innovate achieve lead

# Natural Language Processing

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in

BITS Pilani
Pilani Campus

## Session Content

**Vector Semantics and Word Embedding (Chapter 6 Jurafsky and Martin Book)**

- Lexical semantics
- Vector semantics
- Word and Vectors
- TFIDF
- Word2Vec
  - Skip gram
  - CBOW
- Glove
- Visualizing Embedding’s

# Lexical semantics

- **Structure** of words : **morphology**
- **Distribution** of words: **language modeling**
- **Meaning** of words: **lexical semantics**
- **Distributional hypothesis:** a way to identify words with similar meanings
  - Words that occur in **similar contexts** tend to have **similar meanings**
  - E.g. oculist and eye-doctor occur near words like eye or examined
  - Amount of meaning difference between two words "corresponding roughly to the amount of difference in their environments (Context)"
  - two words that occur in very similar distributions (whose neighboring words are similar) have similar meanings.
  - **Basic idea: Measure the semantic similarity of words in terms of the similarity of the contexts in which they appear**

## Applications

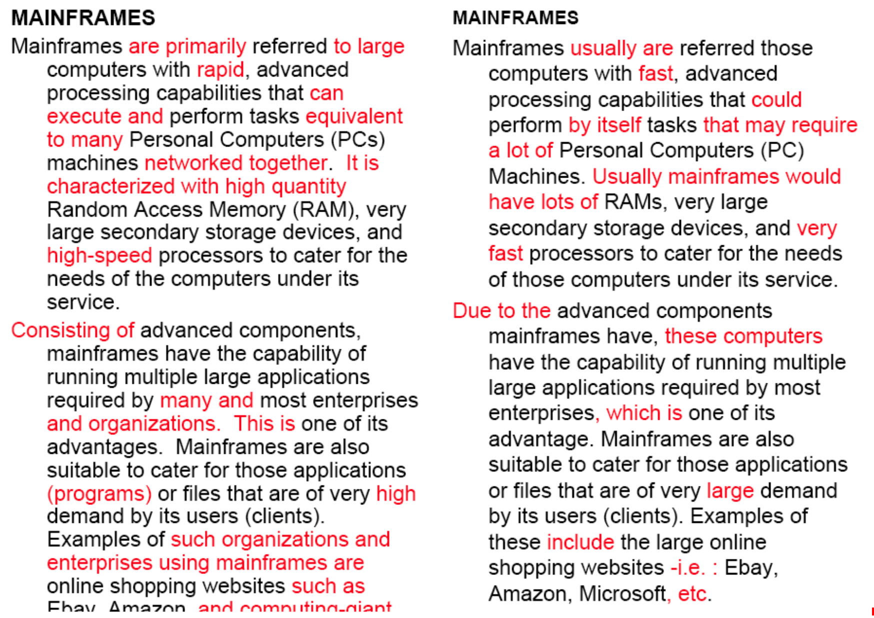
**Question answering:**

Q: “How ***tall*** is Mt. Everest?”

Ans: “The official ***height*** of Mount Everest is 29029 feet”

***“tall” is similar to “height”***

**Plagiarism detection**



**MAINFRAMES**

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as Ebay, Amazon, and computing-giant

**MAINFRAMES**

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

## What do words mean?

- N-gram or text classification methods we've seen so far
  - Words are just strings (or indices $w_i$ in a vocabulary list)
  - That's not very satisfactory!
- Introductory logic classes:
  - The meaning of "dog" is DOG; cat is CAT

    $\forall x\ \mathrm{DOG}(x) \longrightarrow \mathrm{MAMMAL}(x)$
- Old linguistics joke by Barbara Partee in 1967:
  - Q: What's the meaning of life?
  - A: LIFE
- That seems hardly better!

## Lemmas and senses

lemma

mouse (N)

sense

1. any of numerous small rodents...
2. a hand-operated device that controls a cursor...

