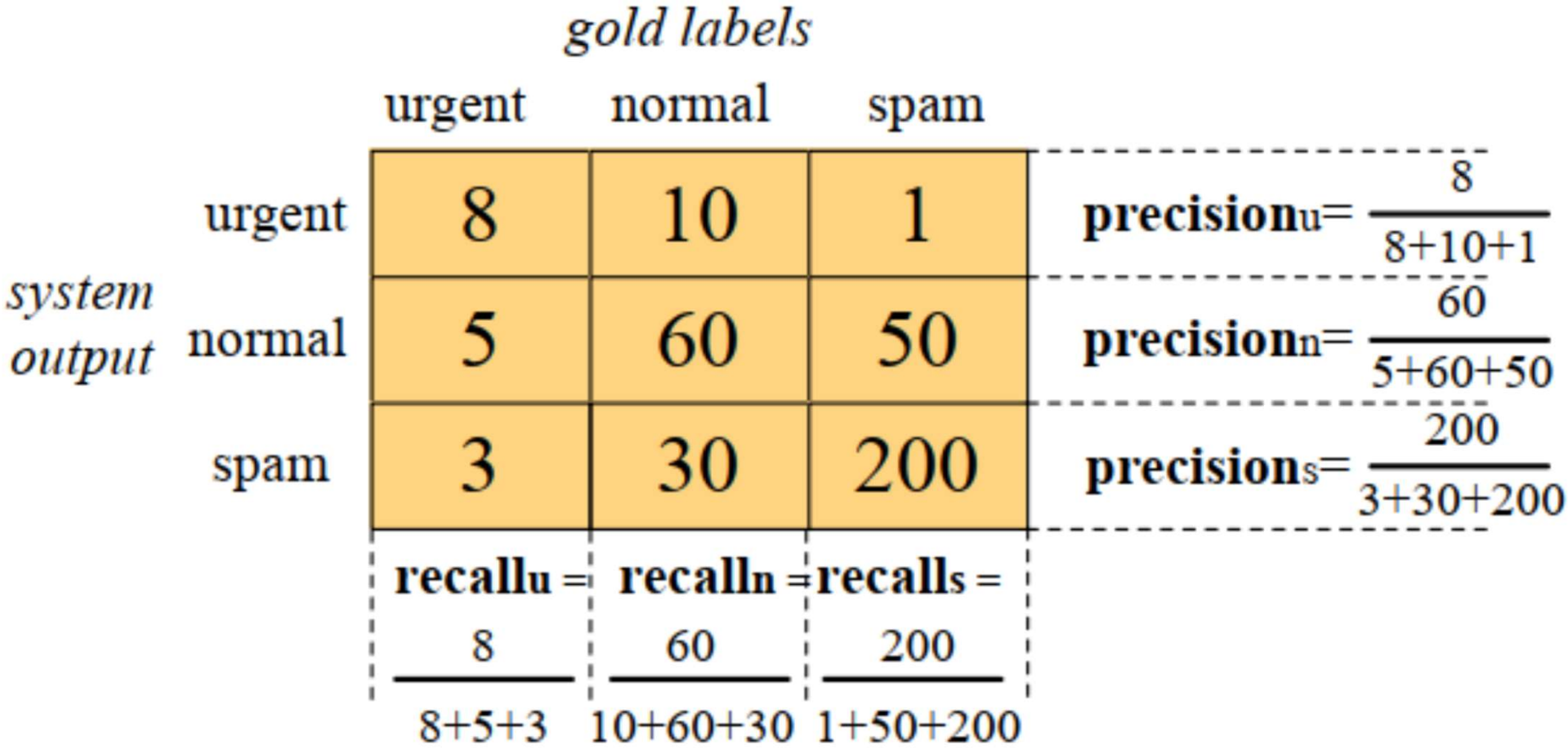
- Sentiment Analysis
- Words and their meaning

Classifier Evaluation: F-Score

- F-Score is a measure of a model's accuracy on a dataset
- evaluates binary classification systems
- F-score is a way of combining the precision and recall of the model
- Used in NLP, Information Retrieval, ML

$$F = \frac{2}{\frac{1}{recall} * \frac{1}{precision}} = 2 * \frac{recall * precision}{recall + precision} = \frac{TP}{TP + 0.5 * (FP + FN)}$$

3-class Confusion Matrix



Macroaveraging and Microaveraging

Macroaveraging:

- **compute the performance for each class, and then average over classes**

Microaveraging:

- **collect decisions for all classes into one confusion matrix**
- **compute precision and recall from that table.**

Macroaveraging and Microaveraging

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

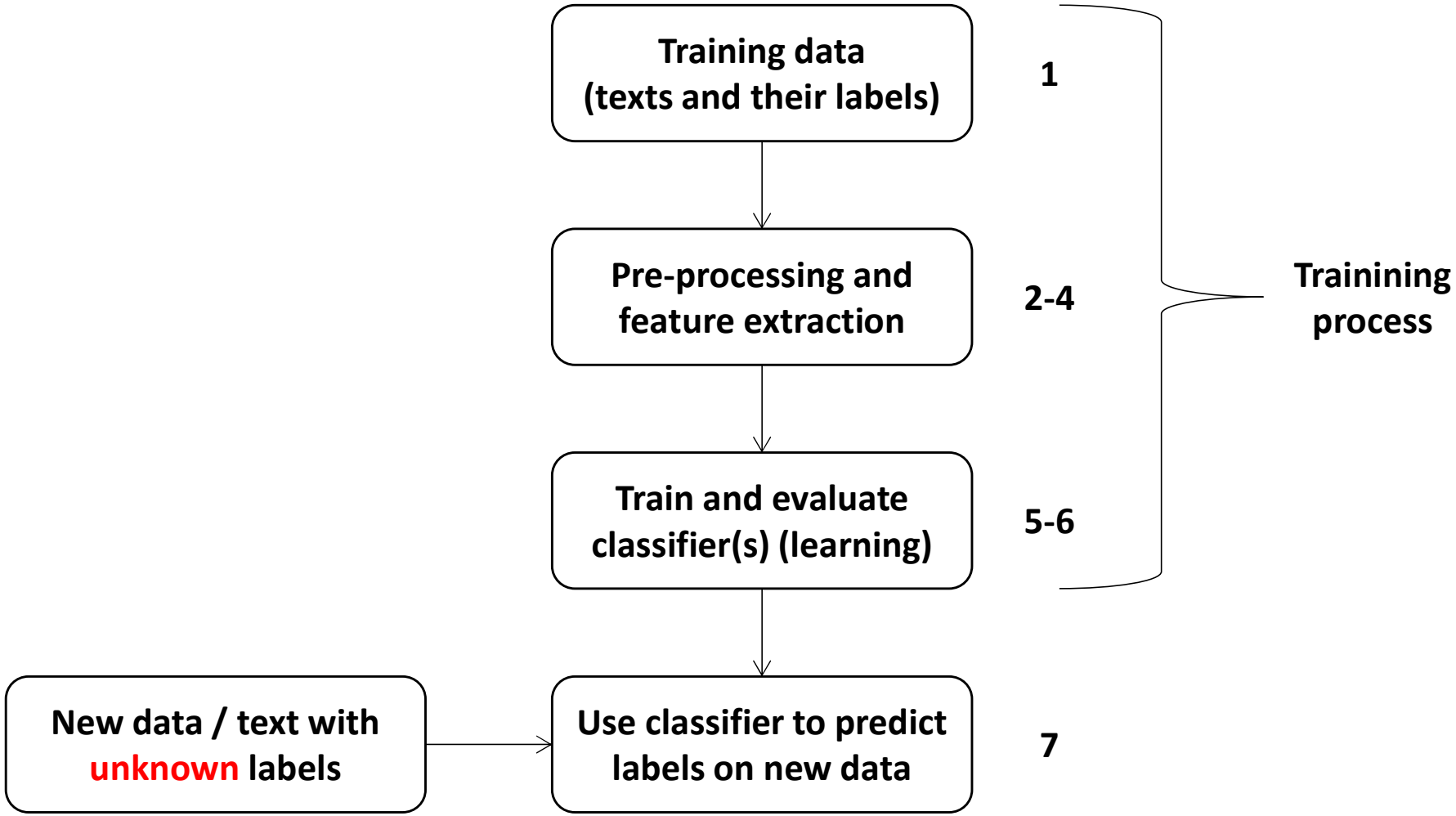
$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$

Text Classification System Pipeline

1. Obtain / collect / create labeled data set suitable for the task
2. Split the data set into:
   - two (training and test sets) parts OR
   - three (training, validation, and test sets) parts
3. Choose evaluation metric
4. Transform raw text into feature vectors:
   - bag of words
   - other types
5. Using feature vectors and labels from the training set, train the classifier / create a model
6. Using evaluation metric from (3) benchmark the classifier / model performance using the test set
7. Deploy the classifier / model to serve a real world application and monitor its performance

