Natural Language Processing: Word Semantics

18 February 2022

Naive Bayes: Another Example Clustering

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

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c) = \frac{3}{4}$

$P(j) = \frac{1}{4}$

Conditional Probabilities:

P(Chinese|c) = (5+1) / (8+6) = 6/14 = 3/7

P(Tokyo|c) = (0+1) / (8+6) = 1/14

P(Japan|c) = (0+1) / (8+6) = 1/14

P(Chinese|j) = (1+1) / (3+6) = 2/9

P(Tokyo|j) = (1+1) / (3+6) = 2/9

P(Japan|j) = (1+1) / (3+6) = 2/9

Choosing a class:

P(c|d5) $\propto$ 3/4 * $(3/7)^3$ * 1/14 * 1/14
≈ 0.0003

P(j|d5) $\propto$ 1/4 * $(2/9)^3$ * 2/9 * 2/9
≈ 0.0001

Word Meaning

Previous lectures

- In anything we have seen so far, words were just strings (or indices in a vocabulary list)
- Not very satisfactory..

What do words mean?

First thought: look in a dictionary, e.g.,

http://www.oed.com/

Reminder: lemma and wordform

- A **lemma** or **citation form**
  - Same stem, part of speech, rough semantics
- A **wordform**
  - The “inflected” word as it appears in text

Wordform	Lemma
banks	bank
sung	sing
duermes	dormir

Words, Lemmas, Senses, Definitions

lemma sense definition

pepper, *n.*

Pronunciation: Brit. /ˈpɛpə/ , U.S. /ˈpɛpər/

Forms: OE **peopor** (*rare*), OE **pipcer** (transmission error), OE **pipor**, OE **pipur** (*rare*

Frequency (in current use):

Etymology: A borrowing from Latin. **Etymon:** Latin *piper*.

< classical Latin *piper*, a loanword < Indo-Aryan (as is ancient Greek *πέπερι*); compare Sa

I. The spice or the plant.

1.

a. A hot pungent spice derived from the prepared fruits (peppercorns) of the pepper plant, *Piper nigrum* (see sense 2a), used from early times to season food, either whole or ground to powder (often in association with salt). Also (locally, chiefly with distinguishing word): a similar spice derived from the fruits of certain other species of the genus *Piper*; the fruits themselves.

The ground spice from *Piper nigrum* comes in two forms, the more pungent *black pepper*, produced from black peppercorns, and the milder *white pepper*, produced from white peppercorns: see BLACK *adj.* and *n.* Special uses 5a, PEPPERCORN *n.* 1a, and WHITE *adj.* and *n.*[1] Special uses 7b(a).

2.

a. The plant *Piper nigrum* (family Piperaceae), a climbing shrub indigenous to South Asia and also cultivated elsewhere in the tropics, which has alternate stalked entire leaves, with pendulous spikes of small green flowers opposite the leaves, succeeded by small berries turning red when ripe. Also more widely: any plant of the genus *Piper* or the family Piperaceae.

b. Usu. with distinguishing word: any of numerous plants of other families having hot pungent fruits or leaves which resemble pepper (1a) in taste and in some cases are used as a substitute for it.

c. *U.S.* The California pepper tree, *Schinus molle*. Cf. PEPPER TREE *n.* 3.

3. Any of various forms of capsicum, esp. *Capsicum annuum* var. *annuum*. Originally (chiefly with distinguishing word): any variety of the *C. annuum* Longum group, with elongated fruits having a hot, pungent taste, the source of cayenne, chilli powder, paprika, etc., or of the perennial *C. frutescens*, the source of Tabasco sauce. Now frequently (more fully ***sweet pepper***): any variety of the *C. annuum* Grossum group, with large, bell-shaped or apple-shaped, mild-flavoured fruits, usually ripening to red, orange, or yellow and eaten raw in salads or cooked as a vegetable. Also: the fruit of any of these capsicums.

Sweet peppers are often used in their green immature state (more fully ***green pepper***), but some new varieties remain green when ripe.

Lemma pepper

Sense 1: spice from pepper plant
Sense 2: the pepper plant itself
Sense 3: another similar plant (Jamaican pepper)
Sense 4: another plant with peppercorns (California pepper)
Sense 5: *capsicum* (i.e. chili, paprika, bell pepper, etc.)

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

Homonymy

Homonyms: words that share a form but have **unrelated**, distinct meanings:

- $bank_1$: financial institution, $bank_2$: sloping land
- bat_1: club for hitting a ball, bat_2: nocturnal flying mammal

1. Homographs (bank/bank, bat/bat)
2. Homophones:
   1. Write and right
   2. Piece and peace

Homonymy causes problems for NLP applications

- Information retrieval
  - "`bat care`"
- Machine Translation
  - `bat`: murciélago (animal) or bate (for baseball)
- Text-to-Speech
  - `bass` (stringed instrument) vs. `bass` (fish)

Polysemy

1. The **bank** was constructed in 1875 out of local red brick.
2. I withdrew the money from the **bank**

- Are those the same sense?
  - Sense 2: “A financial institution”
  - Sense 1: “The building belonging to a financial institution”
- A **polysemous** word has **related** meanings
  - Most non-rare words have multiple meanings

Metonymy or Systematic Polysemy: A systematic relationship between senses

- Many types of polysemy are systematic
  - `School, university, hospital, White House, etc.`
  - All can mean the institution or the building
- A systematic relationship:
  - Building ⟺ Organization
- Other such kinds of systematic polysemy:

Author (`Jane Austen wrote Emma`)

⟺ Works of Author (`I love Jane Austen`)

Tree (`Plums have beautiful blossoms`)

⟺ Fruit (`I ate a preserved plum`)

How do we know when a word has more than one sense?

- The "zeugma" test: Two senses of `serve`?
  - `Which flights` **`serve`** `breakfast?`
  - `Does Lufthansa` **`serve`** `Philadelphia?`
  - ?Does Lufthansa serve breakfast and Philadelphia?
- Since this conjunction sounds weird,
  - we say that these are **two different senses of "serve"**

Relations between senses: Synonymy

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

Two lemmas are synonyms

- if they can be substituted for each other in all situations.
- If so they have the same **propositional meaning**

Synonymy

- Note that there are probably no examples of perfect synonymy
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, context, genre, etc.

Synonymy?

water/H_20
big/large
brave/courageous

- "H_20" in a surfing guide?
- “How big is that plane?” vs. “Would I be flying on a large or small plane?”
- “my big sister” or “my large sister”? (*big has a sense that means being older, or grown up, while large lacks this sense*)

Relation: Similarity

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

`car, bicycle`

`cow, horse`

Why word similarity?

- Information retrieval
- Question answering
- Machine translation
- Natural language generation
- Language modeling
- Automatic essay grading
- Plagiarism detection
- Document clustering
- …

Asking humans how similar two words are

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)

Felix Hill, KyungHyun Cho, Sebastien Jean, Coline Devin and Yoshua Bengio. Embedding Word Similarity with Neural Machine Translation. ICLR. 2015.

SimLex-999 vs. WordSim-353

- SimLex-999 provides a way of measuring how well models capture *similarity,* rather than *relatedness* or *association*
- The scores in SimLex-999 therefore differ from other well-known evaluation datasets such as *WordSim-353* [1]

Pair	**Simlex-999 rating**	**WordSim-353 rating**
coast - shore	9.00	9.10
clothes - closet	1.96	8.00

[1] Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. (2002). Placing Search in Context: The Concept Revisited. ACM Transactions on Information Systems, 20(1):116-131.

Relation: Word relatedness/association

- Also called "word association"
- Words can be related in any way, perhaps via a semantic frame or field
  - `car, bicycle`: **similar**
  - `car, gasoline`: **related** (not similar)

Semantic field

- Words that
  - cover a particular semantic domain and
  - 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 frame

- A semantic frame is a set of words that denote perspectives or participants in a particular type of event
- E.g., a commercial transaction - a kind of event in which one entity trades money to another entity in return for some good or service, after which the good changes hands (or perhaps the service is performed)
  - Lexical encoding:
    - *buy* (the event from the perspective of the buyer), *sell* (from the perspective of the seller), *pay* (focusing on the monetary aspect), or nouns like *buyer*, *seller*
    - Frames have semantic roles (like *buyer*, *seller*, *goods*, *money*), and words in a sentence can take on these roles
    - Knowing the roles of *buy* and *sell* in their semantic frame one can paraphrase:
      - Sam bought the book from Ling → Ling sold the book to Sam

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`

Embedding models like word2vec tend to place antonyms near each other in the vector space, thus we may use also thesauri to distinguish antonyms when using such models

Hyponymy and Hypernymy

- One sense is a **hyponym/subordinate** of another if the first sense is more specific, denoting a subclass of the other
  - *car* is a hyponym of *vehicle*
  - *mango* is a hyponym of *fruit*
- Conversely **hypernym/superordinate** ("hyper is super")
  - *vehicle* is a **hypernym** of *car*
  - *fruit* is a hypernym of *mango*

Superordinate/hyper	vehicle	fruit	furniture
Subordinate/hyponym	car	mango	chair

Hyponymy more formally

- Extensional:
  - The class denoted by the superordinate extensionally includes the class denoted by the hyponym
- Entailment:
  - A sense A is a hyponym of sense B if *being an A* entails *being a B*
- Hyponymy is usually transitive
  - (A hypo B and B hypo C entails A hypo C)
- Another name: the **IS-A hierarchy**
  - A IS-A B
  - B **subsumes** A

Hyponyms and Instances

- WordNet has both **classes** and **instances**
- An **instance** is an individual, a proper noun that is a unique entity
    - `San Francisco` is an **instance** of `city`
  - But `city` is a class
    - `city` is a **hyponym** of `municipality`, `location...`

The levels are not symmetric..