Modified from the online thesaurus WordNet

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

## Lexical semantics

- lexical semantics, the linguistic **study of word meaning**
- Concepts or **word senses**
  - Have a complex many-to-many association with words
- Have relations with each other
  - Synonymy
  - Similarity
  - Relatedness: semantic field and frame
  - Antonymy
  - Connotation
- **More generally, a model of word meaning should allow us to draw inferences to address meaning-related tasks like question-answering or dialogue.**

## Relations between senses: **Synonymy**

- Synonyms have the same meaning in some or all contexts.
  - Same word sense, substitutable for one another, same propositional meaning, truth preserving
  - E.g couch/sofa, vomit/throw up, car/automobile

- water/H20 : "H20" in a surfing guide?
- big/large: my big sister != my large sister

- In practice, the word synonym is therefore used to describe a relationship of approximate or rough synonymy,

## Relation: **Similarity**

- Words with similar meanings. Not synonyms, but sharing some element of meaning (Cat is not a synonym of dog, but cats and dogs are certainly similar words)
  - car, bicycle ; cow, horse
- The notion of word similarity is very useful in larger semantic tasks.
  - Help in computing how similar the meaning of two phrases or sentences are
  - Applications : question answering, paraphrasing, and summarization.
- Humans to judge how similar one word is to another

| word1 | word2 | similarity |
|---|---|---|
| vanish | disappear | 9.8 |
| behave | obey | 7.3 |
| belief | impression | 5.95 |
| muscle | bone | 3.65 |
| modest | flexible | 0.98 |
| hole | agreement | 0.3 |

SimLex-999 dataset (Hill et al., 2015)

## Relation: Word relatedness

- Also called "word association"
- Words can be related in any way, perhaps via a semantic frame or field
  - `coffee, tea:` **similar**
  - `coffee, cup:` **related**, not similar

## 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*

## Relation: Antonymy

- Senses that are opposites with respect to only one feature of meaning
- Otherwise, they are very similar!

  `dark/light` `short/long` `fast/slow` `rise/fall`

  `hot/cold` `up/down` `in/out`

- 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`

## Connotation (sentiment)

- Words have affective meanings
  - Positive connotations (happy)
  - Negative connotations (sad)
- Evaluation (sentiment!)
  - Positive evaluation (great, love)
  - Negative evaluation (terrible, hate)

# Connotation

Words seem to vary along 3 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

Osgood et al. (1957)

| | 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 |

Values from NRC VAD Lexicon (Mohammad 2018)

## Computational models of word meaning

Can we build a theory of how to represent word meaning?

We'll introduce **vector semantics**

- The standard model in language processing!
- Handles many of our goals!

## Let's define words by their usages

One way to define "usage":

words are defined by their environments (the words around them)

Zellig Harris (1954):

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

## What does recent English borrowing *ongchoi* mean?

- 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:
  - Ongchoi is a leafy green like spinach, chard, or collard greens
    - We could conclude this based on words like "leaves" and "delicious" and "sauteed"

## Vector Semantics

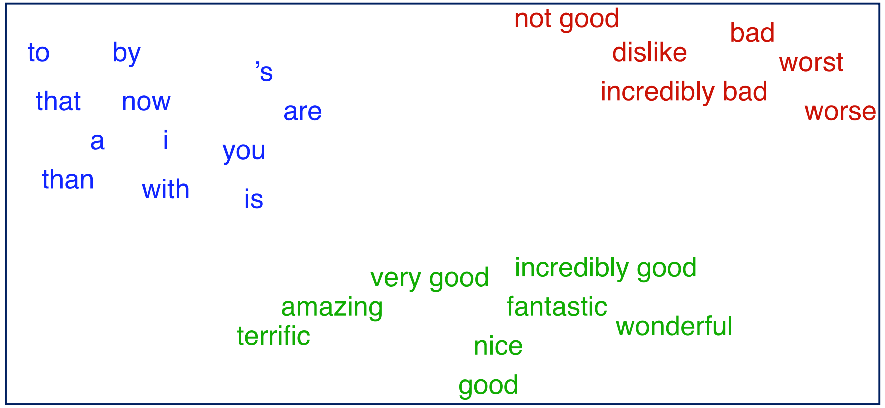
Idea 1: Defining meaning by linguistic distribution