Text Classification System Pipeline



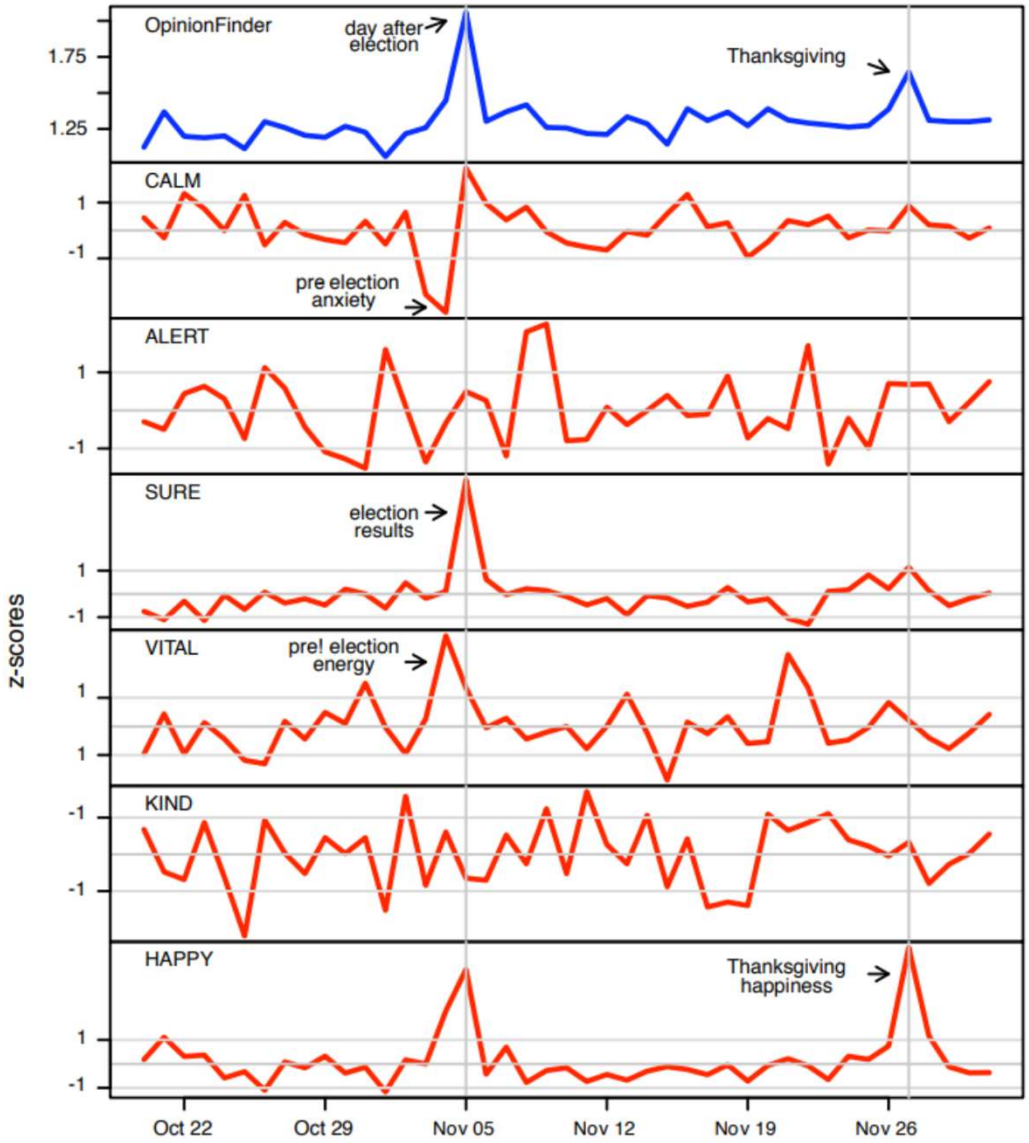
Poor Classifier Performance: Reasons

1. With all possible features extracted, we ended up with a sparse feature vector (some features are too rare and end up being noise) $\rightarrow$ makes training hard
2. Few (~20%) relevant samples compared to non-relevant (~80%) samples in the data set $\rightarrow$ skews learning towards non-relevant data
3. Need better learning algorithm
4. Need better pre-processing / feature extraction
5. Classifier parameters / hyperparameters need tuning

Sentiment Analysis: Motivation

- Movie: is this review positive or negative?
- Products: what do people think about the new iPhone?
- Public sentiment: what is consumer confidence?
- Politics: what do people think about this candidate or issue?
- Prediction: predict election outcomes or market trends from sentiment

Sentiment Analysis: Twitter Mood

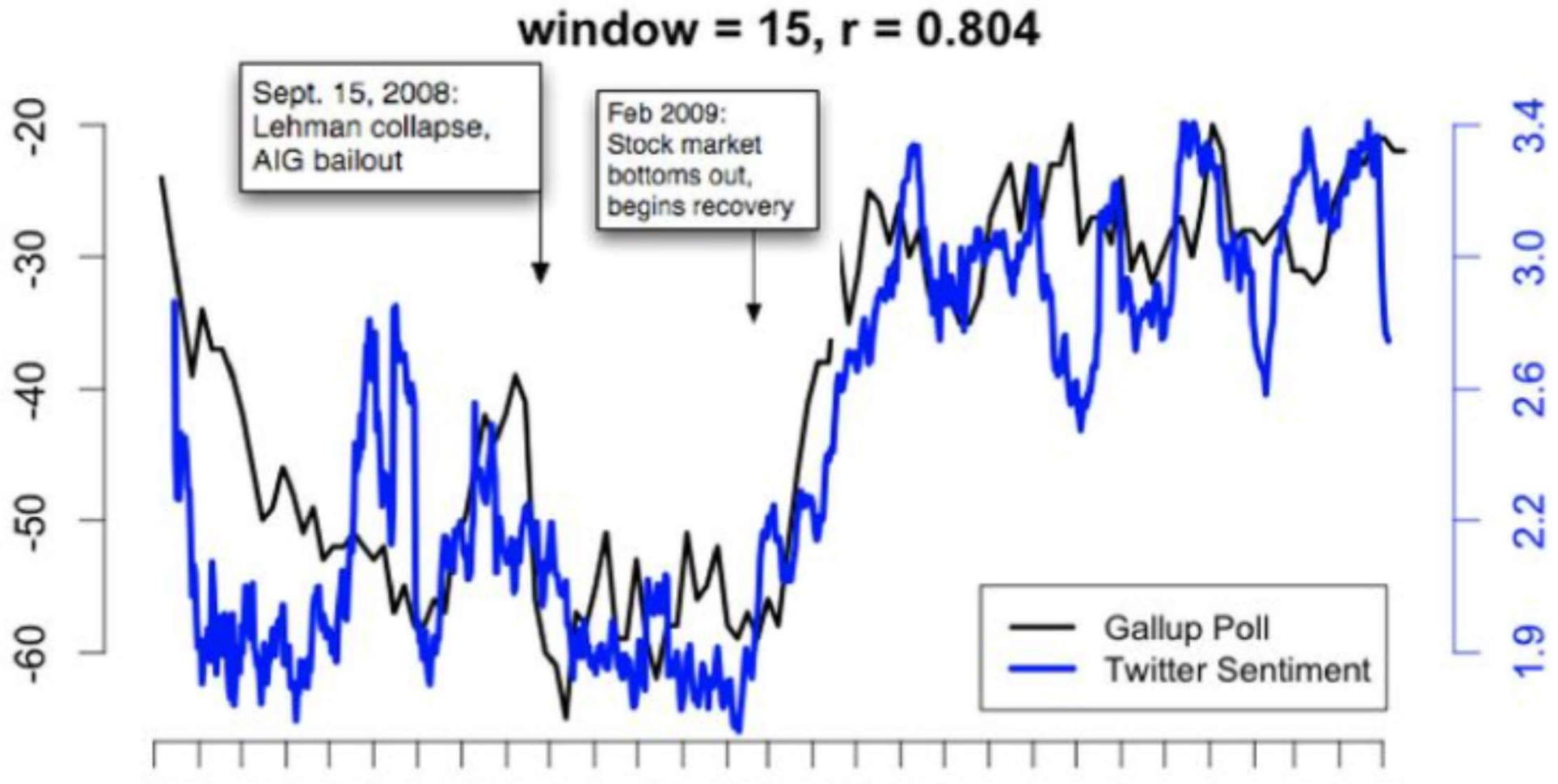


source: https://arxiv.org/pdf/1010.3003.pdf

Sentiment Analysis: Tweets

source: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

Sentiment Analysis: Text and Polls



source: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1536/1842

Affective State

Affective state: any kind of sentimental condition, often in which someone's feelings control their consciousness.

Scherer Typology of Affective States

- Emotion: brief organically synchronized … evaluation of a major event
  - angry, sad, joyful, fearful, ashamed, proud, elated
- Mood: diffuse non-caused low-intensity long-duration change in subjective feeling
  - cheerful, gloomy, irritable, listless, depressed, buoyant
- Interpersonal stances: affective stance toward another person in a specific interaction
  - friendly, flirtatious, distant, cold, warm, supportive, contemptuous
- Attitudes: enduring, affectively colored beliefs, dispositions towards objects or persons
  - liking, loving, hating, valuing, desiring
- Personality traits: stable personality dispositions and typical behavior tendencies
  - nervous, anxious, reckless, morose, hostile, jealous

Scherer Typology of Affective States

- Emotion: brief organically synchronized … evaluation of a major event
  - angry, sad, joyful, fearful, ashamed, proud, elated
- Mood: diffuse non-caused low-intensity long-duration change in subjective feeling
  - cheerful, gloomy, irritable, listless, depressed, buoyant
- Interpersonal stances: affective stance toward another person in a specific interaction
  - friendly, flirtatious, distant, cold, warm, supportive, contemptuous
- Attitudes: enduring, affectively colored beliefs, dispositions towards objects or persons
  - liking, loving, hating, valuing, desiring
- Personality traits: stable personality dispositions and typical behavior tendencies
  - nervous, anxious, reckless, morose, hostile, jealous

Sentiment Analysis: Detecting Attitude

Sentiment analysis deals with detection of attitudes ("enduring, affectively colored beliefs, dispositions towards objects or persons"). It involves four components:

1. **Holder (source)** of attitude
2. **Target (aspect)** of attitude
3. **Type** of attitude:
   - **from a set of types:** *like*, *love*, *hate*, *value*, *desire*, **etc.**
   - **simple (weighted) polarity:** *positive*, *negative*, *neutral* **(together with** *strength***)**
4. **Text / data** containing attitude:
   - **sentences,**
   - **entire documents, etc.**

Sentiment Analysis: Other Names

- Opinion extraction
- Opinion mining
- Sentiment mining
- Subjectivity analysis

Sentiment Analysis: Complexity Levels

Sentiment analysis tasks can have different levels of complexity:

- **Simple tasks:**
  - **decide if the attitude contained within text is positive or negative**
- **More complex / average complexity tasks:**
  - **rank that attitude on a scale from 1 to 5**
- **Advanced / complex tasks:**
  - **detect target (stance detection)**
  - **detect source**
  - **detect complex attitude types**

Sentiment Analysis: Baseline Algo

- Pre-processing: tokenization
- Feature extraction: bag of words, etc.
- Classification using a chosen classifier
  - Naive Bayes
  - Perceptron
  - Support Vector Machines
  - etc.

Naive Bayes Classifier

$$\text{category/class} = \text{h}(\textbf{document})$$

Finding model / hypothesis h → Finding probabilities for y_{MAP}

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Note: logarithms (to handle underflow) will be ignored in the following example

MAP: Maximum a posteriori (corresponds to the most likely class).

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable with no fun	$y_6 = ???$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable with no fun	$y_6 = ???$

Learning

Probability estimates:

Naive Bayes Classifier:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Probability estimates (Maximum Likelihood estimation):

$$P(y = +) = \frac{N_{samples\ labeled\ +}}{N}$$

$$P(y = -) = \frac{N_{samples\ labeled\ -}}{N}$$

$$P(x_i = word \mid y = CLASS) = \frac{count(x_i = word, y = CLASS) + 1}{\sum_{x \in V} count(x, y = CLASS) + |V|}$$

Note the add-1 smoothing!

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable with no fun	$y_6 = ???$

Learning

$$P(y = +) = \frac{N_{samples\ labeled\ +}}{N} = \frac{2}{5}$$

$$P(y = -) = \frac{N_{samples\ labeled\ -}}{N} = \frac{3}{5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable with no fun	y_6=???