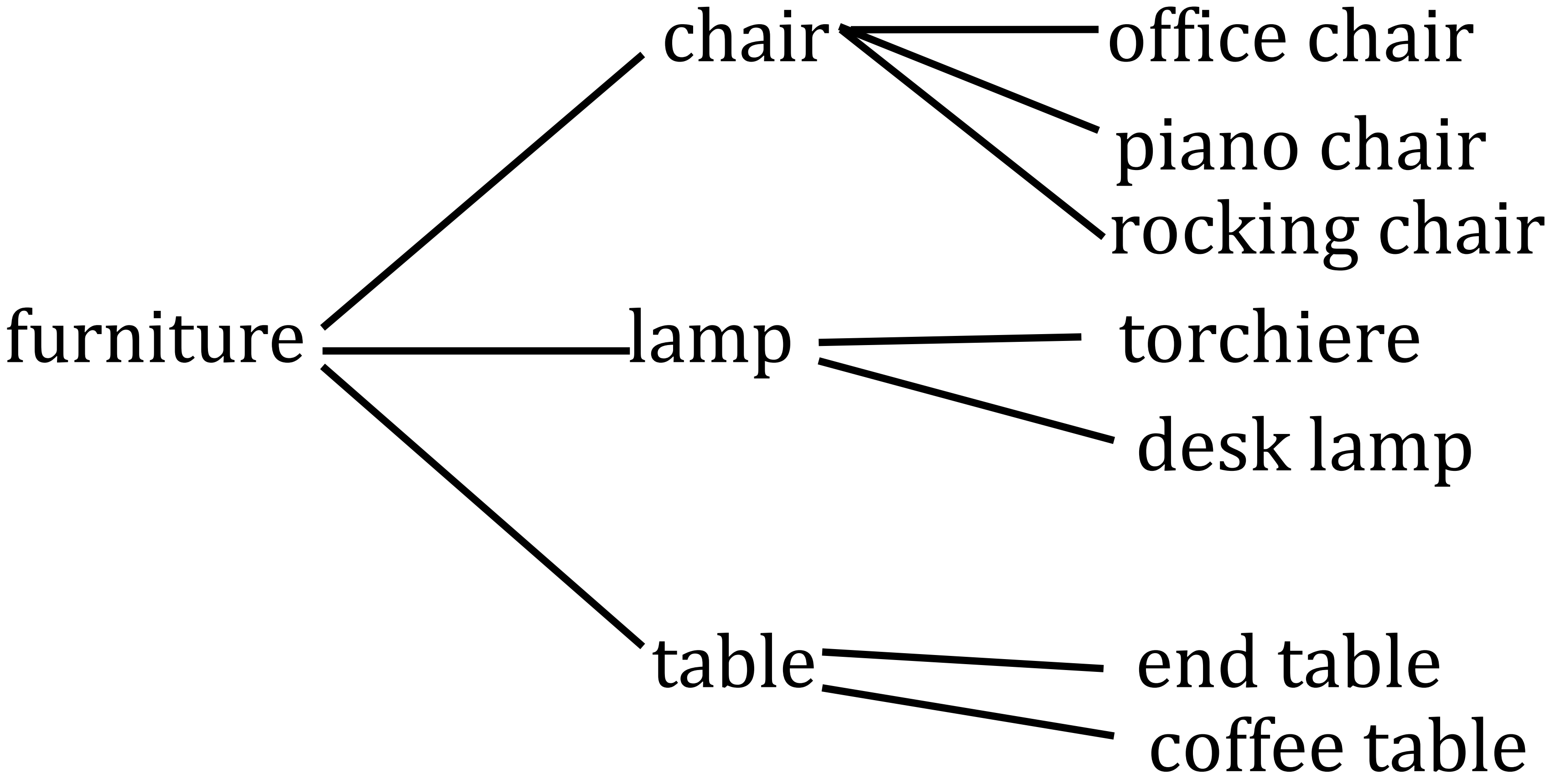
- One level of category is distinguished from the others
- The "basic level"

Name these items





Superordinate **Basic** **Subordinate**



Cluster of Interactional Properties

- Basic level things are “human-sized”
- Consider chairs
  - We know how to interact with a chair (sit)
  - Not so clear for superordinate categories like furniture
    - “Imagine a furniture without thinking of a bed/table/chair/specific basic-level category”

The basic level

- Distinctive actions
- Learned earliest in childhood
- Names are shortest
- Names are most frequent

Connotation (sentiment)

- Words have **affective** meanings
  - positive connotations (*happy*)
  - negative connotations (*sad*)

```
fake vs. replica
innocent vs. naive
```

  - positive evaluation (*great, love*)
  - negative evaluation (*terrible, hate*)

So far

- **Concepts** or word senses
  - Have a complex many-to-many associations with **words** (homonymy, multiple senses)
- Have relations with each other
  - Synonymy
  - Antonymy
  - Similarity
  - Relatedness
  - Superordinate/subordinate, basic level
  - Connotations

WordNet

WordNet 3.0

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary
    - Some other languages available or under development
        - (Arabic, Finnish, German, Portuguese…)

Category	Unique Strings
Nouns	117,798
Verbs	11,529
Adjectives	22,479
Adverbs	4,481

Senses of “bass” in WordNet

Noun

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
- **S: (n) bass, basso (an adult male singer with the lowest voice)**
- S: (n) sea bass, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) freshwater bass, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, bass voice, basso (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Adjective

- S: (adj) **bass**, deep (having or denoting a low vocal or instrumental range) *"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"*

How is sense defined in WordNet?

- The **synset (synonym set)**, the set of near-synonyms, instantiates a sense or concept, with a gloss
- Example: chump as a noun with the gloss:

  "a person who is gullible and easy to take advantage of"
- This sense of "chump" is shared by 9 words:

  `chump`^{1}, `fool`^{2}, `gull`^{1}, `mark`^{9}, `patsy`^{1}, `fall guy`^{1}, `sucker`^{1}, `soft touch`^{1}, `mug`^{2}
- Each of **these** senses have this same gloss
  - (Not **every** sense of a word; e.g., sense 2 of gull is the aquatic bird)

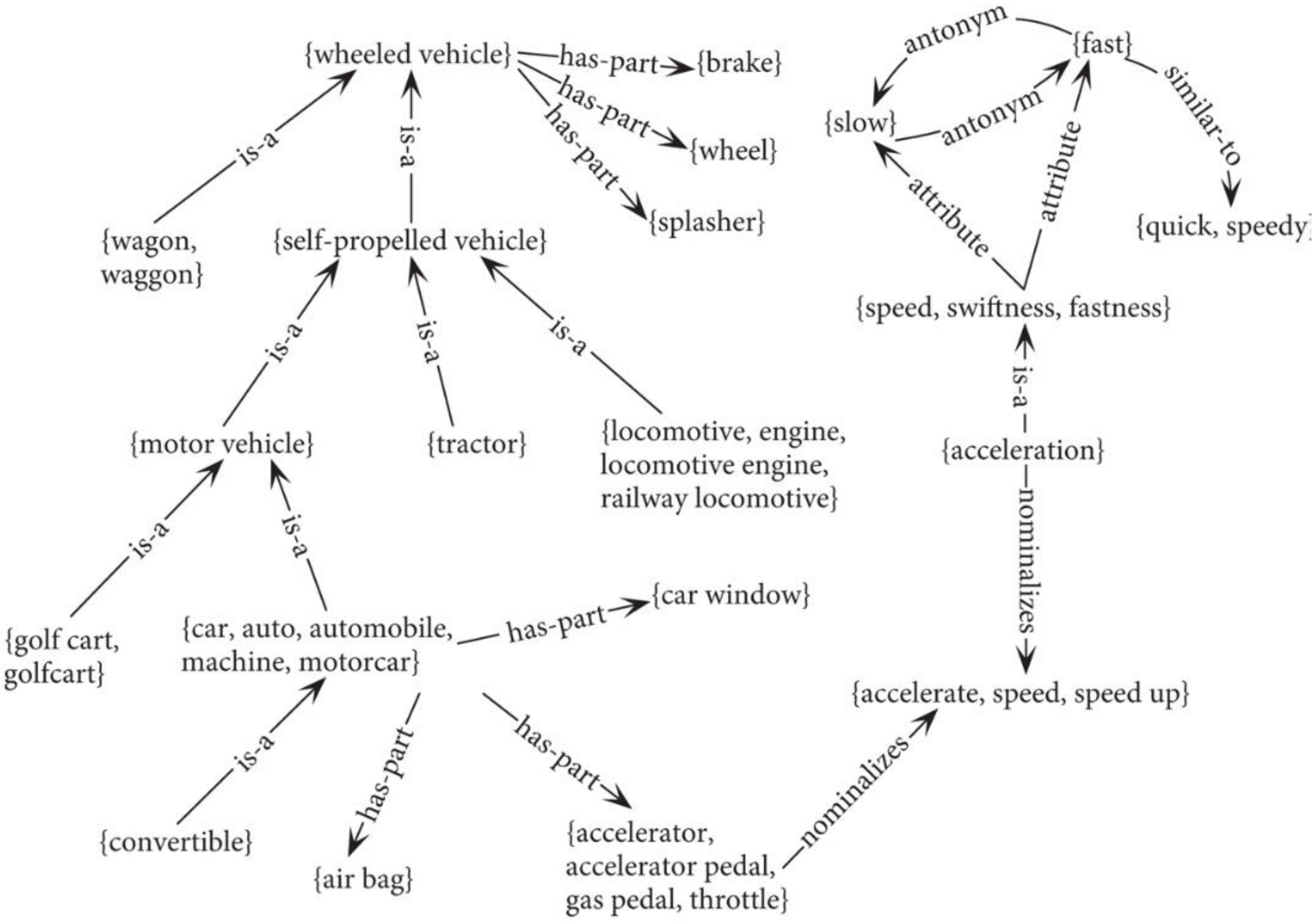
On average noun has 1.23 senses and verb has 2.16 senses

WordNet Hypernym Hierarchy for "bass"

- S: (n) **bass**, basso (an adult male singer with the lowest voice)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
      - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
        - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
          - S: (n) entertainer (a person who tries to please or amuse)
            - S: (n) person, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
              - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                - S: (n) living thing, animate thing (a living (or once living) entity)
                  - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                    - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                      - S: (n) physical entity (an entity that has physical existence)
                        - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

WordNet Noun Relations

Relation	Also called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	*breakfast*^{1} $\rightarrow$ *meal*^{1}
Hyponym	Subordinate	From concepts to subtypes	*meal*^{1} $\rightarrow$ *lunch*^{1}
Member Meronym	Has-Member	From groups to their members	*faculty*^{2} $\rightarrow$ *professor*^{1}
Has-Instance		From concepts to instances of the concept	*composer*^{1} $\rightarrow$ *Bach*^{1}
Instance		From instances to their concepts	*Austen*^{1} $\rightarrow$ *author*^{1}
Member Holonym	Member-Of	From members to their groups	*copilot*^{1} $\rightarrow$ *crew*^{1}
Part Meronym	Has-Part	From wholes to parts	*table*^{2} $\rightarrow$ *leg*^{3}
Part Holonym	Part-Of	From parts to wholes	*course*^{7} $\rightarrow$ *meal*^{1}
Antonym		Opposites	*leader*^{1} $\rightarrow$ *follower*^{1}

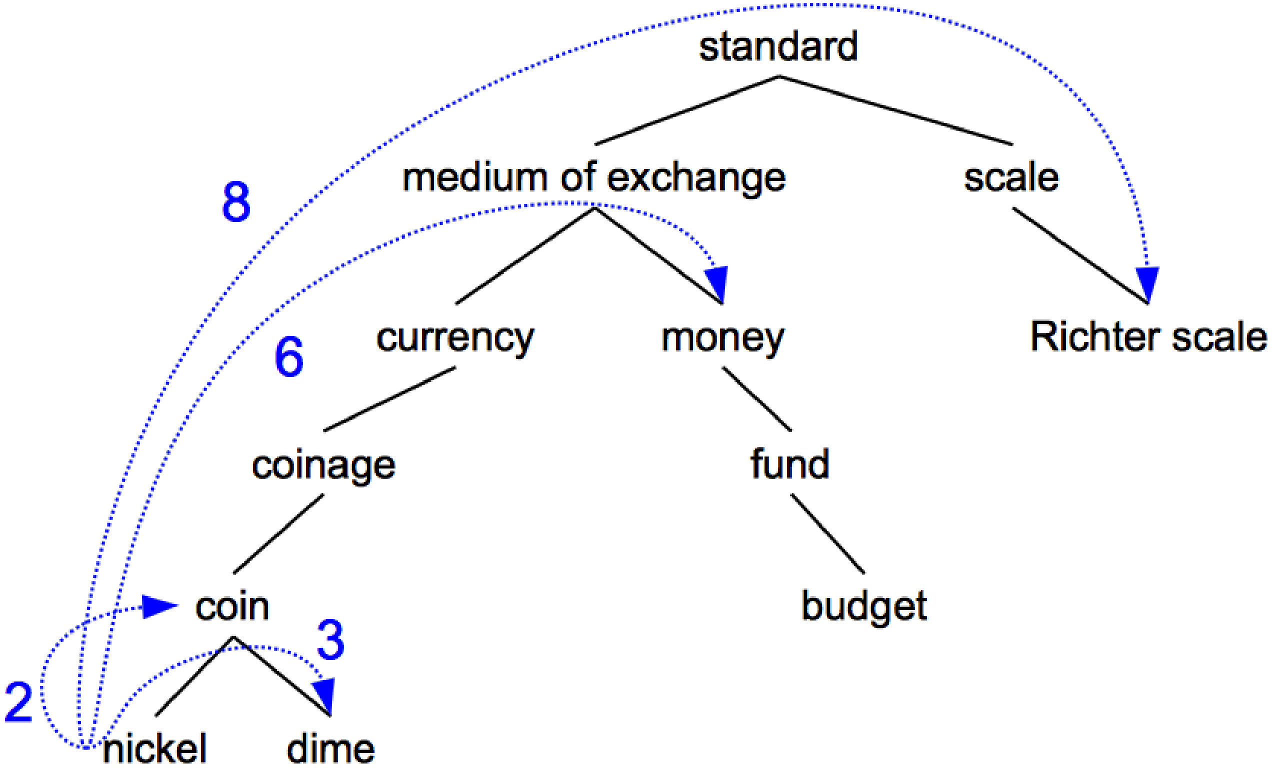


Navigli, R. (2016). Ch 20, Ontologies. In Mitkov, R. (Ed.),The Oxford handbook of computational linguistics. Oxford University Press.