Idea 2: Meaning as a point in multidimensional space

# Word embeddings

- Vectors for representing words are called embeddings
- Each word = a vector (not just "good" or "$w_{45}$")
- Defining meaning as a point in space based on distribution
- Similar words are "**nearby in semantic space**"
- **Every modern NLP algorithm uses embeddings as the representation of word meaning**

two-dimensional (t-SNE) projection of embeddings



## We define meaning of a word as a vector

- Called an "embedding" because it's embedded into a space
- The standard way to represent meaning in NLP
- **Every modern NLP algorithm uses embeddings as the representation of word meaning**
- Fine-grained model of meaning for similarity

## Intuition: why vectors?

- Consider sentiment analysis:
  - With **words**, a feature is a word identity
    - Feature 5: 'The previous word was "terrible"'
    - requires **exact same word** to be in training and test
  - With **embeddings**:
    - 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!!!

# Word embeddings: types

1. Frequency based Embedding
   - A common baseline model
   - **Sparse** long vectors
   - Words are represented by (a simple function of) the **counts** of nearby words
   - **dimensions** corresponding to **words** in the vocabulary or **documents** in a collection
   - **Count Vector**
   - **TF-IDF Vector**
   - **Co-Occurrence Vector**
2. Prediction based Embedding
   - **Dense** vectors
   - Representation is created by training a classifier to **predict** whether a word is likely to appear nearby
   - **Word2vec: Skip-gram and CBOW**
   - **GloVe**

# Vectors are the basis of information retrieval

Each document is represented by a vector of words

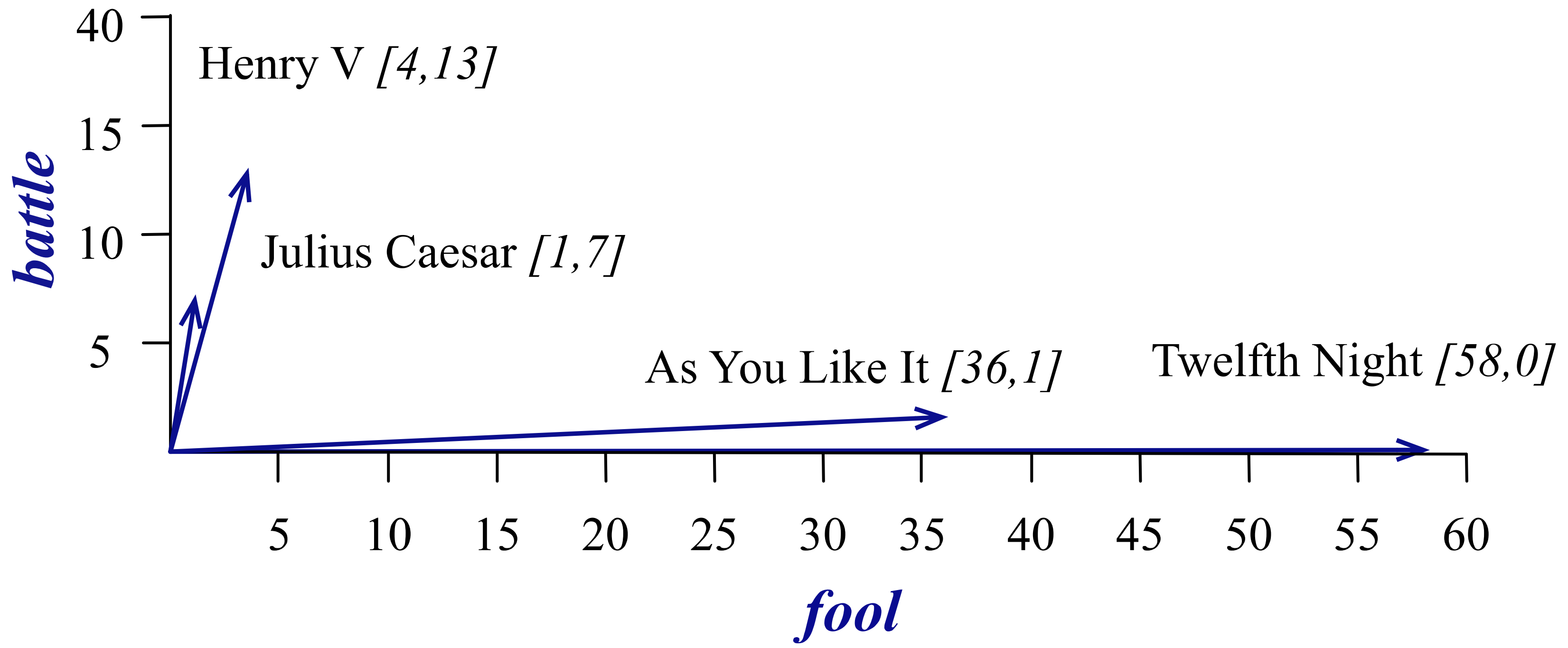
| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| battle | 1 | 0 | 7 | 13 |
| good | 114 | 80 | 62 | 89 |
| fool | 36 | 58 | 1 | 4 |
| wit | 20 | 15 | 2 | 3 |