Learning

word	+	-
just	0	1
plain	0	1
boring	0	1
entirely	0	1
predictable	0	1
and	0	2
lacks	0	1
energy	0	1
no	0	1
surprises	0	1
very	1	1
few	0	1
laughs	0	1
powerful	1	0
the	2	0
most	1	0
fun	1	0
film	1	0
of	1	0
summer	1	0

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	$y_6 = ???$

unknown word (not in V) → ignore

Learning

$$P(x_i = predictable \mid y = +) = \frac{count(x_i = predictable, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{count(x_i = no, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{count(x_i = fun, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{1+1}{9+20} = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{count(x_i = predictable, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{count(x_i = no, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{count(x_i = fun, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{0+1}{14+20} = \frac{1}{34}$$

Other probabilities not shown

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	$y_6 = ???$

unknown word (not in V) → ignore

Model

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Other probabilities not shown

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	y_6=???

unknown word (not in V) → ignore

Prediction

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Label + case:

$$P(y = +) * P(x_6 \mid y = +) = \frac{2}{5} * \frac{1}{29} * \frac{1}{29} * \frac{2}{29} = 3.2 * 10^{-5}$$

Label - case:

$$P(y = -) * P(x_6 \mid y = -) = \frac{3}{5} * \frac{2}{34} * \frac{2}{34} * \frac{1}{34} = 6.1 * 10^{-5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	$y_6 = -$

unknown word (not in V) → ignore

Model

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Label + case:

$$P(y = +) * P(x_6 \mid y = +) = \frac{2}{5} * \frac{1}{29} * \frac{1}{29} * \frac{2}{29} = 3.2 * 10^{-5}$$

Label - case:

$$P(y = -) * P(x_6 \mid y = -) = \frac{3}{5} * \frac{2}{34} * \frac{2}{34} * \frac{1}{34} = 6.1 * 10^{-5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Feature Extraction

- What should we extract?
  - just adjectives?
  - or perhaps all words?

- Extracting all words is likely to work better

Optimizing for Sentiment Analysis

- For tasks like sentiment, word occurrence seems to be more important than word frequency
  - the occurrence of the word *fantastic* tells us a lot
  - the fact that it occurs 5 times may not tell us much more.

- Use binary multinominal Naive Bayes, or binary NB
  - clip word counts at 1 (1 per document even if more)

Sentiment Analysis: Binarization

Four original documents:

- – it was pathetic the worst part was the boxing scenes
- – no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- – it was pathetic the worst part boxing scenes
- – no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

	NB Counts		Binary Counts	
	+	–	+	–
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Sentiment Analysis: Binarization

Four original documents:

- – it was pathetic the worst part was the boxing scenes
- – no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- – it was pathetic the worst part boxing scenes
- – no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

duplicate words removed

	NB Counts		Binary Counts	
	+	–	+	–
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Sentiment Analysis: Binarization

Four original documents:

- – it was pathetic the worst part was the boxing scenes
- – no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- – it was pathetic the worst part boxing scenes
- – no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

	NB Counts		Binary Counts	
	+	–	+	–
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

great **and** *scenes* **appear in multiple documents**

Sentiment Analysis: Negations

Consider the following sentences / documents:

- *I really like this movie*
- *I really don't like this movie*

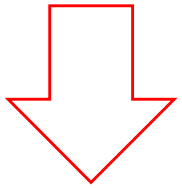
- **Negation changes the meaning of "like" to negative.**
- **Negation can also change negative to positive-ish**
  - *Don't dismiss this film*
  - *Doesn't let us get bored*

Sentiment Analysis: Negations

Simple baseline method solution:

- **Add NOT_ prefix to every word between negation and following punctuation:**

didn't like this movie , but I



didn't NOT_like NOT_this NOT_movie , but I

Sentiment Analysis: Lexicons

What if we don't have enough labeled training data?

- **In that case, we can make use of pre-built word lists → lexicons**

There are various publicly available lexicons

Sentiment Analysis: Lexicons

Add a feature that gets a count whenever a word from the lexicon occurs:

- **for example: a feature called "this word occurs in the positive lexicon" or "this word occurs in the negative lexicon"**
- **all positive words (good, great, beautiful, wonderful) or negative words count for that feature.**

Using 1-2 features isn't as good as using all the words.

But when training data is sparse or not representative of the test set, dense lexicon features can help

MPQA Subjectivity Cues Lexicon

Home page:
https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

6885 words from 8221 lemmas, annotated for intensity (strong/weak)

2718 positive

4912 negative

+ : admirable, beautiful, confident, dazzling, ecstatic, favor, glee, great

– : awful, bad, bias, catastrophe, cheat, deny, envious, foul, harsh, hate

The General Inquirer

Home page:

http://www.wjh.harvard.edu/~inquirer

List of Categories:

http://www.wjh.harvard.edu/~inquirer/homecat.htm

Categories:

- **Positive (1915 words) and Negative (2291 words)**
- **Strong vs Weak, Active vs Passive, Overstated versus Understated**
- **Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc**

Free for Research Use!

Sentiment Analysis: Lexicons

What if there is no lexicon for a given domain?

- **In that case, we can learn it:**
  - **start with a few labeled examples and hand-built patterns**
  - **build lexicon using the procedure below:**

function BUILDSENTIMENTLEXICON(*posseeds*,*negseeds*) **returns** *poslex*,*neglex*

poslex ← *posseeds*
neglex ← *negseeds*
Until done
  poslex ← *poslex* + FINDSIMILARWORDS(*poslex*)
  neglex ← *neglex* + FINDSIMILARWORDS(*neglex*)
poslex,*neglex* ← POSTPROCESS(*poslex*,*neglex*)

Identifying Word Polarity

Intuition:

- **words joined by "*and*" have the same polarity:**
  - *fair **and** legitimate, corrupt **and** brutal*

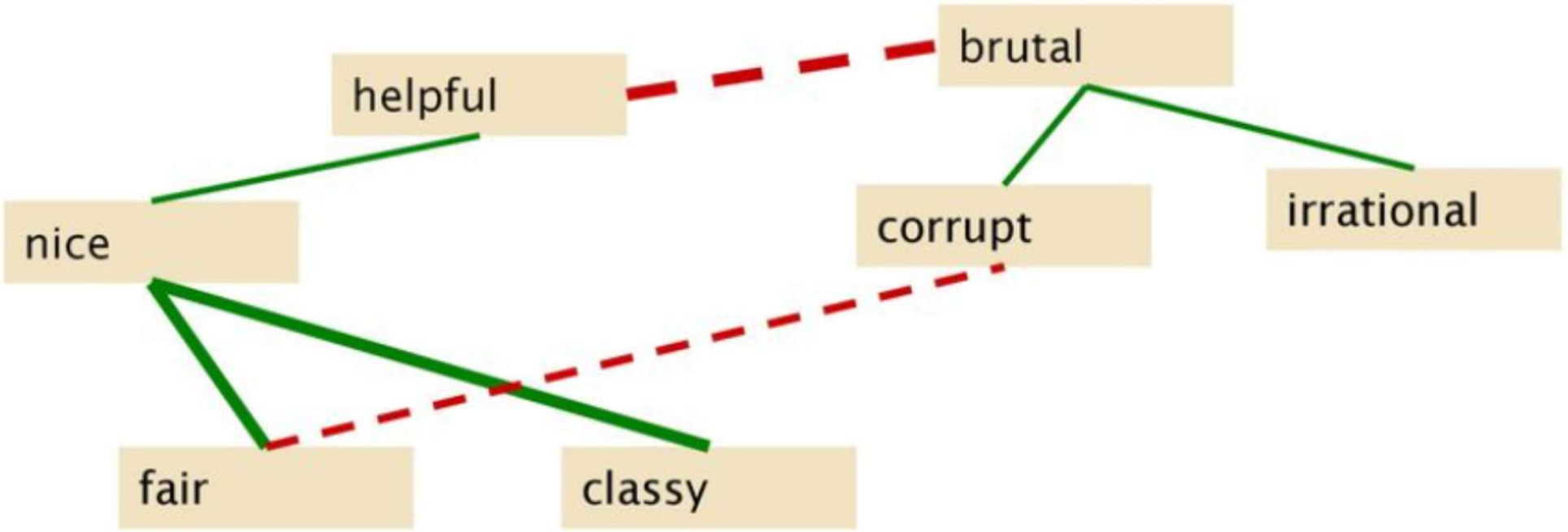
- **words joined by "*but*" do not:**
  - *fair, **but** brutal*

Sentiment Analysis: Learning Lexicons

- **Motivation:**
  - **learn a domain-specific lexicon**
  - **more words (more robust) than off-the-shelf lexicon**

- **Intuition:**
  - **start with "seed" words (*good*, *poor*)**
  - **find other words with similar polarity:**
    - **use "*and*" and "*but*"**
    - **use nearby words in the same document**
    - **add them to lexicon**

Identifying Word Polarity: Graph



Sentiment Analysis: Challenges

Subtlety:

- **Perfume review in** Perfumes: The Guide**:**
  - *"If you are reading this because it is your darling fragrance, please wear it at home exclusively and tape the windows shut"*
- **Dorothy Parker (writer) on Katherine Hepburn (actress):**
  - *"She runs the gamut of emotions from A to B"*

Sentiment Analysis: Challenges

Thwarted expectations:

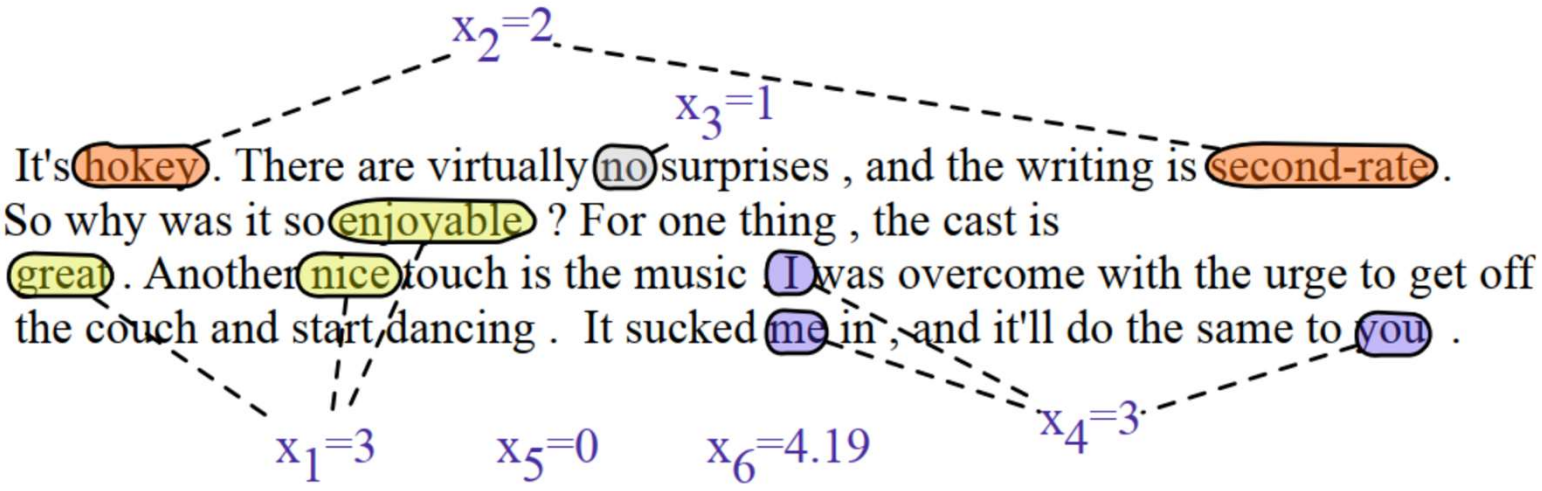
- *"This film should be **brilliant**. It sounds like a **great** plot, the actors are **first grade**, and the supporting cast is **good** as well, and Stallone is attempting to deliver a **good** performance. However it **can't hold up**."*

Sentiment Analysis: Challenges

Ordering effect:

- *"Well as usual Keanu Reeves is nothing special, but surprisingly, the **very talented** Laurence Fishbourne **not so good** either. I was surprised."*

Sentiment Analysis: Example



Feature vector:

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Sentiment Analysis: Example

Feature vector x:

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

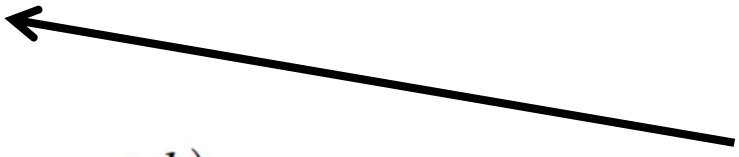
What is w·x + b ???