(FYI) How to compute similarity of two words?

- Several methods possible (ones listed later tend to be better):
  1. cosine similarity of definitions of two words (e.g., extended Lesk algorithm)
  2. distance of words in the ontology tree of senses (hypernymy tree in WordNet)
  3. cosine similarity measure between two vectors containing tfidf values in term-document matrix
  4. cosine similarity measure between two vectors containing raw co-occurrence counts of words (each word represented as a vector over vocabulary V)
  5. cosine similarity measure between two vectors of PMI associations computed based on word co-occurrence counts
  6. cosine similarity measure between two dense vectors such as ones obtained using deep learning approaches (e.g., word2vec vectors)
  7. …

Path based similarity



- Two concepts (senses/synsets) are similar if they are near each other in the thesaurus hierarchy
  - =have a short path between them
  - concepts have path 1 to themselves

Path based similarity

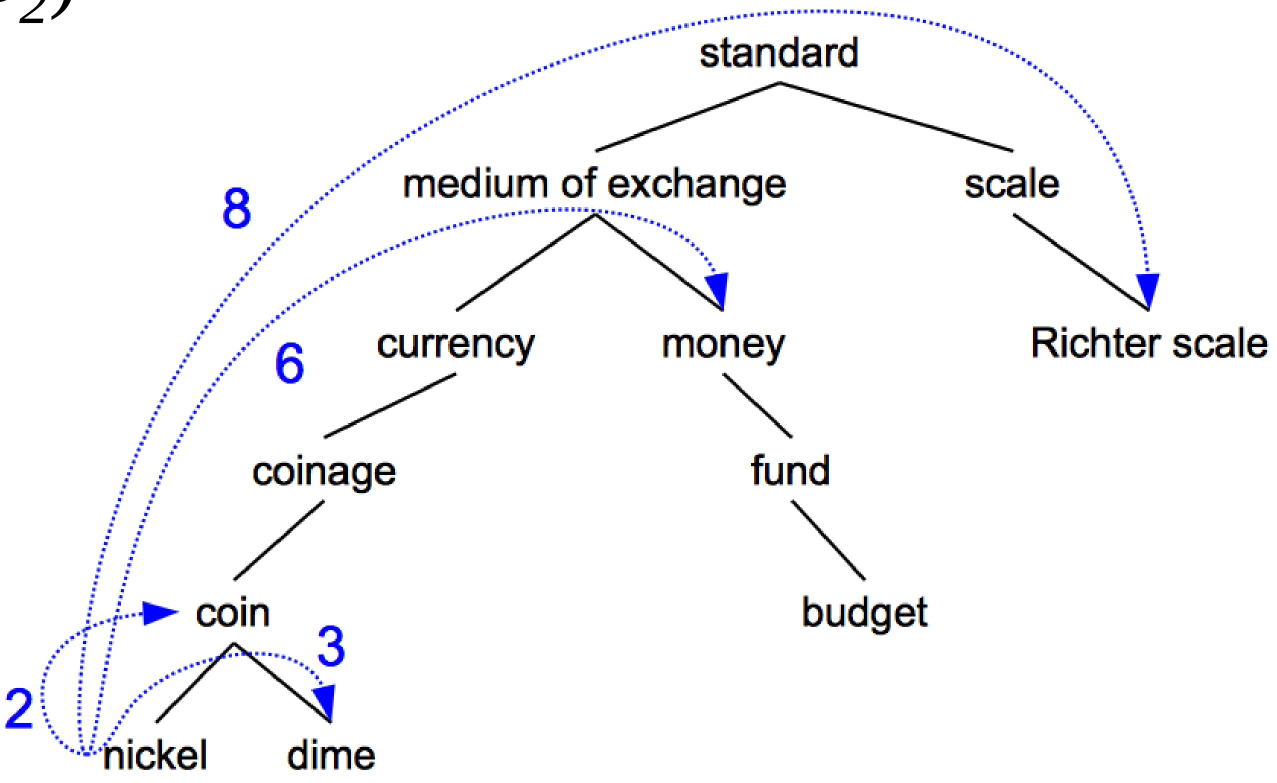
pathlen(c_1,c_2) = 1 + number of edges in the shortest path in the hypernym graph between sense nodes c_1 and c_2

- ranges from 0 to 1 (identity)

simpath(c_1,c_2) = $$\frac{1}{\text{pathlen}(c_1, c_2)}$$

Example: path-based similarity

$\text{simpath}(c_1,c_2) = 1/\text{pathlen}(c_1,c_2)$



simpath(*nickel,coin*) = 1/2 = .5

simpath(*fund,budget*) = 1/2 = .5

simpath(*nickel,currency*) = 1/4 = .25

simpath(*nickel,money*) = 1/6 = .17

simpath(*coinage,Richter scale*) = 1/6 = .17

Problem with basic path based similarity

- Assumes each link represents a uniform distance
  - But *nickel* to *money* seems to us to be closer than *nickel* to *standard*
  - Nodes high in the hierarchy are very abstract
- We instead want a metric that
  - Represents the cost of each edge independently
  - Words connected only through abstract nodes are less similar

The (extended) Lesk Algorithm

- A thesaurus-based measure that looks at **glosses**
- Two concepts are similar if their glosses contain similar words
  - ***Drawing paper***: paper that is specially prepared for use in drafting
  - ***Decal***: the art of transferring designs from specially prepared paper to a wood or glass or metal surface
- For each *n*-word phrase that's in both glosses
  - Add a score of n^2
  - Paper and specially prepared for $1 + 2^2 = 5$
  - Compute the overlap also for other relations
    - glosses of hypernyms and hyponyms

$$\mathrm{sim}_{eLesk}(c_1, c_2) = \sum_{r,q \in RELS} \mathrm{overlap}(\mathrm{gloss}(r(c_1)), \mathrm{gloss}(q(c_2)))$$

WordNet 3.0

- Where it is:
  - http://wordnetweb.princeton.edu/perl/webwn
- Libraries
  - R: wordnet package
  - Python: WordNet from NLTK
    - http://www.nltk.org/Home
  - Java:
    - JWNL, extJWNL on sourceforge

MeSH: Medical Subject Headings thesaurus from the National Library of Medicine

- **MeSH (Medical Subject Headings)**
  - 177,000 entry terms that correspond to 26,142 biomedical "headings"

- **Hemoglobins**

  Entry Terms: Eryhem, Ferrous Hemoglobin, Hemoglobin

  Definition: The oxygen-carrying proteins of ERYTHROCYTES. They are found in all vertebrates and some invertebrates. The number of globin subunits in the hemoglobin quaternary structure differs between species. Structures range from monomeric to a variety of multimeric arrangements

The MeSH Hierarchy

1. + Anatomy [A]
2. + Organisms [B]
3. + Diseases [C]
4. + Chemicals and Drugs [D]
5. + Analytical, Diagnostic and T
6. + Psychiatry and Psychology |
7. + Phenomena and Processes [(
8. + Disciplines and Occupation:
9. + Anthropology, Education, S
10. + Technology, Industry, Agri(
11. + Humanities [K]
12. + Information Science [L]
13. + Named Groups [M]
14. + Health Care [N]
15. + Publication Characteristics
16. + Geographicals [Z]

1. + Anatomy [A]
2. + Organisms [B]
3. + Diseases [C]
4. − Chemicals and Drugs [D]
    - Inorganic Chemicals [D01] +
    - Organic Chemicals [D02] +
    - Heterocyclic Compounds [D03] +
    - Polycyclic Compounds [D04] +
    - Macromolecular Substances [D05] +
    - Hormones, Hormone Substitutes, and Hormone
    - Enzymes and Coenzymes [D08] +
    - Carbohydrates [D09] +
    - Lipids [D10] +
    - Amino Acids, Peptides, and Proteins [D12] +
    - Nucleic Acids, Nucleotides, and Nucleosides [D1:
    - Complex Mixtures [D20] +
    - Biological Factors [D23] +
    - Biomedical and Dental Materials [D25] +
    - Pharmaceutical Preparations [D26] +
    - Chemical Actions and Uses [D27] +
5. + Analytical, Diagnostic and Therapeutic Techniques
6. + Psychiatry and Psychology [F]
7. + Phenomena and Processes [G]

Amino Acids, Peptides, and Proteins [D12]
- Proteins [D12.776]
    - Blood Proteins [D12.776.124]
        - Acute-Phase Proteins [D12.776.124.050] +
        - Anion Exchange Protein 1, Erythrocyte [D12.776.124.078
        - Ankyrins [D12.776.124.080]
        - beta 2-Glycoprotein I [D12.776.124.117]
        - Blood Coagulation Factors [D12.776.124.125] +
        - Cholesterol Ester Transfer Proteins [D12.776.124.197]
        - Fibrin [D12.776.124.270] +
        - Glycophorin [D12.776.124.300]
        - Hemocyanin [D12.776.124.337]
        - ➤ Hemoglobins [D12.776.124.400]
            - Carboxyhemoglobin [D12.776.124.400.141]
            - Erythrocruorins [D12.776.124.400.220]

Uses of the MeSH Ontology

- Provide synonyms (“entry terms”)
  - E.g., glucose and dextrose
- Provide hypernyms (from the hierarchy)
  - E.g., glucose ISA monosaccharide
- Indexing in MEDLINE/PubMED database
  - NLM’s bibliographic database:
    - 20 million journal articles
    - Each article hand-assigned 10-20 MeSH terms

Vector Semantics

How about a radically different approach?

Ludwig Wittgenstein

"The meaning of a word is its use in the language"

Let's define words by their usage

- 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.

"You shall know a word by a company it keeps" (Firth 1957: 11)

What does recent English borrowing *ong choy* mean?

- Suppose you see these sentences:
  - Ong choy is delicious **sautéed with garlic**.
  - Ong choy is superb **over rice**
  - Ong choy **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 choy is a leafy green like spinach, chard, or collard greens

Ong choy: *Ipomoea aquatica "Water Spinach"*

空心菜
kangkong
rau muống
...