Vectors are similar for the two comedies

But comedies are different than the other two

Comedies have more *fools* and *wit* and fewer *battles*.

# Visualizing document vectors



## Idea for word meaning: Words can be vectors too!!!

| | 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 |

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

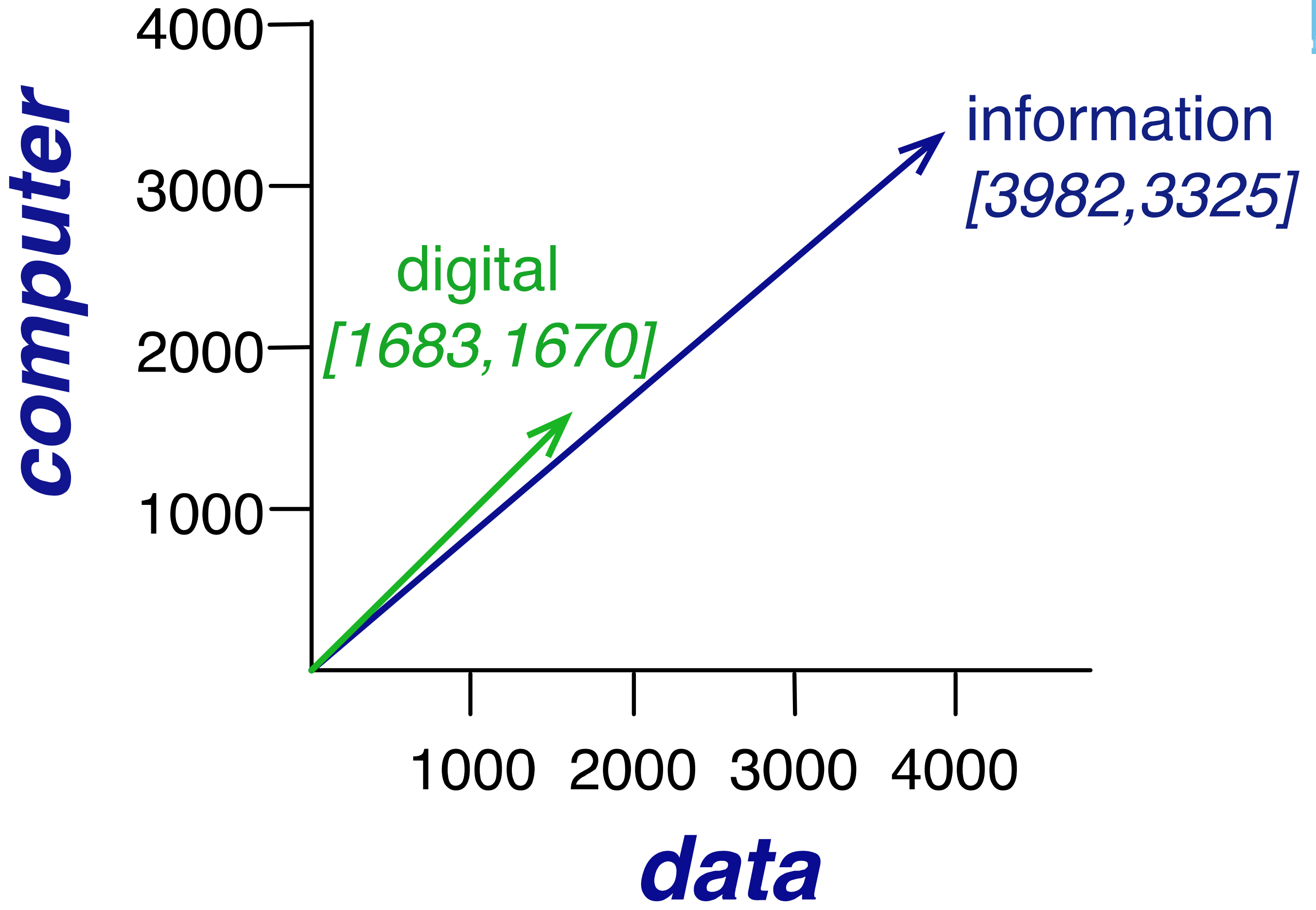
*fool* is "the kind of word that occurs in comedies, especially Twelfth Night"

## More common: word-word matrix (or "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 | ... |



computer
4000
3000
2000
1000
digital
[1683,1670]
information
[3982,3325]
1000
2000
3000
4000
data

## Computing word similarity: Dot product and cosine

- The dot product between two vectors is a scalar:

$$\text{dot product}(\mathbf{v},\mathbf{w}) = \mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