Suppose: w = [2.5, 5.0, 1.2, 0.5, 2.0, 0.7]
and b = 0.1

Sentiment Analysis: Example

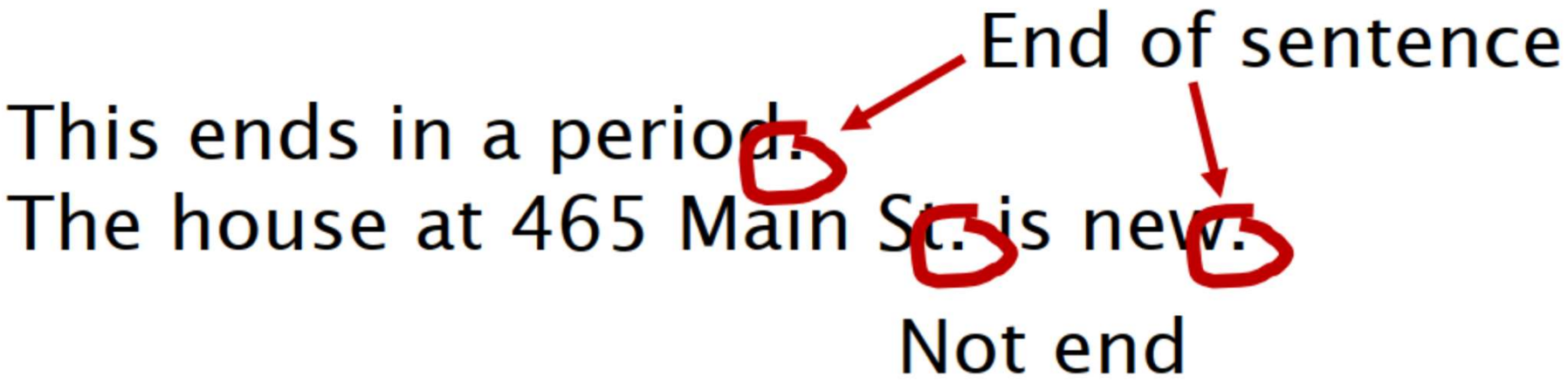
$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$



Positive sentiment

Period Disambiguation: Example



Feature vector x

$$x_1 = \begin{cases} 1 & \text{if } \text{“}Case(w_i) = \text{Lower”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if } \text{“}w_i \in \text{AcronymDict”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if } \text{“}w_i = \text{St. \& } Case(w_{i-1}) = \text{Cap”} \\ 0 & \text{otherwise} \end{cases}$$

What do Words Mean?

In methods (N-grams, text classification, etc.) we've seen:

- words are just strings
- **meaning** is not considered

Meaning in logic:

- The meaning of "dog" is DOG (predicates and symbols)

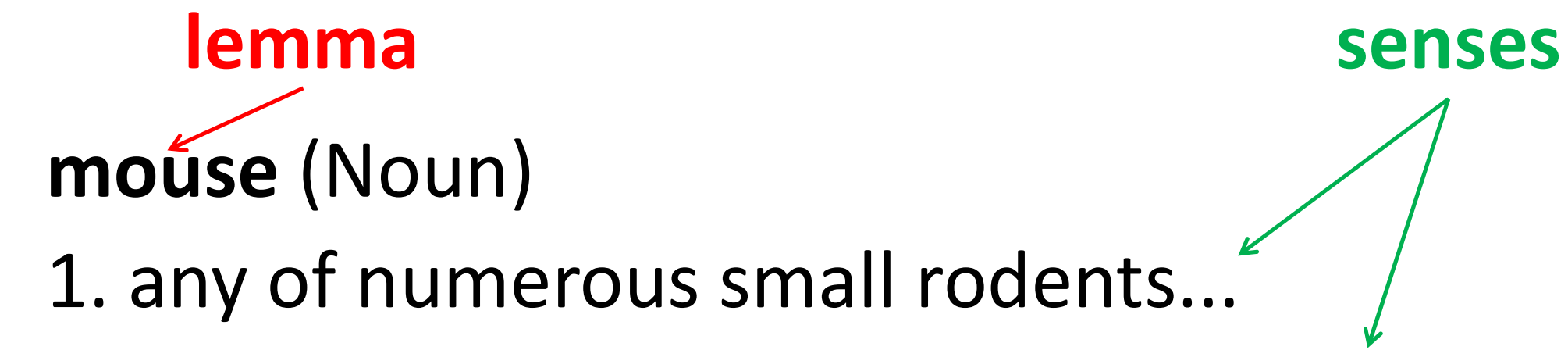
$$\forall x\, \mathrm{DOG}(x) \rightarrow \mathrm{MAMMAL}(x)$$

Old linguistics joke by Barbara Partee in 1967:

- Q: What's the **meaning** of life?
- A: LIFE

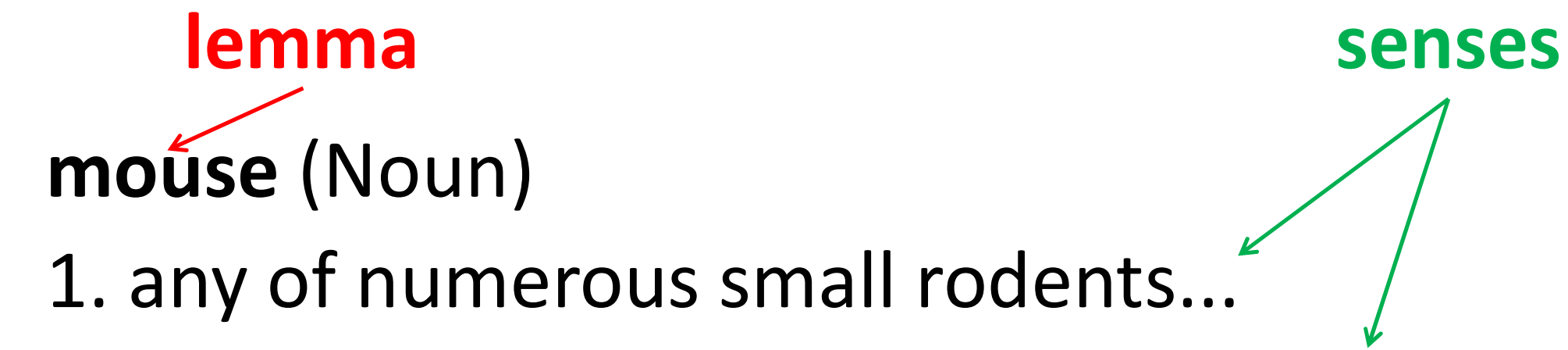
That is not very helpful.

Words: Lemmas and Senses



from the online thesaurus WordNet

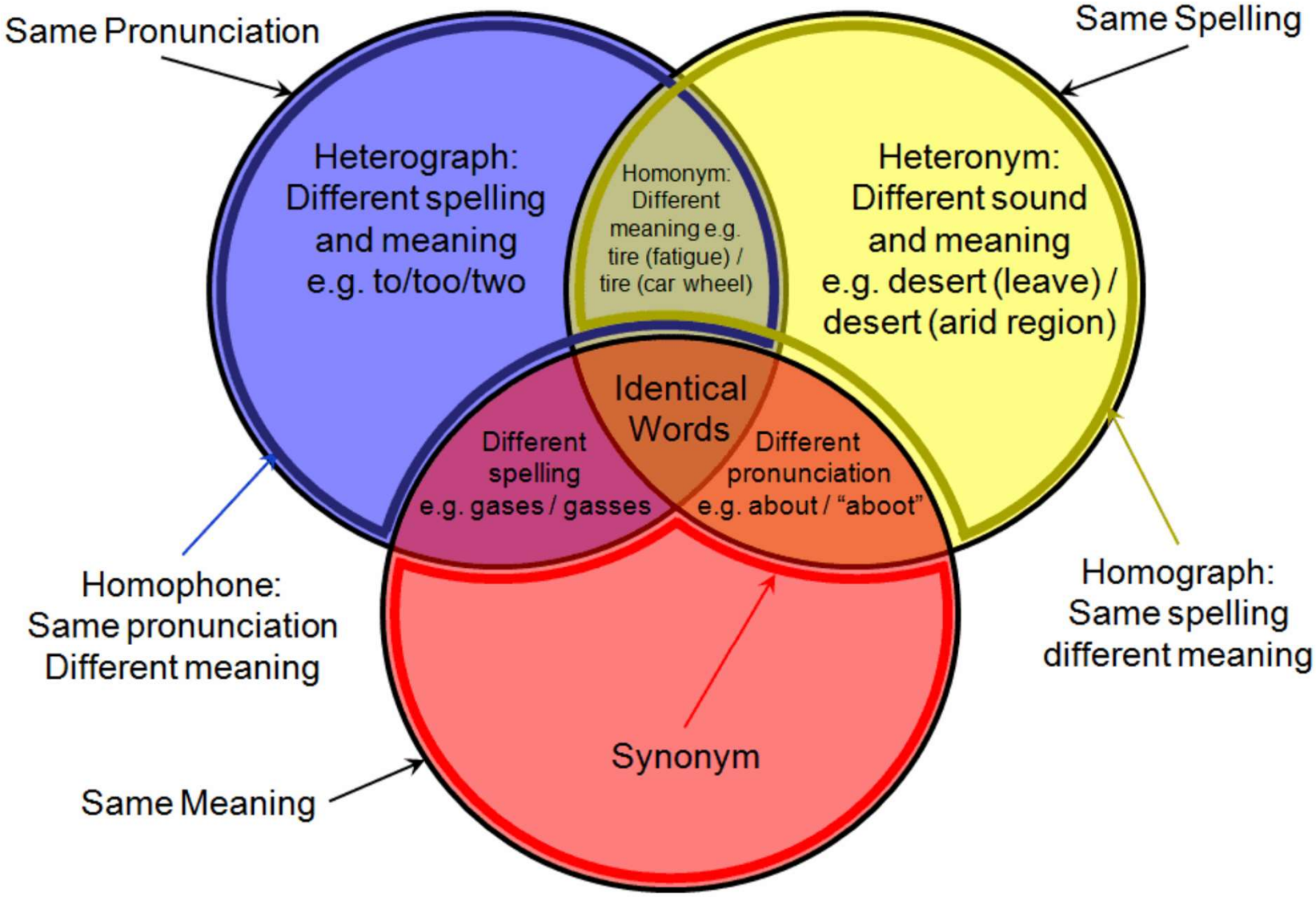
Words: Lemmas and Senses



from the online thesaurus WordNet

- **A sense or "concept" is the meaning component of a word**
- **Lemmas can be polysemous (have multiple senses)**