Yamaguchi, Wikimedia Commons, public domain

A new model of meaning focusing on distributional similarity

- Each word = a vector
- Similar words are "nearby in space"

We define 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 then generalize to **similar but unseen** words

We'll discuss two kinds of embeddings

- tf-idf
  - Information Retrieval workhorse
  - A common baseline model
  - **Sparse** vectors
  - Words are represented by (a simple function of) the **counts** of nearby words
- Word2vec
  - **Dense** vectors
  - Representation is created by training a classifier to **predict** whether a word is likely to appear nearby
  - There are extensions such as **contextual embeddings**

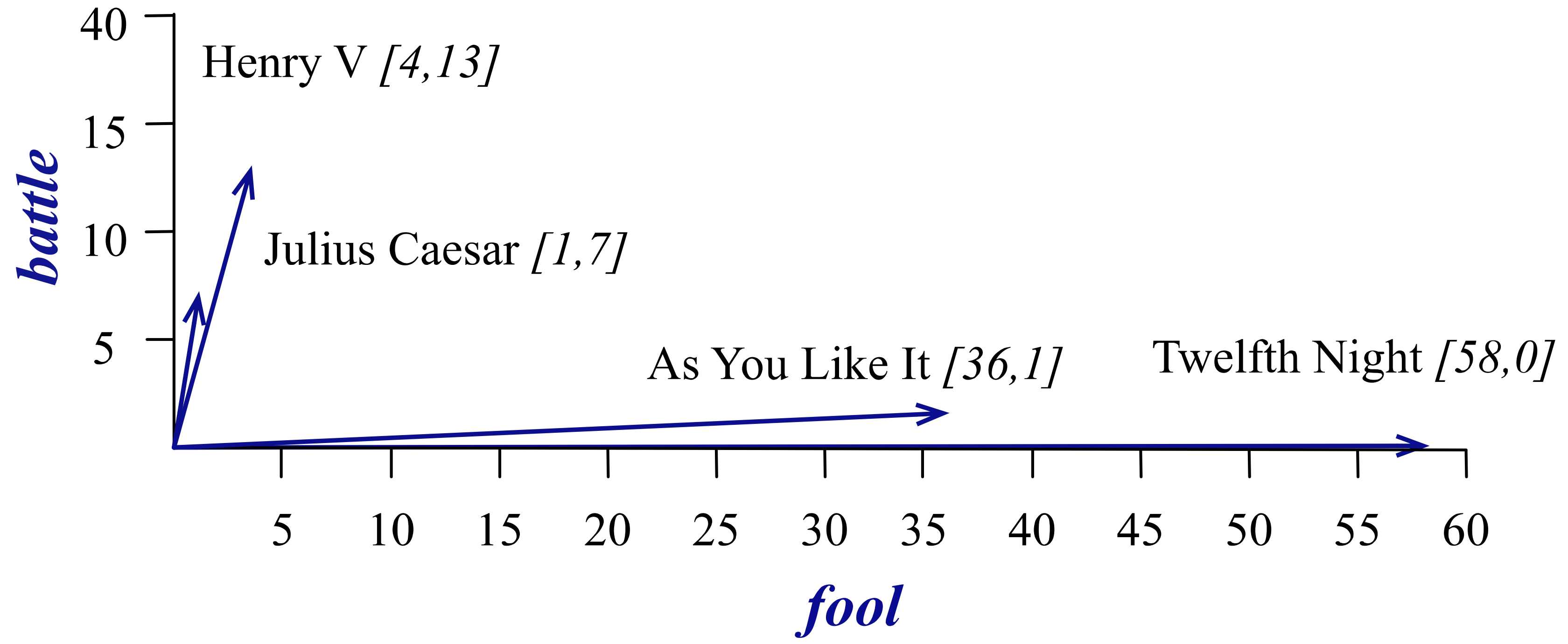
Words and Vectors

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

Visualizing document vectors



Vectors are the basis of information retrieval

	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
- Different than the history
- Comedies have more *fools* and *wit* and fewer *battles*.

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"

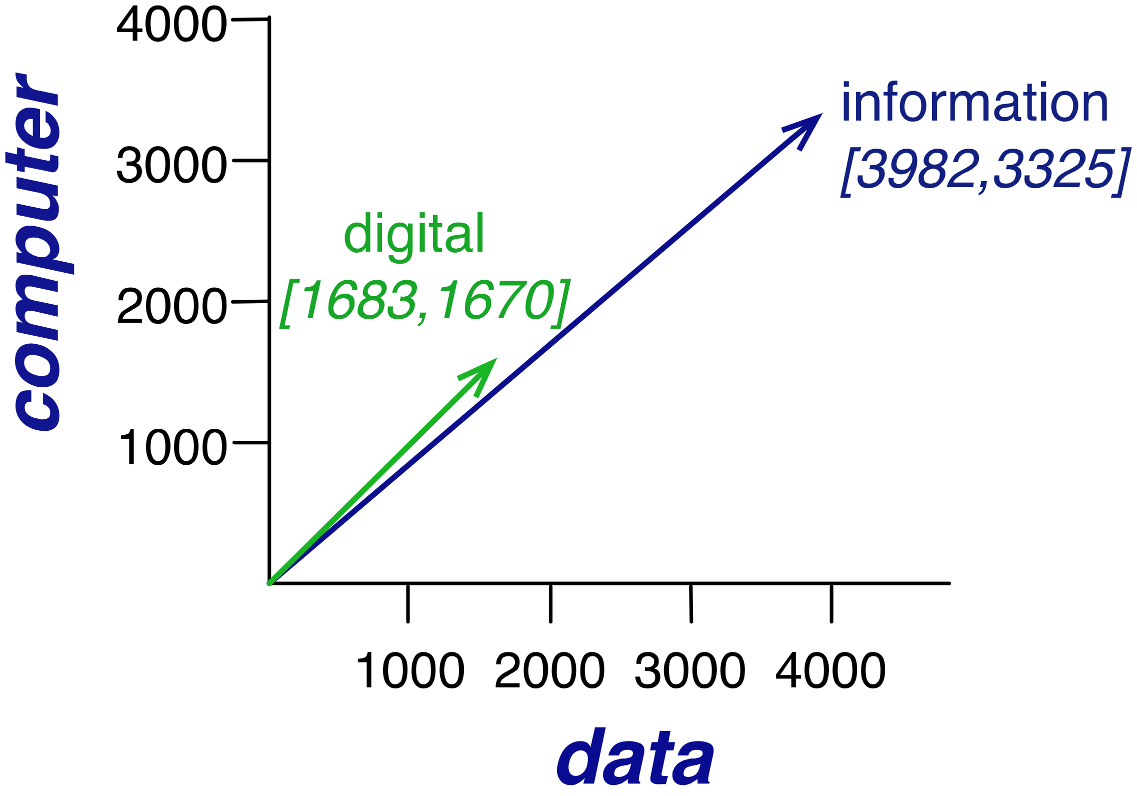
fool is "the kind of word that occurs in comedies, especially Twelfth Night"

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

Cosine for 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 be a 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 dimensions)
- Vector length:

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

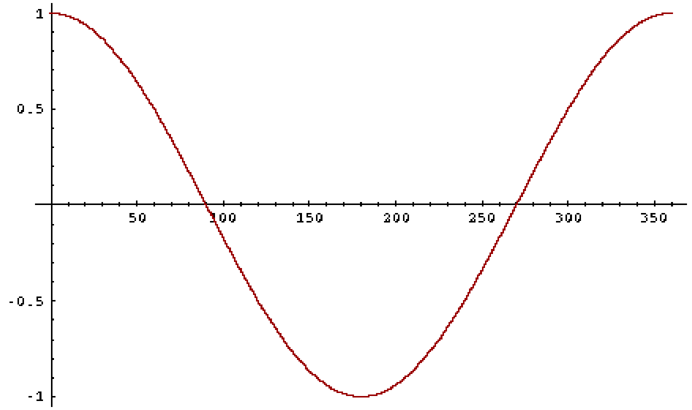
- Frequent words (of, the, you) have long vectors (since they occur many times with other words)
- So dot product overly favors frequent words

Alternative: cosine for computing word similarity

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

Cosine as a similarity metric

- -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 to 1

Cosine examples

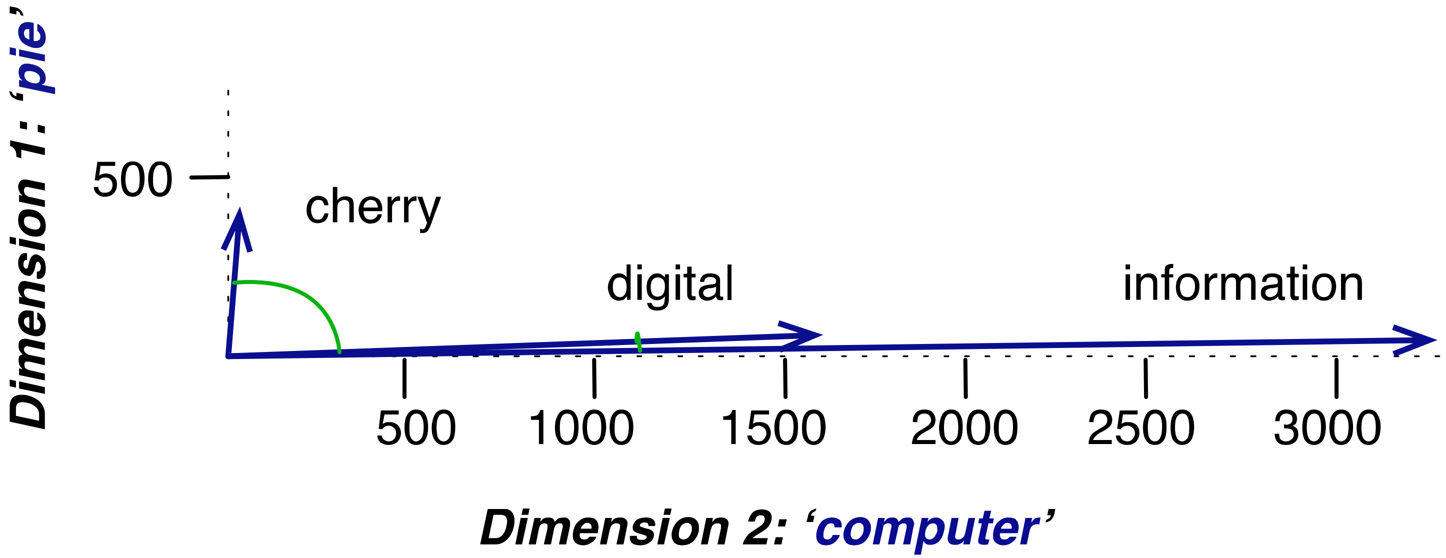
$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|}\cdot\frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

	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

But raw frequency is a bad representation

- 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
- We need a function that resolves this frequency paradox!

Term frequency (tf)

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

Instead of using the raw count, we may squash a bit:

- $\text{tf}_{t,d} = \log_{10}(\text{count}(t,d)+1)$

Document frequency (df)

- df_t *is* the number of documents that t occurs in

(note: this is not a 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

- Important: documents can be **anything**; we can call each paragraph a document

Inverse document frequency (idf)

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

N is the total number of documents in the collection

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:

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

	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

Dense vectors

Sparse versus dense vectors

- tf-idf 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)

Word Representations: High vs. Low Dimension Vectors (Dense vs. Sparse)

Sparse Vector Space representation, FCM (term co-occurrence, etc.)