- The dot product tends to be high when the two vectors have large values in the same dimensions
- Dot product can thus be a useful similarity metric between vectors

## Problem with raw dot-product

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

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^{N} v_i^2}$$

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

## Alternative: cosine for computing word similarity

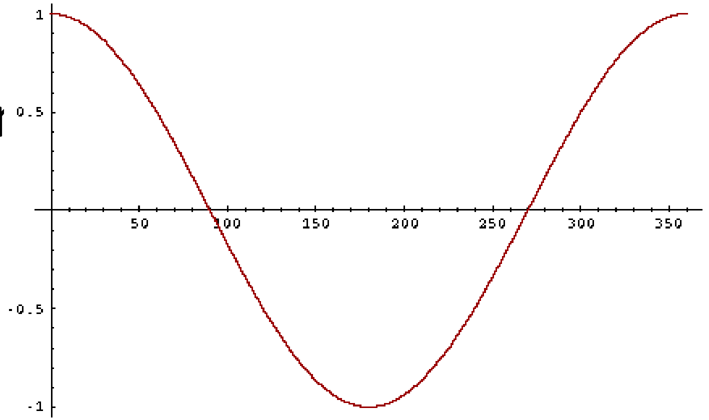
$$\text{cosine}(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

Based on the definition of the dot product between two vectors a and b

$$\begin{aligned} \mathbf{a}\cdot\mathbf{b} &= |\mathbf{a}||\mathbf{b}|\cos\theta \\ \frac{\mathbf{a}\cdot\mathbf{b}}{|\mathbf{a}||\mathbf{b}|} &= \cos\theta \end{aligned}$$