Relationships Between Words



Words Different In Pronunciation, Spelling, and Meaning

Source: https://owlcation.com/humanities/Lexical-Relations-Describing-Similarities-In-The-English-Language

Lexical Relationships

Lexical relationships are the connections established between one word and another:

- **Synonymy is the idea that some words have the same meaning as others**
  - *quick* **is similar to** *fast*
- **Antonymy is precisely the opposite of synonymy**
  - *good* **is the opposite** *bad*
- **Hyponymy is similar to the notion of embeddedness**
  - *Human* ← *Female* **(***Female* **is a more specific concept than** *Human***)**
- **Holonomy and Meronomy describe relationships between an object and its parts:**
  - *tree* **is a holonym of** *bark* **(***tree* **has bark)**
  - *bark* **is a meronym of** *tree* **(***bark* **is a part of** *tree***)**

Lexical Semantics: Definition

Lexical semantics:

*the branch of linguistics and logic concerned with **meaning**. There are a number of branches and subbranches of semantics, including:*

- **formal semantics**, *which studies the logical aspects of **meaning**, such as **sense**, **reference**, **implication**, and **logical form**,*
- **lexical semantics**, *which studies **word meanings** and **word relations**, and **conceptual semantics**, which studies the **cognitive structure of meaning**.*

from Oxford Dictionary

Sense Relationships: Synonymy

- **Synonyms have the same meaning in some or all contexts:**
  - *filbert / hazelnut*
  - *couch / sofa*
  - *big / large*
  - *automobile / car*
  - *vomit / throw up*
  - *water / H_20*

Sense Relationships: Synonymy

- There are probably no examples of perfect synonymy:
  - many aspects of meaning maybe identical, but not necessarily all aspects
  - words may differ based on:
    - politeness
    - slang
    - register,
    - genre, etc.

Sense Relationships: Synonymy?

- **Some examples:**
  - *water / H_20*

    Would "*H_20*" be used in a surfing guide?
  - *car / automobile*
  - *big / large*
    - *my big sister* **is NOT always going to be synonymous with** *my large sister*

The Linguistic Principle of Contrast

- **Substitutions between some pairs of words like *car / automobile* or *water / H_2O* are truth preserving, the words are still not identical in meaning**

- **The Linguistic Principle of Contrast**

  difference in form → difference in meaning

Sense Relationships: Similarity

- **Words with similar meanings.**
- **Not synonyms, but sharing some element of meaning**

- **Some examples:**
  - *cow / horse*
  - *car / bicycle*

Sense Relationships: Similarity

- Knowing how similar two words are can:
  - help in computing how similar the meaning of two phrases or sentences are
  - assist in higher level tasks:
    - question answering
    - paraphrasing
    - summarization

Sense Relationships: Similarity

Human-evaluated word similarity:

Word 1	Word 2	Similarity [0-10]
vanish	*disappear*	9.8
behave	*obey*	7.3
belief	*impression*	5.95
muscle	*bone*	3.65
modest	*flexible*	0.98
hole	*agreement*	0.3

Source: SimLex-999 dataset (Hill et al., 2015) | https://fh295.github.io/simlex.html

Sense Relationships: Relatedness

- **Also called "word association"**
- **Words can be related in any way, for example via a semantic frame or field**

- **Some examples:**
  - *coffee, tea:* **similar**
  - *coffee, cup:* **related, not similar**

Semantic Frame: Definition

Semantic Frame:

*a semantic frame is defined as a **coherent structure of concepts that are related** such that **without knowledge of all of them, one does not have complete knowledge of one** of the either*

from *https://cogling.fandom.com/wiki/Semantic_frame*

Semantic Field: Definition

Semantic Field:

*a **lexical set** of semantically **related items***

from Oxford Dictionary

Semantic Field

Words that

- **cover a particular semantic domain**
- **bear structured relations with each other.**

hospitals

surgeon, scalpel, nurse, anaesthetic, hospital

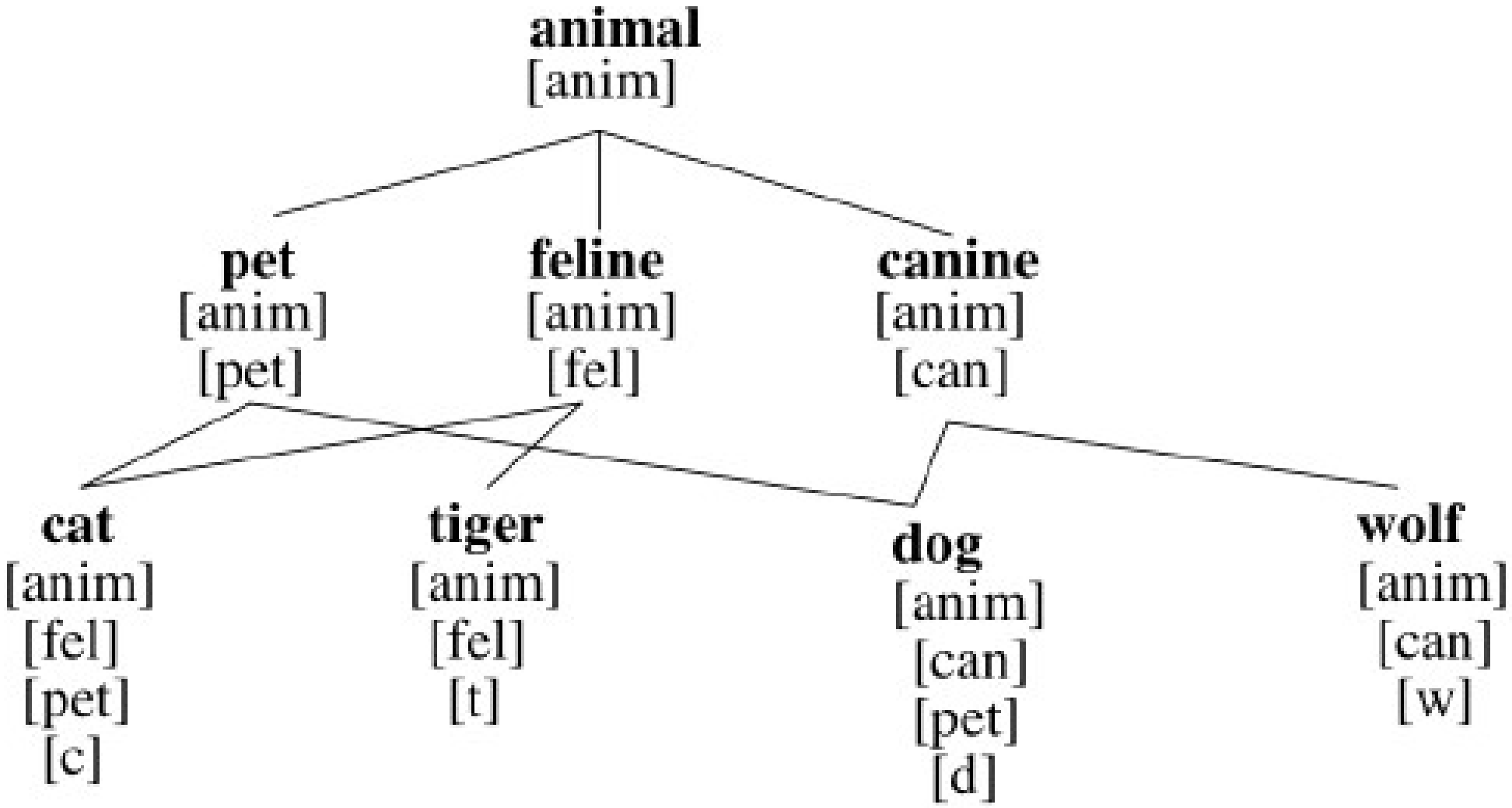
restaurants

waiter, menu, plate, food, menu, chef

houses

door, roof, kitchen, family, bed

Semantic Field



Source: Helge Dyvik - "Translations as a semantic knowledge source"

Sense Relationships: Antonymy

- **Senses that are opposites with respect to only one feature of meaning**

- **Otherwise, they are very similar (sharing some element of meaning):**
  - *dark/light* *short/long* *fast/slow*
  - *hot/cold* *up/down* *in/out*

Sense Relationships: Antonymy

- More formally, antonyms can
  - define a binary opposition or be at opposite ends of a scale:
    - *long / short*, *fast / slow*
  - be reversives:
    - *rise/fall*, *up/down*

Words and Meaning: Connotations

- **Words have affective meanings**
  - **positive connotations (*happy*)**
  - **negative connotations (*sad*)**
- **Connotations can be subtle:**
  - **positive connotation: *copy*, *replica*, *reproduction***
  - **negative connotation: *fake*, *knockoff*, *forgery***
- **Evaluation (sentiment):**
  - **positive evaluation (*great*, *love*)**
  - **negative evaluation (*terrible*, *hate*)**

Words and Meaning: Connotations

- **Words seem to vary along three affective dimensions:**
  - **valence: the pleasantness of the stimulus**
  - **arousal: the intensity of emotion provoked by the stimulus**
  - **dominance: the degree of control exerted by the stimulus**

	Word	Score		Word	Score
valence	love	1.000		toxic	0.008
	happy	1.000		nightmare	0.005
arousal	elated	0.960		mellow	0.069
	frenzy	0.965		napping	0.046
dominance	powerful	0.991		weak	0.045
	leadership	0.983		empty	0.081

Source: NRC VAD Lexicon (https://saifmohammad.com/WebPages/nrc-vad.html)

Words and Meaning: Summary

- Concepts or word senses
  - have a complex many-to-many association with words (homonymy, multiple senses)
- Have relations with each other
  - synonymy
  - antonymy
  - similarity
  - relatedness
  - connotation

WordNet

WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Synsets are interlinked by means of conceptual-semantic and lexical relations.

Link: https://wordnet.princeton.edu/

Challenge

- We know word relationships exist
- How can we quantify them in a automated fashion?
- How do we represent them in numerical way?
- How can we use them in computational models and processes?