Size of Vocabulary ≈ 10,000 - 400,000

$$\overrightarrow{\text{ipod}} = \begin{array}{c} \begin{array}{ccccccccccccc} \text{a} & \text{an} & \ldots & \text{music} & \ldots & \text{road} & \ldots & \text{usa} & \ldots & \text{zebra} & \ldots & \text{zoo} \end{array} \\ \begin{bmatrix} 50 & 150 & \ldots & 200 & \ldots & 5 & \ldots & 100 & \ldots & 0 & \ldots & 0 \end{bmatrix} \end{array}$$

$$\overrightarrow{\text{car}} = \begin{array}{c} \begin{array}{cccccccccccc} \text{a} & \text{an} & \ldots & \text{music} & \ldots & \text{road} & \ldots & \text{usa} & \ldots & \text{zebra} & \ldots & \text{zoo} \end{array} \\ \begin{bmatrix} 60 & 180 & \ldots & 50 & \ldots & 200 & \ldots & 350 & \ldots & 0 & \ldots & 0 \end{bmatrix} \end{array}$$

Dense Vector Space representation, e.g., LSA, Word2vec, GLoVe

Size of Dimensions 50 - 1000

$$\overrightarrow{\text{ipod}} = \begin{array}{c} \begin{array}{cccccc} \chi_0 & \chi_1 & \cdots & \chi_i & \cdots & \chi_{200} \end{array} \\ \begin{bmatrix} 0.1 & -0.2 & \ldots & 0.3 & \ldots & 0.8 \end{bmatrix} \end{array}$$

$$\overrightarrow{\text{car}} = \begin{array}{c} \begin{array}{cccccc} \chi_0 & \chi_1 & \cdots & \chi_i & \cdots & \chi_{200} \end{array} \\ \begin{bmatrix} 0.4 & 0.1 & \ldots & -0.8 & \ldots & 1.1 \end{bmatrix} \end{array}$$

- Captures semantic meaning better
- Semantically similar words are located close in vector spaces

Sparse versus dense vectors

- Why dense vectors?
  - Short vectors may be easier to use as **features** in machine learning (fewer weights to tune)
  - Dense vectors may **generalize** better than explicit counts
  - They may do better at capturing synonymy:
    - *car* and *automobile* are synonyms; but are distinct dimensions
      - a word with *car* as a neighbor and a word with *automobile* as a neighbor should be similar, but aren't..
  - **In practice, they work better**

Common methods for getting short dense vectors

- "Neural Language Model"-inspired models
  - Word2vec (skipgram, CBOW), Glove
- Singular Value Decomposition (SVD)
  - A special case of this is called LSA – Latent Semantic Analysis
- Alternative to these "static embeddings":
  - Contextual Embeddings (ELMo, BERT)
  - Compute distinct embeddings for a word in its context
  - Separate embeddings for each token of a word

Word2vec: The classifier

Trained Embeddings that can be downloaded

- Word2vec (Mikolov et al.)

https://code.google.com/archive/p/word2vec/

- Glove (Pennington, Socher, Manning)

http://nlp.stanford.edu/projects/glove/

Word2vec

- Popular embedding method
- Very fast to train
- Code available on the web
- Idea: **predict** rather than **count**

Word2vec

- Instead of **counting** how often each word *w* occurs near "*apricot*"
  - Train a classifier on a binary **prediction** task:
    - Is *w* likely to show up near "*apricot*"?
- We don't actually care about this task
    - But we'll take the learned classifier weights as the word embeddings
- Big idea: **self-supervision**:
    - A word c that occurs near apricot in the corpus acts as the gold "correct answer" for supervised learning
    - No need for human labels

    Bengio et al. (2003); Collobert et al. (2011)

Word2Vec: Skip-Gram Task

Word2vec provides a variety of options. We'll cover:

skip-gram with negative sampling (SGNS)

Approach: predict if candidate word *c* is a "neighbor"

1. Treat the target word *t* and a neighboring context word *c* as **positive examples**
2. Randomly sample other words in the lexicon to get negative examples
3. Use logistic regression to train a classifier to distinguish those two cases
4. Use the learned weights as the embeddings

Skip-Gram Training Data

- Assume a +/- 2 word window, given training sentence:

...lemon, a [tablespoon of apricot jam, a] pinch...

c1 c2 [target] c3 c4

Skip-Gram Training Data

Assume a +/- 2 word window, given training sentence:

...lemon, a [tablespoon of apricot jam, a] pinch...

c1 c2 [target] c3 c4

Goal: train a classifier that is given a candidate (**w**ord, **c**ontext) pair

(apricot, tablespoon)

(apricot, aardvark)

...

And assigns each pair a probability: $P(+|w, c)$

Similarity is computed from dot product

Two vectors are similar if they have a high dot product
  - Cosine is just a normalized dot product
- So:
  - Similarity(w,c) $\propto$ w $\cdot$ c
- We'll need to normalize to get a probability
  - (cosine isn't a probability either)

Turning dot products into probabilities

- Sim(w, c) ≈ w $\cdot$ c
- To turn this into a probability
- We'll use the sigmoid from logistic regression:

$$P(+|w,c) \;=\; \sigma(c \cdot w) = \frac{1}{1+\exp(-c \cdot w)}$$

$$\begin{aligned} P(-|w,c) &= 1 - P(+|w,c) \\ &= \sigma(-c \cdot w) = \frac{1}{1+\exp(c \cdot w)} \end{aligned}$$

How Skip-Gram Classifier computes $P(+|w, c)$

$$P(+|w,c) \;=\; \sigma(c \cdot w) = \frac{1}{1+\exp(-c \cdot w)}$$

This is for one context word, but we have lots of context words.
We assume independence and just multiply them:

$$P(+|w,c_{1:L}) \;=\; \prod_{i=1}^{L} \sigma(c_i \cdot w)$$

$$\log P(+|w,c_{1:L}) \;=\; \sum_{i=1}^{L} \log \sigma(c_i \cdot w)$$

Skip-gram classifier: summary

A probabilistic classifier that,

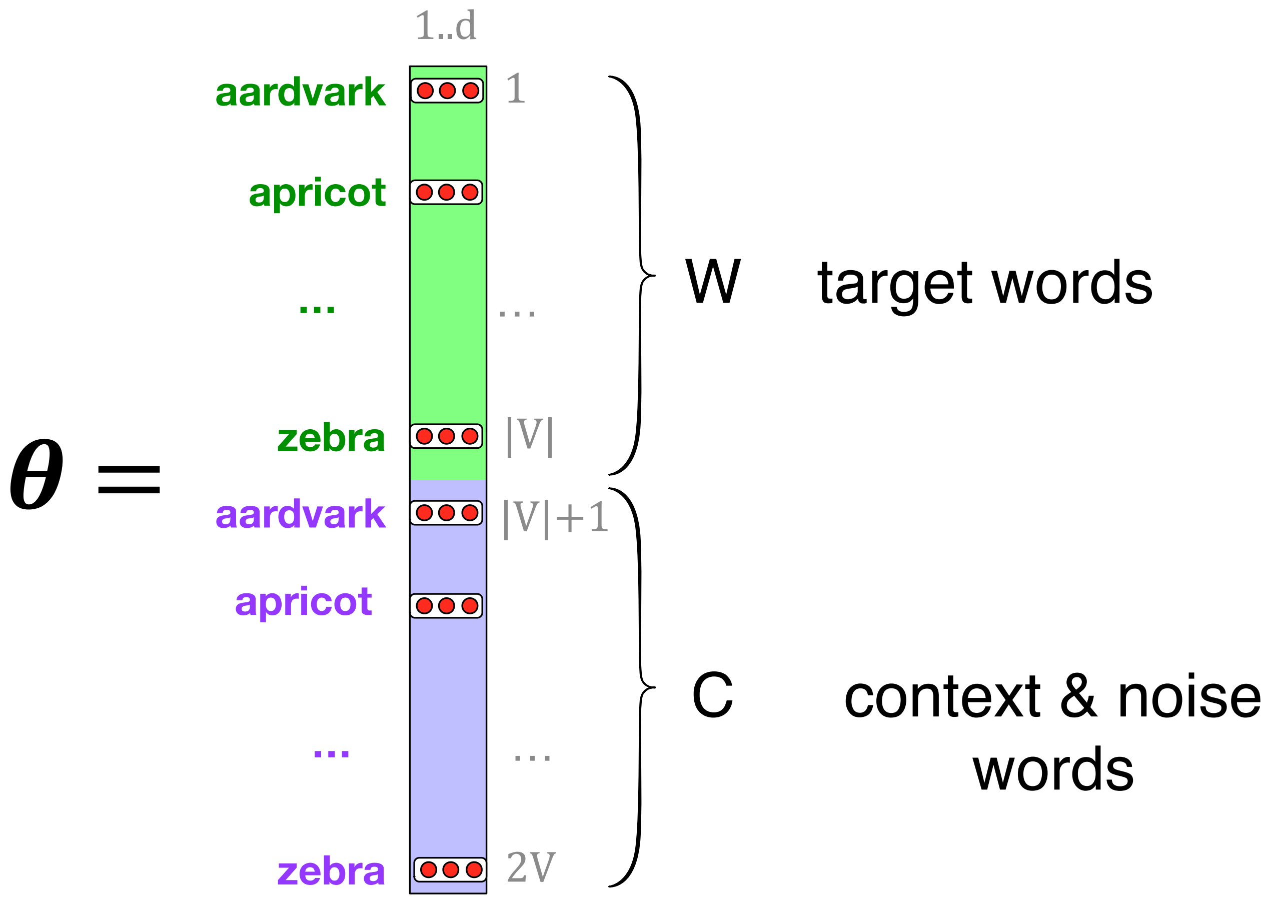
given a test target word w and

its context window of L words $c_{1:L}$,

assigns a probability that w occurs in this window

- To compute this, we just need embeddings for all the words

These embeddings we'll need: a set for w, a set for c



Word2vec: Learning the embeddings

Skip-Gram Training data

...lemon, a [tablespoon of apricot jam, a] pinch...

c1 c2 [target] c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

Skip-Gram Training data

...lemon, a [tablespoon of apricot jam, a] pinch...

c1 c2 [target] c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

For each positive example we'll grab k negative examples, sampling by frequency

Skip-Gram Training data

...lemon, a [tablespoon of apricot jam, a] pinch...
c1 c2 [target] c3 c4

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

Word2vec: how to learn vectors

- Given the set of positive and negative training instances, and an initial set of embedding vectors:
- The goal of learning is to adjust those word vectors such that we:
  - **Maximize** the similarity of the target word, context word pairs (w, c_{pos}) drawn from the positive data
  - **Minimize** the similarity of the (w, c_{neg}) pairs drawn from the negative data

Properties of Word Embeddings

Effect of window size in dense embedding training

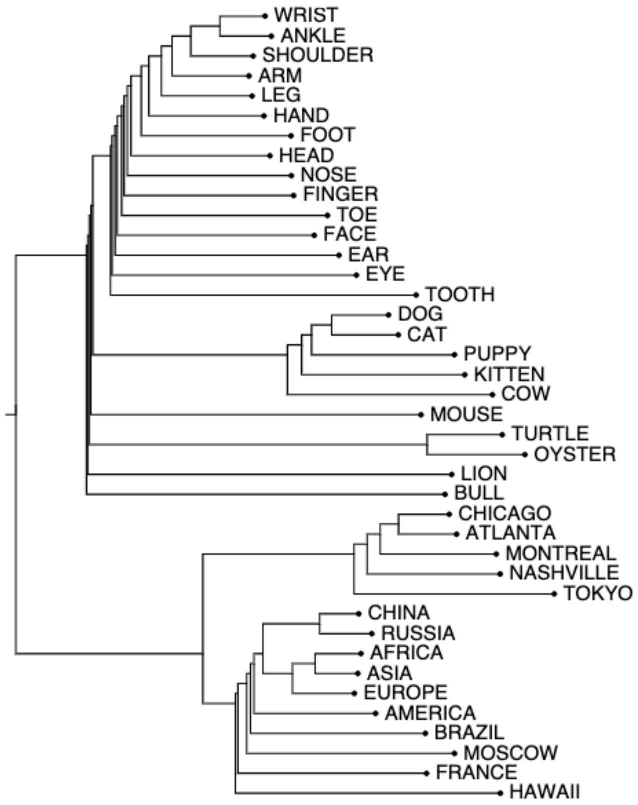
- Shorter context windows tend to lead to more syntactic representations as the information comes from immediately nearby words
  - The most similar words to a target word *w* tend to be semantically similar words with the same parts of speech
- For longer context windows, the most similar words to a target word *w* tend to be words that are topically related but not similar