## Cosine as a similarity metric

- -1: vectors point in opposite directior
- +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

# Cosine examples

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

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

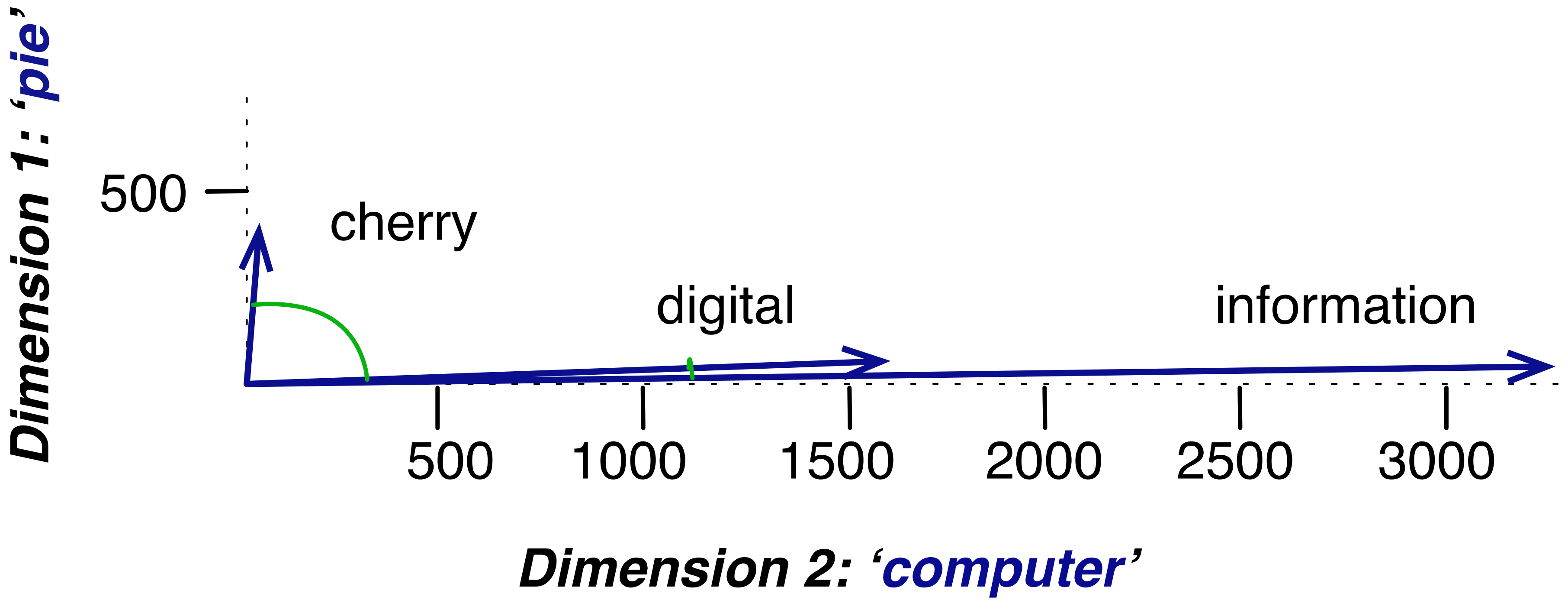
$$\cos(\text{cherry},\text{information}) =$$

$$\frac{442 * 5 + 8 * 3982 + 2 * 3325}{\sqrt{442^2 + 8^2 + 2^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .017$$

$$\cos(\text{digital},\text{information}) =$$

$$\frac{5 * 5 + 1683 * 3982 + 1670 * 3325}{\sqrt{5^2 + 1683^2 + 1670^2}\sqrt{5^2 + 3982^2 + 3325^2}} = .996$$