Computational Models of Meaning

"*a word is characterized by the company it keeps*"

- John Rupert Firth (English linguist)

"*In most cases, the meaning of a word is its use*"

- Ludwig Wittgenstein (Austrian philosopher)

"*If A and B have almost identical environments we say that they are synonyms.*"

- Zellig Harris (American linguist)

Words + Their Environment: Example

- **Suppose you see these sentences:**
  - *Ong choi is delicious sautéed with garlic.*
  - *Ong choi is superb over rice*
  - *Ong choi leaves with salty sauces*
- **And you've also seen these:**
  - *…spinach sautéed with garlic over rice*
  - *Chard stems and leaves are delicious*
  - *Collard greens and other salty leafy greens*
- **Conclusion:**
  - **Ong choi is a leafy green like spinach, chard, or collard greens**
  - **We could conclude this based on words like "*leaves*" and "*delicious*" and "*sautéed*"**

Computational Models of Meaning

- So:
  - words are defined by their environments (the words around them)

- How can we represent word meaning with word environment?
  - Vector semantics

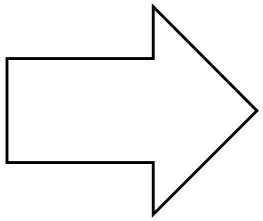
Vector Semantics: Two Ideas

- **Idea 1:**
  - **Let's define the meaning of a word by its distribution in language use (neighboring words or grammatical environments)**

- **Idea 2:**
  - **Let's define the meaning of a word as a point in space**

Bag of Words: Strings Representation

Some document:
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

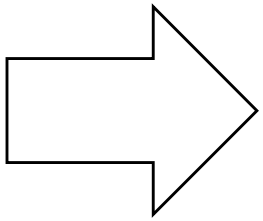


Word:	Frequency:
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
times	1
....	...

Bag of words assumption: word/token position does not matter.

Bag of Words: Meaning Ignored!

Some document:
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



Word:	Frequency:
it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
whimsical	1
times	1
....	...

Bag of words assumption: word/token position does not matter.

Connotation as a Point in Space

- **Words seem to vary along three affective DIMENSIONS:**
  - **valence: the pleasantness of the stimulus**
  - **arousal: the intensity of emotion provoked by the stimulus**
  - **dominance: the degree of control exerted by the stimulus**

	Word	Score		Word	Score
valence	love	1.000		toxic	0.008
	happy	1.000		nightmare	0.005
arousal	elated	0.960		mellow	0.069
	frenzy	0.965		napping	0.046
dominance	powerful	0.991		weak	0.045
	leadership	0.983		empty	0.081

Source: NRC VAD Lexicon (https://saifmohammad.com/WebPages/nrc-vad.html)

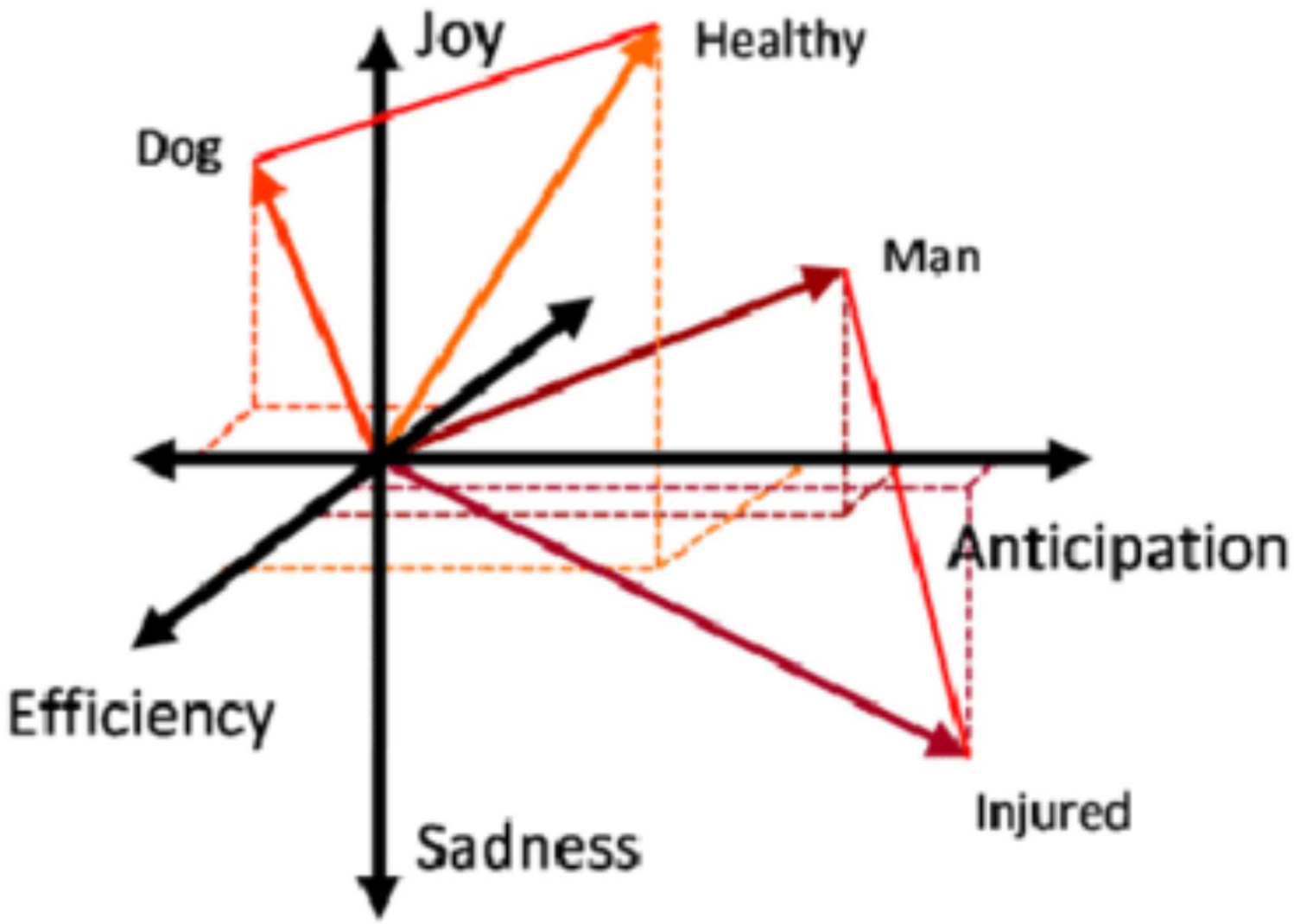
Vector Semantics

- The idea:
  - represent a word as a point in a multidimensional semantic space that is derived from the distributions of word neighbors

Point in Space Based on Distribution

- Each word = a vector
  - not just "*good*" or "$word_{45}$"
- Similar words: "nearby in semantic space"
- We build this space automatically by seeing which words are nearby in text

Vector Semantics: Words as Vectors



Source: Signorelli, Camilo & Arsiwalla, Xerxes. (2019). Moral Dilemmas for Artificial Intelligence: a position paper on an application of Compositional Quantum Cognition

Word Embedding: Definition

Word Embedding:

*a term used for the **representation of words** for text analysis, **typically in the form of a real-valued vector** that **encodes the meaning of the word such** that the **words that are closer in the vector space are expected to be similar** in meaning*

from Wikipedia

Word Embedding

- **Embedding:**
  - **“embedded into a space”**
  - **mapping from one space or structure to another**
- **The standard way to represent meaning in NLP**
- **Fine-grained model of meaning for similarity**

The Why: Sentiment Analysis

- Using **words only**:
  - a feature is a word identity
  - for example
    - feature $x_5 = \begin{cases} 1 \; if \; the \; previous \; word \; was \; 'terrible' \\ 0 \; otherwise \end{cases}$
  - requires exact same word to be in training and test

The Why: Sentiment Analysis

- Using **embeddings**:
  - a feature is a word vector
  - the previous word was vector $[35, 22, 17]$
  - now in the test set we might see a similar vector $[34, 21, 14]$
  - we can **generalize** to **similar but unseen** words

Term-Document Matrix

- **Each document is represented by a vector of words**

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Term-Document Matrix

- **Vectors are similar for the two comedies**
  - **"As you like it" and "Twelfth Night"**

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- **But comedies are different than the other two**
  - **more** *fools* **and** *wit* **and fewer** *battles*

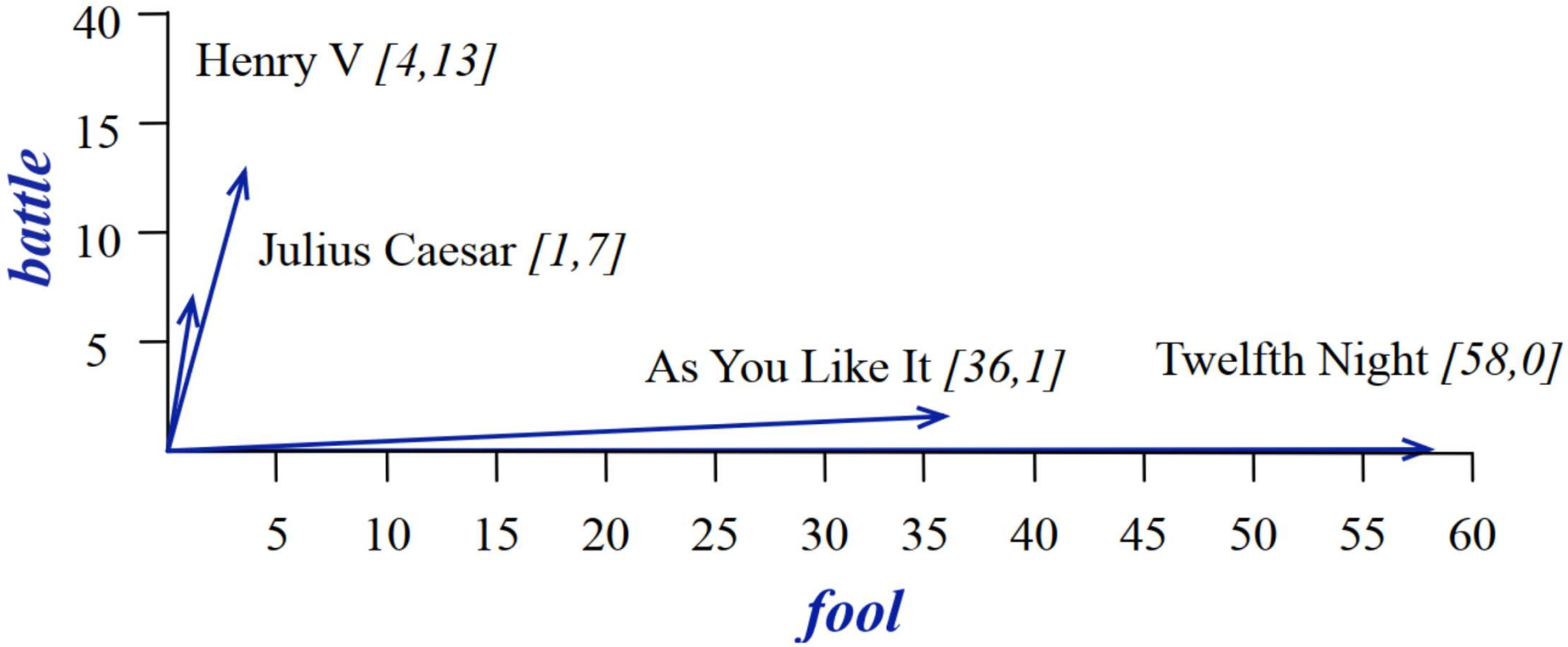
Term-Document Matrix

- **Vectors are similar for the two comedies**
  - **"As you like it" and "Twelfth Night"**