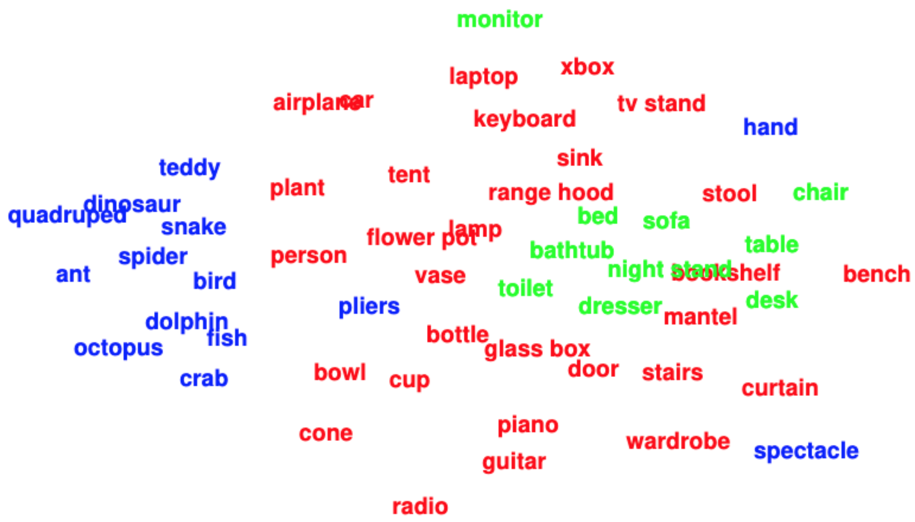
Examples

- **Small windows** (C= +/- 2) : nearest words are similar nouns, words in the same taxonomy, e.g.:
  - *Hogwarts* nearest neighbors are other fictional schools *Sunnydale, Evernight, Blandings*
- **Large windows** (C= +/- 5): nearest words are related words in the same semantic field, e.g.:
  - *Hogwarts* nearest neighbors are terms from the Harry Potter world: *Dumbledore, Half-blood, Malfoy*

Levy, O. and Goldberg, Y. (2014).Dependency-based word embeddings. ACL

Visualizing Word Embeddings



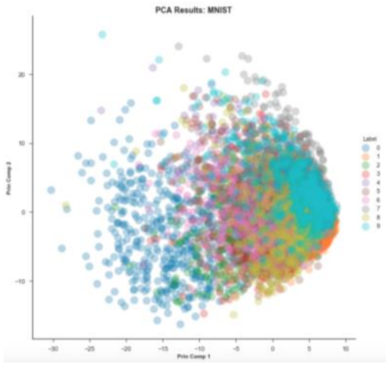


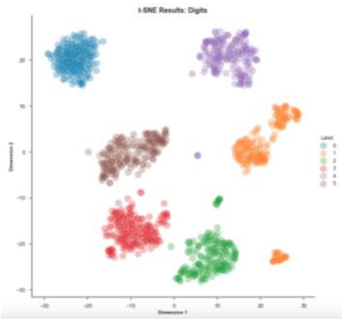
Rohde, D. L. T., Gonnerman, L. M.,and Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. CACM 8, 627–633
Cheraghian, Ali, Shafin Rahman, and Lars Petersson. "Zero-shot learning of 3d point cloud objects." *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019.

t-SNE

- t-distributed stochastic neighbor embedding (t-SNE): statistical method for visualizing high-dimensional data by giving each datapoint a location in a two or three-dimensional map
- A non-linear dimensionality reduction technique well-suited for embedding high-dimensional data
- Goal is to take a set of points in a high-dimensional space and find a faithful representation of those points in a lower-dimensional space, typically the 2D plane
- Implemented in Scikit-learn

MNIST dataset by PCA

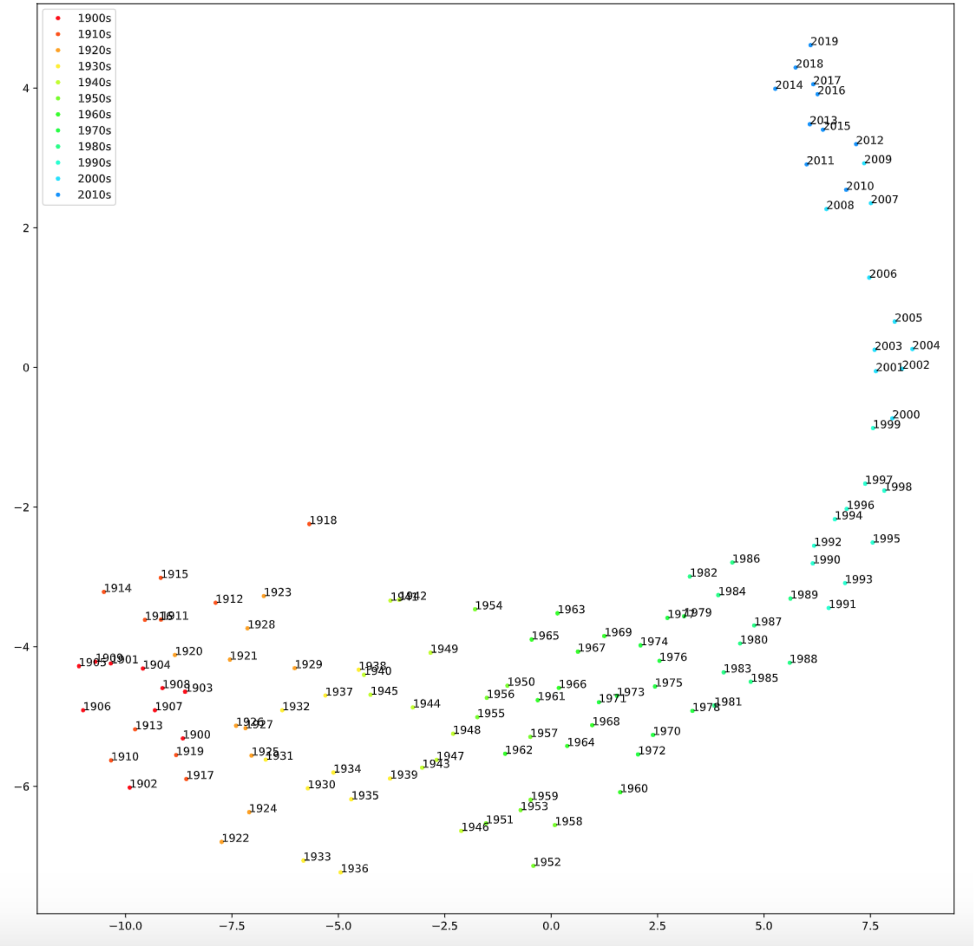




MNIST dataset by T-SNE

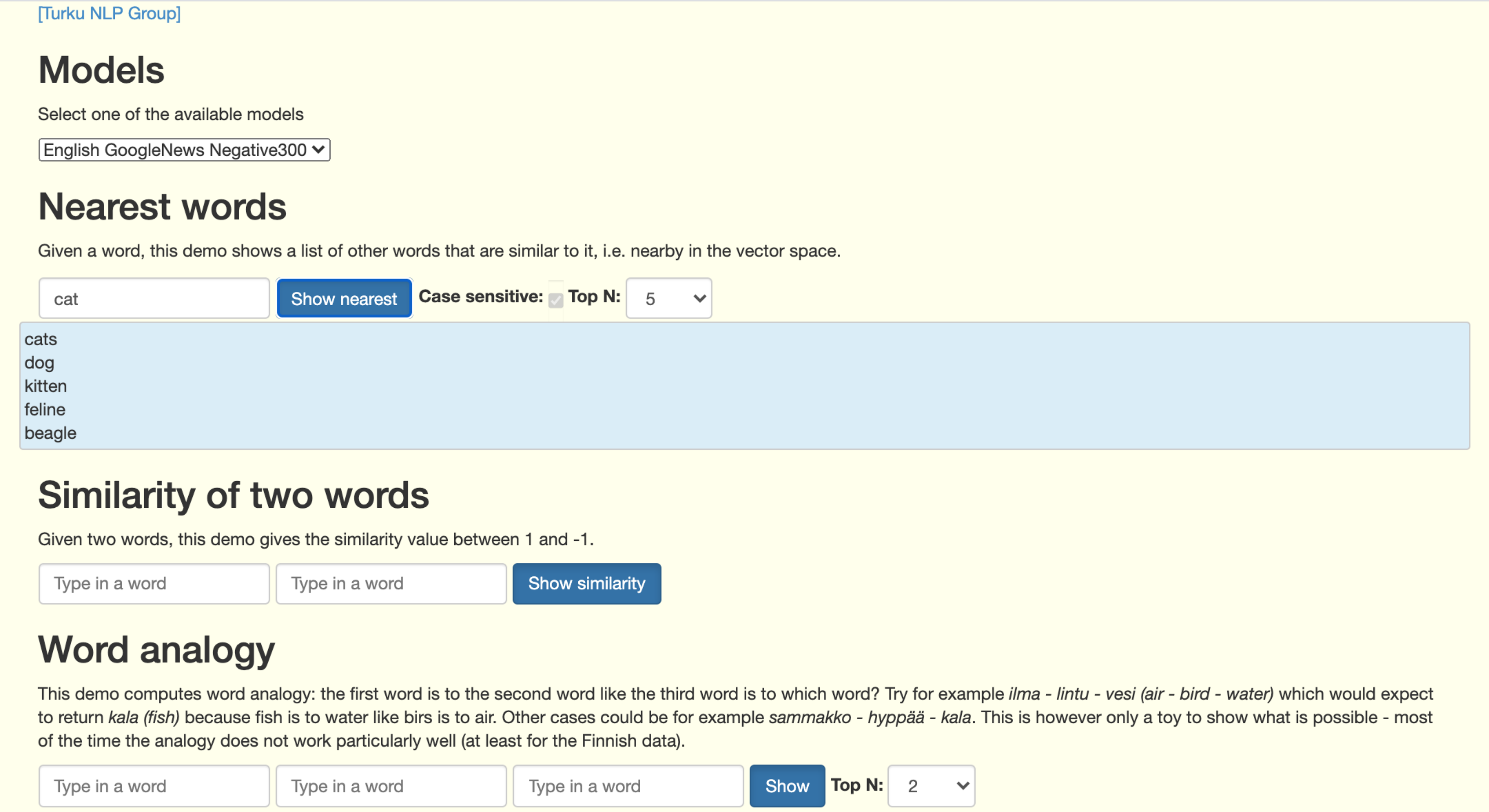
https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
https://medium.com/@violante.andre/an-introduction-to-t-sne-with-python-example-47e6ae7dc58f

Example of visualizing year embeddings using t-SNE



Target: year references extracted from New York Times articles that were published from 1981 to 2013