## Visualizing cosines (well, angles)



# Tf-idf

- Raw frequency is a bad representation
- The co-occurrence matrices we have seen represent each cell by word frequencies.
- Frequency is clearly useful; if *sugar* appears a lot near *apricot*, that's useful information.
- But overly frequent words like *the, it,* or *they* are not very informative about the context
- It's a paradox! How can we balance these two conflicting constraints?
- when the dimensions are document word weighting using tf-idf
  - **tf-idf:** tf-idf value for word t in document d:

$$\boxed{w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t}$$

  - Words like "the" or "it" have very low idf

## Term frequency (tf)

$$\mathbf{tf}_{t,d} = \mathbf{count}(t,d)$$

Instead of using raw count, we squash a bit:

$$\mathbf{tf}_{t,d} = \mathbf{log}_{10}(\mathbf{count}(t,d)+1)$$

## Document frequency (df)

- The second factor in ***tf-idf*** is used to give a higher weight to words that occur only in a few documents
- $df_t$ is the number of documents $t$ occurs in.
- (note this is not collection frequency: total count across all documents)
- "Romeo" is very distinctive for one Shakespeare play:

| | **Collection Frequency** | **Document Frequency** |
|---|---|---|
| Romeo | 113 | 1 |
| action | 113 | 31 |

# Inverse document frequency (idf)

$$\text{idf}_t = \log_{10}\left(\frac{N}{\text{df}_t}\right)$$

N is the total number of documents in the collection

**The fewer documents in which a term occurs, the higher this weight**

| Word | df | idf |
|---|---|---|
| Romeo | 1 | 1.57 |
| salad | 2 | 1.27 |
| Falstaff | 4 | 0.967 |
| forest | 12 | 0.489 |
| battle | 21 | 0.246 |
| wit | 34 | 0.037 |
| fool | 36 | 0.012 |
| good | 37 | 0 |
| sweet | 37 | 0 |

# Final tf-idf weighted value for a word

Raw counts:

| | 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 |

tf-idf: $\boxed{w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t}$

**Word: *wit***

tf = $\log_{10}(20+1)$ = 1:322, idf = $\log_{10}(37/34)$=0.037, tf-idf value=0.049

**Word: *good***

appears in every document so tf-idf = 0

| | As You Like It | Twelfth Night | Julius Caesar | Henry V |
|---|---|---|---|---|
| **battle** | 0.074 | 0 | 0.22 | 0.28 |
| **good** | 0 | 0 | 0 | 0 |
| **fool** | 0.019 | 0.021 | 0.0036 | 0.0083 |
| **wit** | 0.049 | 0.044 | 0.018 | 0.022 |

# Sparse versus dense vectors

- tf-idf (or PMI) vectors are
  - **long** (length |V|= 20,000 to 50,000)
  - **sparse** (most elements are zero)
- Alternative: learn vectors which are
  - **short** (length 50-1000)
  - **dense** (most elements are non-zero)

## References

- Ch-6 : Speech and Language Processing Daniel Jurafsky and James H. Martin
- https://jalammar.github.io/illustrated-word2vec/
- http://web.stanford.edu/class/cs224n/slides/cs224n-2022-lecture02-wordvecs2.pdf
- https://arxiv.org/pdf/1301.3781.pdf
- https://nlp.stanford.edu/pubs/glove.pdf
- https://jonathan-hui.medium.com/nlp-word-embedding-glove-5e7f523999f6
- www.deeplearning.ai

## References

Word embedding

- https://www.youtube.com/watch?v=ERibwqs9p38
- https://www.youtube.com/watch?v=EsfNYiLVtHI&t=10s
- https://www.youtube.com/watch?v=lrPxo-92GC0
- https://www.coursera.org/lecture/nlp-sequence-models/
- https://www.youtube.com/watch?v=UqRCEmrv1gQ
- https://www.youtube.com/watch?v=g7wEfamF0Eg
- https://www.youtube.com/watch?v=Sho-b4-9ODE