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- **But comedies are different than the other two**
  - **more** *fools* **and** *wit* **and fewer** *battles*

Document Vector Visualization



Words as Vectors

- *battle* **is "the kind of word that occurs in Julius Caesar and Henry V"**

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

- *fool* **is "the kind of word that occurs in comedies, especially Twelfth Night"**

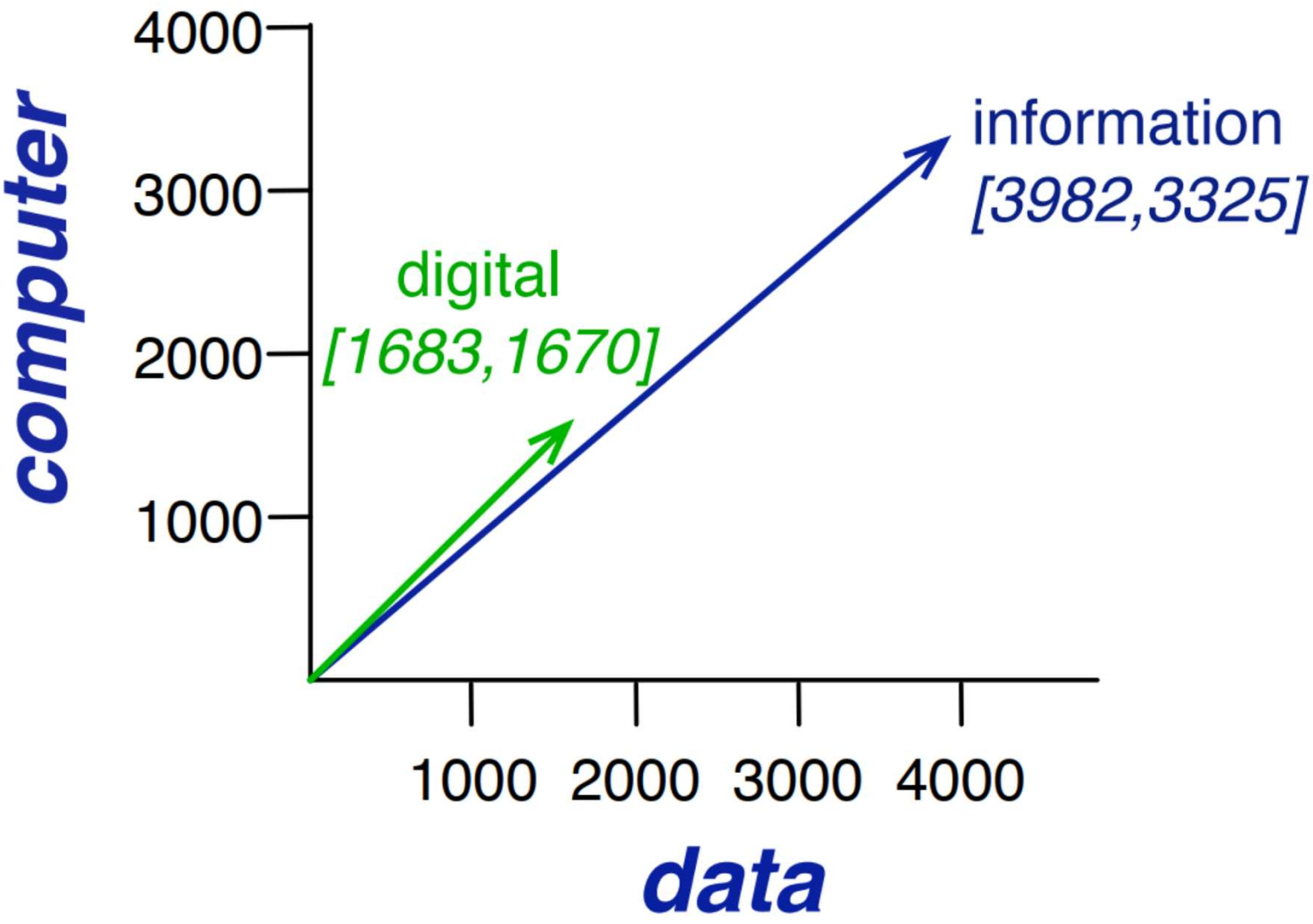
Word-Word (Term-Context) Matrix

- **Two words are similar in meaning if their context vectors are similar**

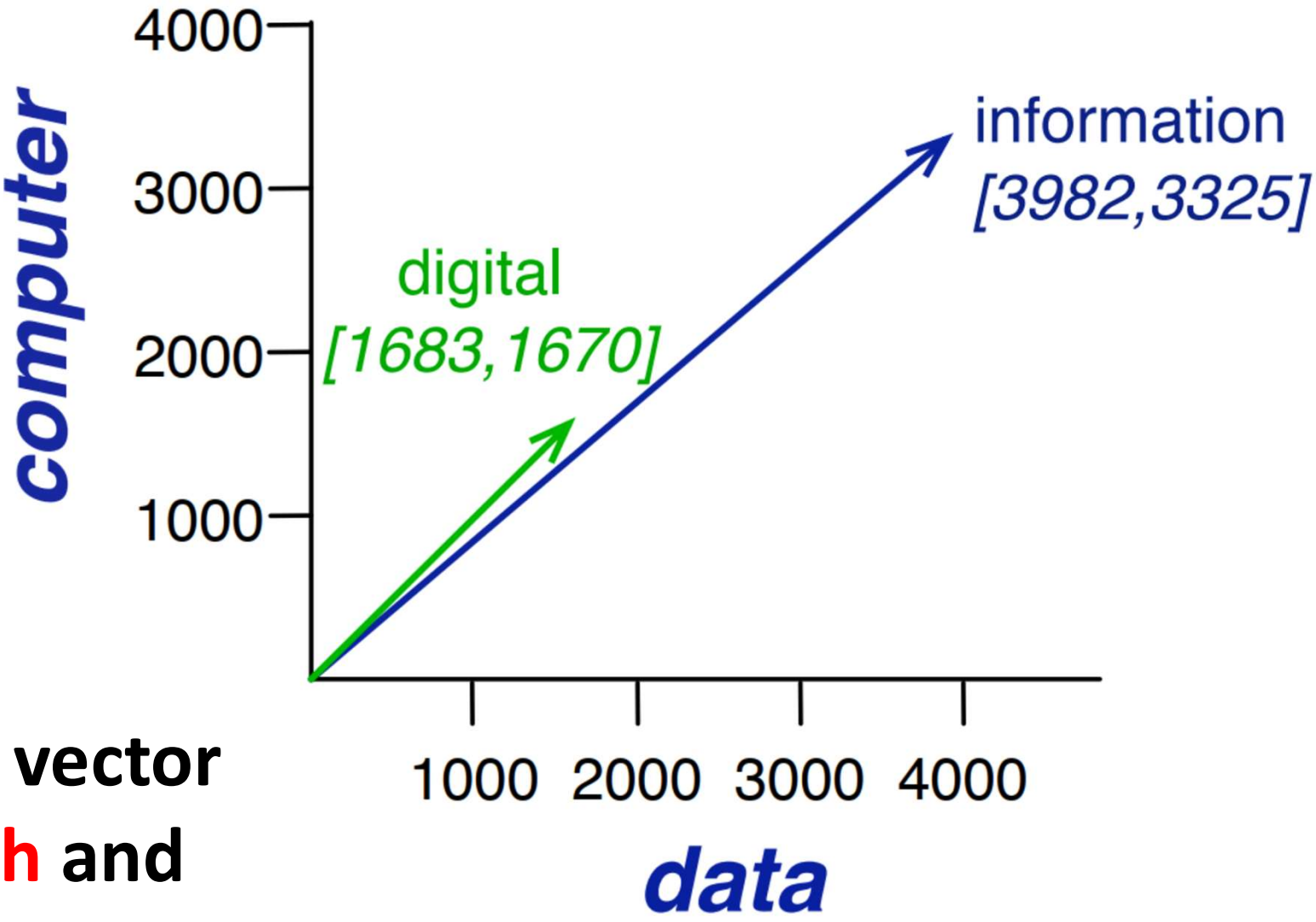
is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	**computer**	**data**	**result**	**pie**	**sugar**	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Document Vector Visualization



Document Vector Visualization



Note vector length and direction

Vector Dot / Scalar Product

Given two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ (N - vector space dimension):

$$\boldsymbol{a} = [a_1, a_2, \ldots, a_N] \text{ and } \boldsymbol{b} = [b_1, b_2, \ldots, b_N]$$

their vector dot/scalar product is:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{N} a_i * b_i = a_1 * b_1 + a_2 * b_2 + \ldots + a_N * b_N$$

Using matrix representation:

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^{\boldsymbol{T}} = [a_1 \quad a_2 \quad \cdots \quad a_N] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix}$$

Vector Dot / Scalar Product

- **Vector dot/scalar product is a scalar:**

$$\boldsymbol{a} \cdot \boldsymbol{b} = \sum_{i=1}^{N} a_i * b_i = a_1 * b_1 + a_2 * b_2 + \ldots + a_N * b_N$$

- **Vector dot/scalar:**
  - **high values when the two vectors have large values in the same dimensions**
  - **useful similarity measure**