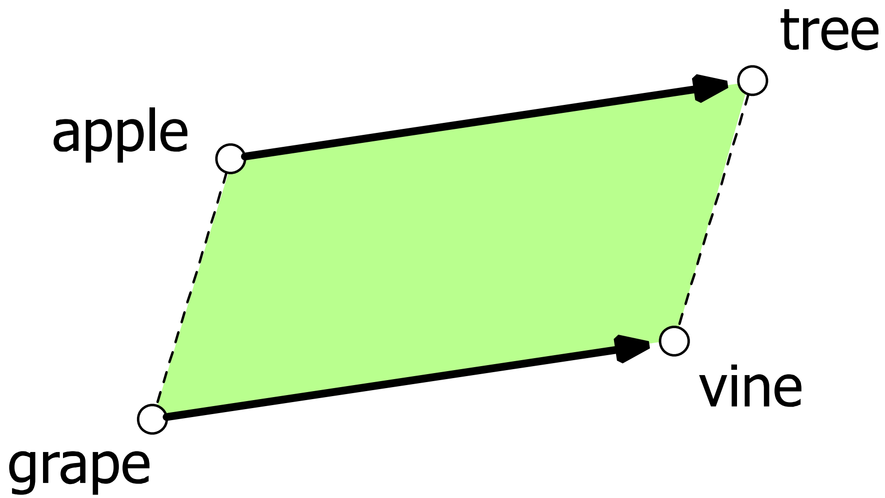
Online Demos



http://bionlp-www.utu.fi/wv_demo/

Analogical relations & capturing relational meanings

- The classic parallelogram model of analogical reasoning (Rumelhart and Abrahamson 1973)
- To solve: *"apple is to tree as grape is to _____"*
- *Add apple – tree to grape to get vine*



Rumelhart, D. E. and Abrahamson, A. A. (1973). A model for analogical reasoning. Cognitive Psychology 5(1),1–28

Analogical relations via parallelogram

- The parallelogram method can solve analogies with both sparse and dense embeddings (Turney and Littman 2005, Mikolov et al. 2013b)

king – man + woman is close to queen

Paris – France + Italy is close to Rome

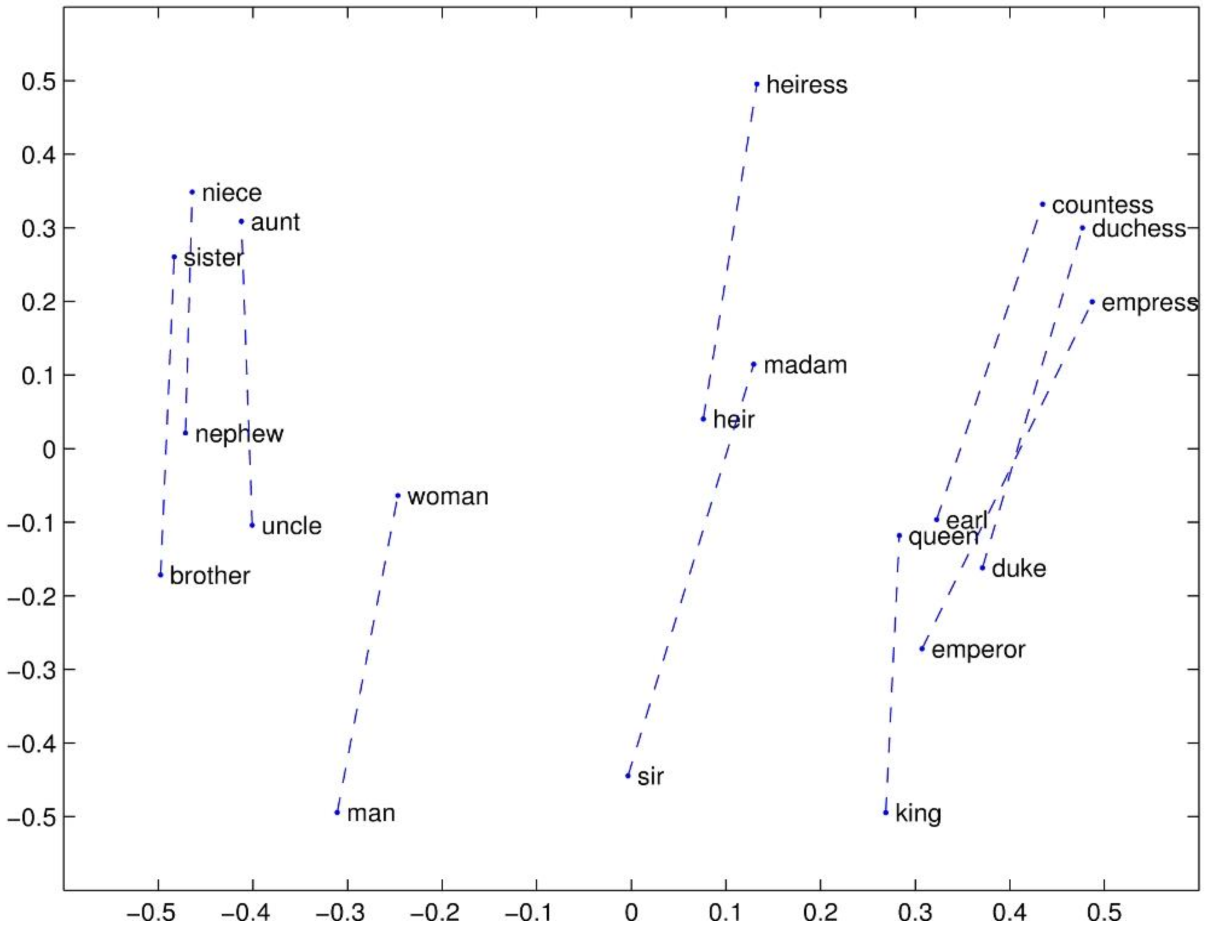
- For a problem a:a*::b:b*, the parallelogram method is:

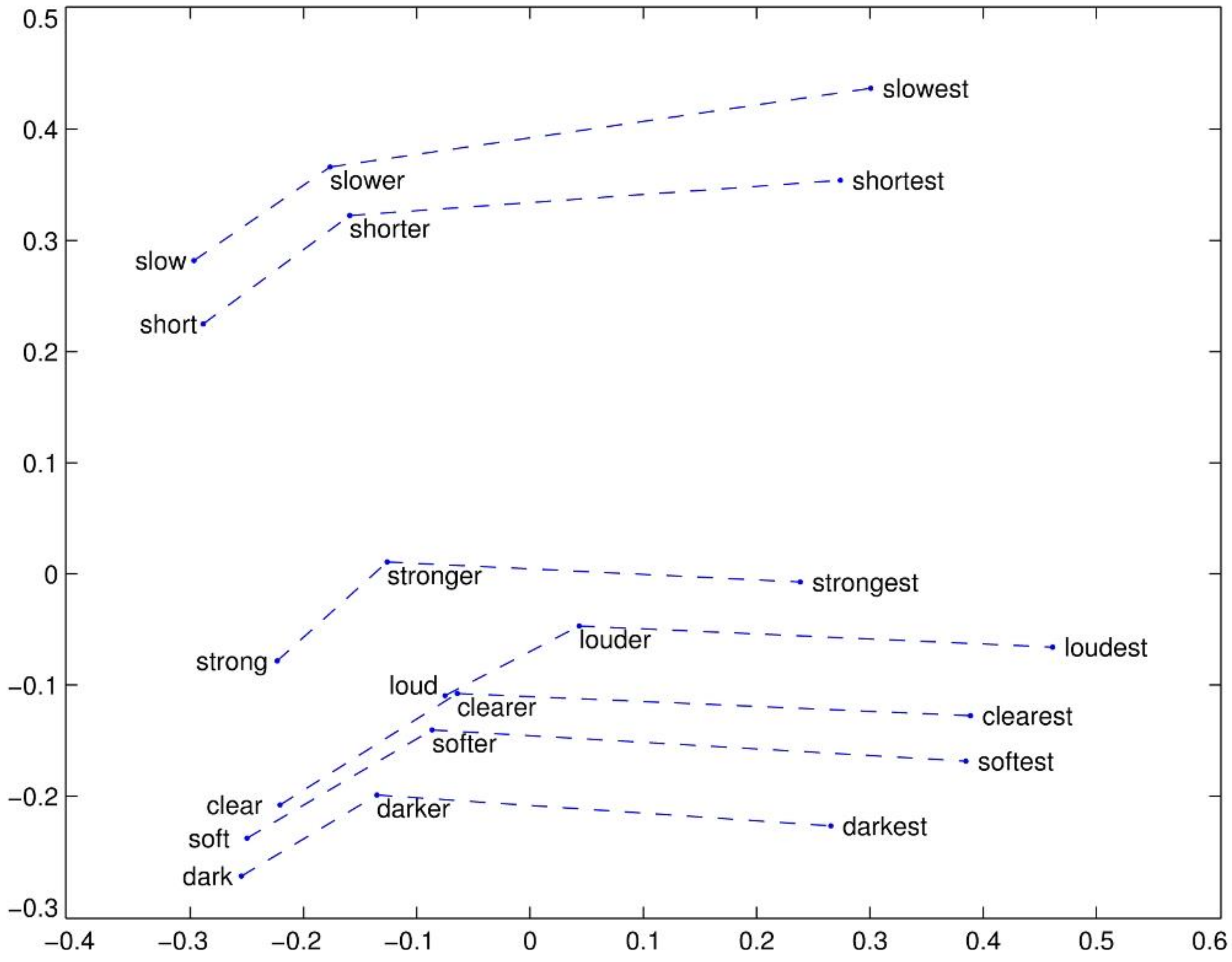
$$\hat{b}^* = \underset{x}{\operatorname{argmax}} \ \text{distance}(x, a^* - a + b)$$

Cosine or Euclidean distance

Turney, P. D. and Littman, M. L.(2005). Corpus-based learning of analogies and semantic relations. Machine Learning 60(1-3), 251–278.
Mikolov, T., Yih, W.-t., and Zweig,G. (2013b). Linguistic regularities in continuous space word representations. NAACL HLT

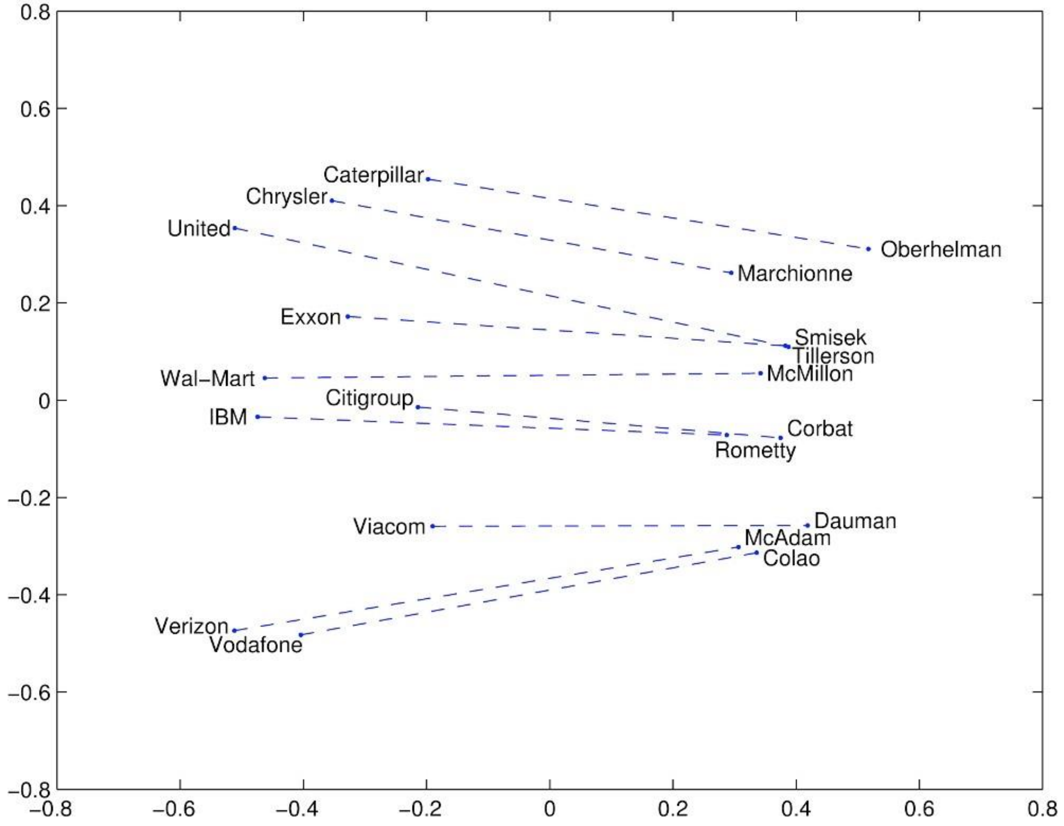
Structure Example in GloVE Embedding space