Vector Dot / Scalar Product: Problem

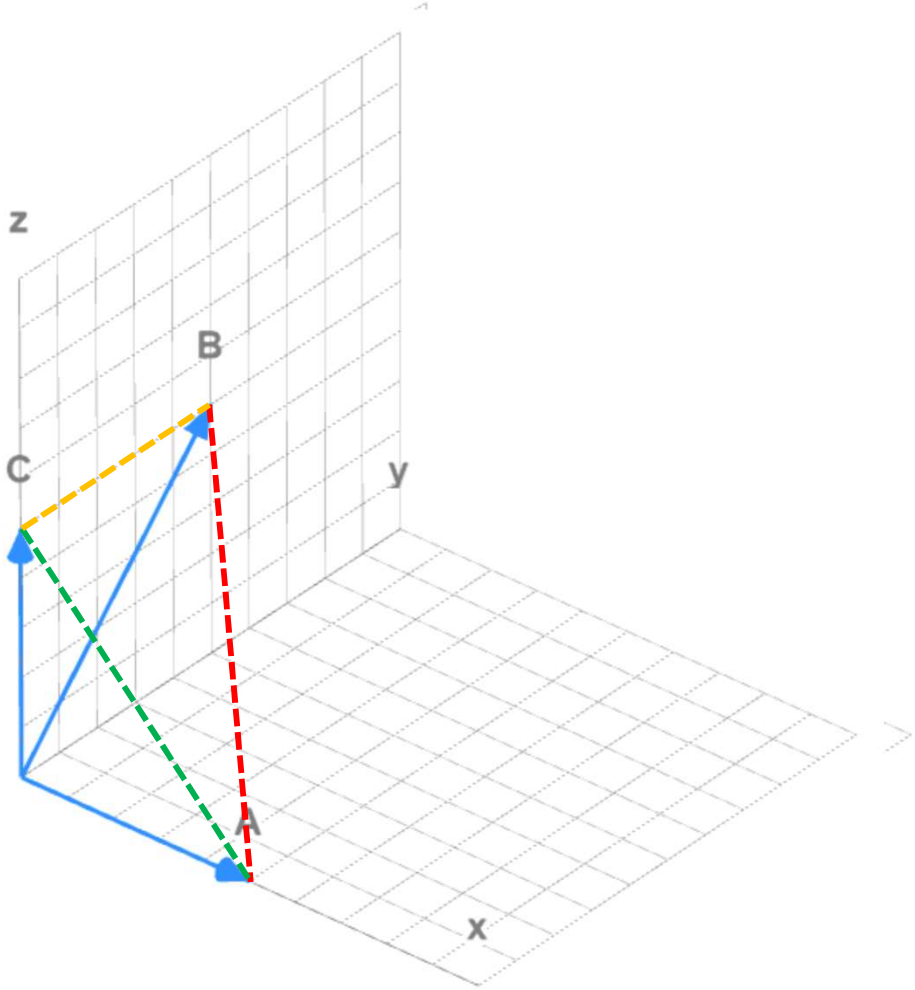
- Dot product favors long vectors: higher if a vector is longer (has higher values in many dimension)
- Vector length:

$$|\boldsymbol{a}| = \sqrt{\sum_{i=1}^{N} a_i^2}$$

- Frequent words (of, the, you) have long vectors (since they occur many times with other words).
- dot product overly favors frequent words

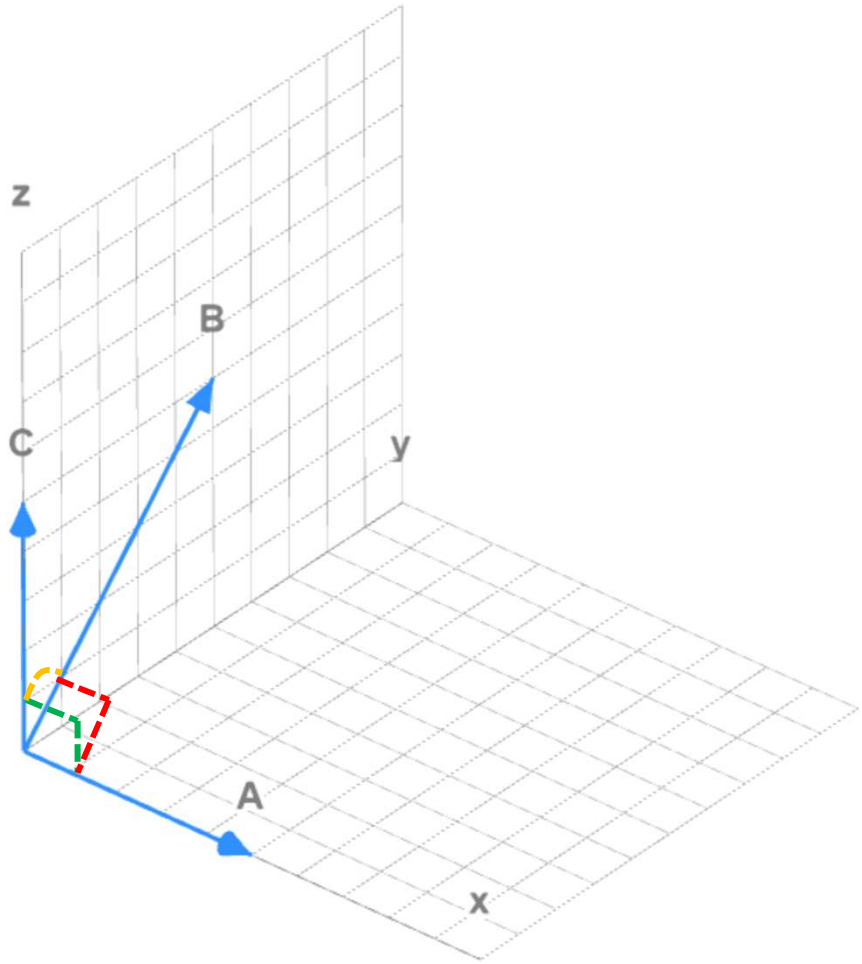
Alternative: Cosine Similarity

Euclidean distance



$$D(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2)}$$

Cosine similarity



$$D(x,y) = cos(\theta) = \frac{x \cdot y}{\|x\| \, \|y\|}$$

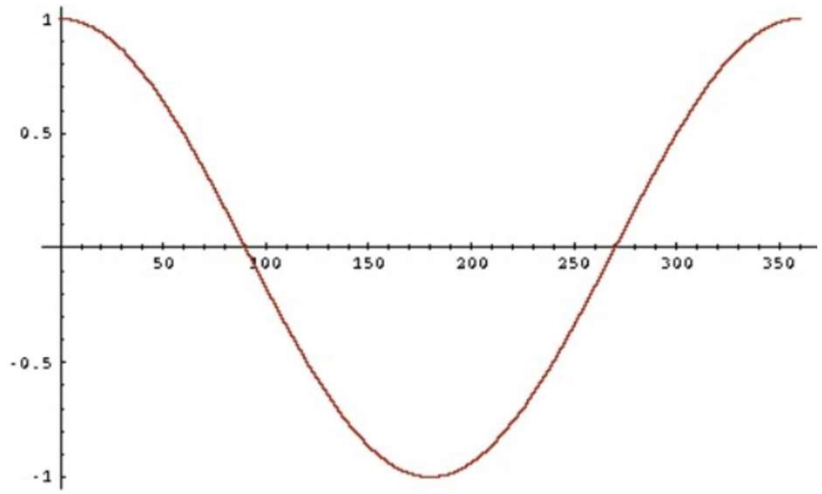
Word Similarity | Cosine Similarity

$$\text{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}}$$

Where: v and w are two different word vectors

Word Similarity | Cosine Similarity

- -1: vectors point in opposite directions
- +1: vectors point in same directions
- 0: vectors are orthogonal



But since raw frequency values are non-negative, the cosine for term-term matrix vectors ranges from 0–1

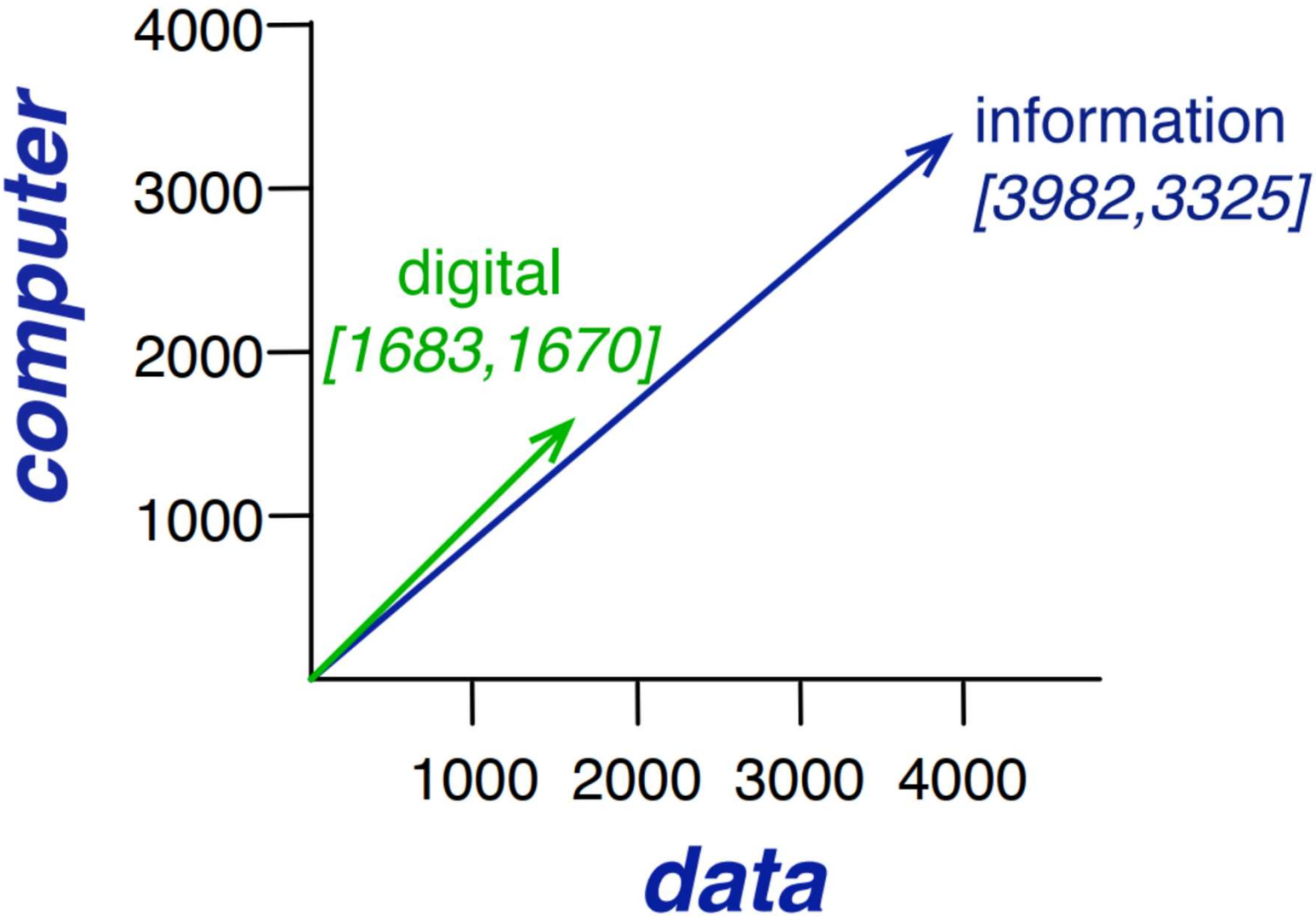
Word Similarity

- **Two words are similar in meaning if their context vectors are similar**

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Word Similarity Visualization



Word Similarity | Cosine Similarity

$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}\bullet\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|}\bullet\frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325

$$\cos(\text{cherry},\text{information}) = \frac{442*5+8*3982+2*3325}{\sqrt{442^2+8^2+2^2}\sqrt{5^2+3982^2+3325^2}} = .017$$

Low similarity

$$\cos(\text{digital},\text{information}) = \frac{5*5+1683*3982+1670*3325}{\sqrt{5^2+1683^2+1670^2}\sqrt{5^2+3982^2+3325^2}} = .996$$

High similarity

Cosine Similarity Visualization