Comparative and superlative morphology

Glove Visualizations: Company - CEO

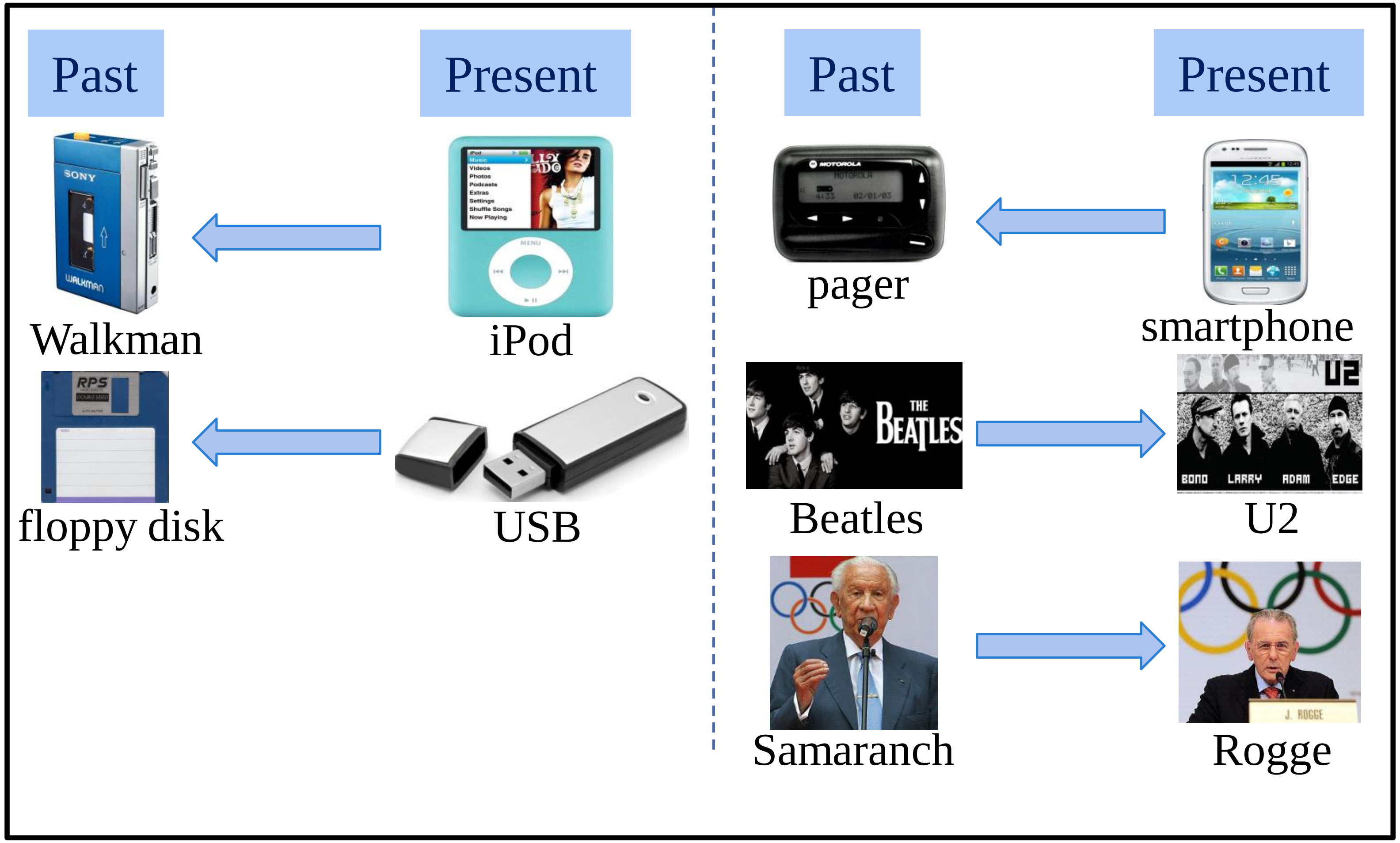


Caveats with the parallelogram method

- It only seems to work for frequent words, small distances and certain relations (e.g., relating countries to capitals, or parts of speech), but not others (Linzen 2016, Ethayarajh et al. 2019a)
- Often morphological variants are returned, e.g., *cherry:red :: potato:x* returns *potatoes* instead of brown so they must be explicitly excluded
- Understanding analogy is an open area of research (Peterson et al. 2020)…

Linzen, T. (2016). Issues in evaluating semantic spaces using word analogies.1st Workshop on Evaluating Vector-Space Representations for NLP
Ethayarajh, K., Duvenaud, D., and Hirst, G. (2019a).Towards under-standing linear word analogies. ACL
Peterson, J. C., Chen, D., and Griffiths,T. L. (2020).Parallelograms revisited: Exploring the limitations of vector space models for simple analogies. Cognition 205

Finding Temporal Analogy



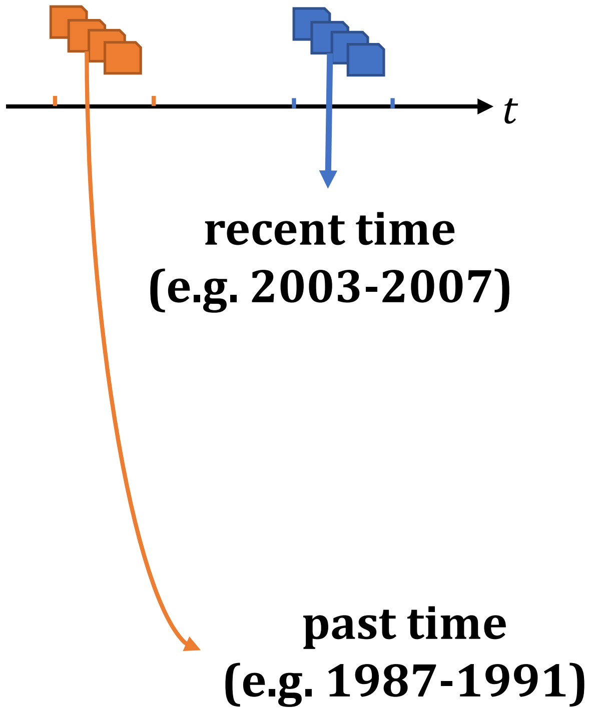
Finding Temporal Analogy: Panta Rei

- **Everything changes**: thus contexts surrounding *temporal analogs* are different

Walkman (1980s)	**iPod (2010s)**
cassette	apple
audio	mp3
video	roqit
tape	player
music	music
sony	geeks
digital	jukebox
stereo	portable
earphone	macintosh
recorder	dlink

** Contexts in the New York Times corpus*

Finding Temporal Analogy: Making NN-based Term Embedding



<u>Distributed Vector Representations</u>
(NN) [Mikolov 2013]

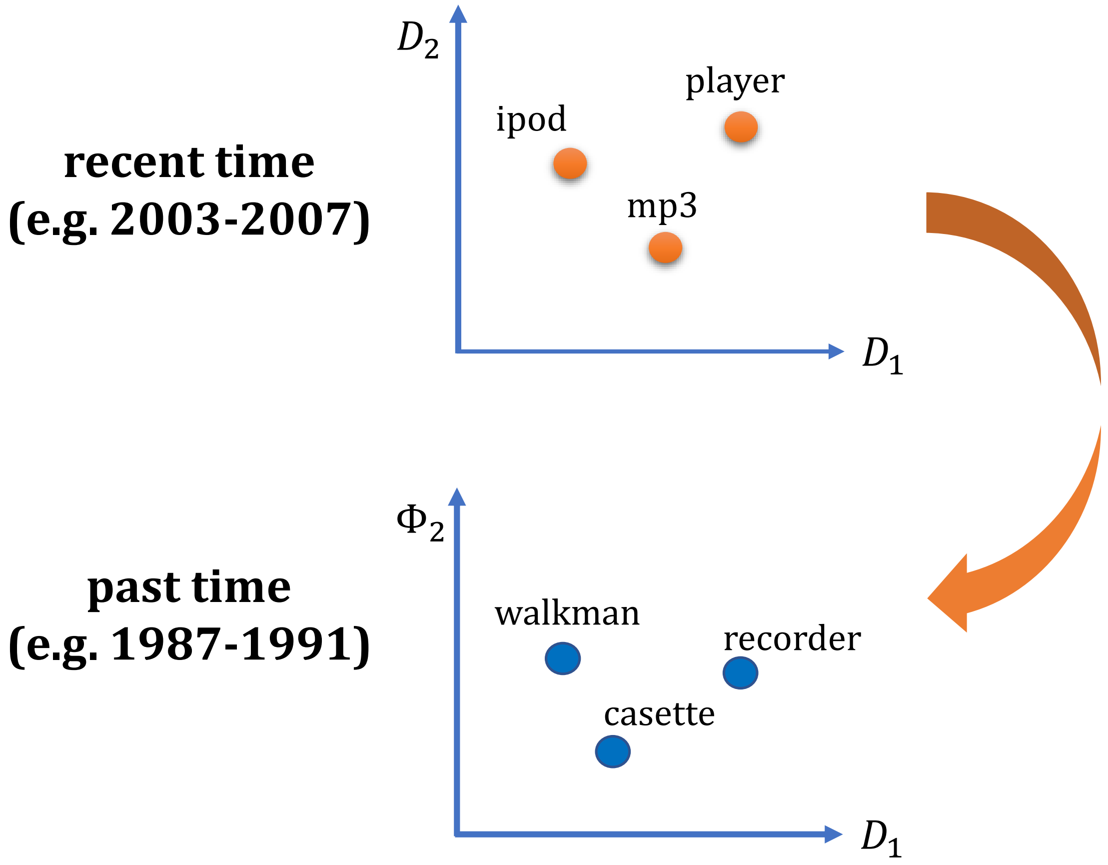
$$
\begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_P \end{array}
\begin{array}{c}
\begin{array}{cccc} D_1 & D_2 & \cdots & D_m \end{array} \\
\begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}
\end{array}
$$

$$
\begin{array}{c} \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_Q \end{array}
\begin{array}{c}
\begin{array}{cccc} \Phi_1 & \Phi_2 & \cdots & \Phi_n \end{array} \\
\begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}
\end{array}
$$

D_i and Φ_k are the dimensions of each vector space

T. Mikolov, K. Chen, G. Corrado and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. ICLR Workshop, 2013

Finding Temporal Analogy: Assumption behind Proposed Approach



Assumption
The <u>relative</u> positions of terms in each vector space remain stable

Finding Temporal Analogy: Constructing Transformation Matrix

Distributed Vector Representations

Recent time
(e.g. 2003-2007)

$$\begin{array}{c} \\ w_1 \\ w_2 \\ \vdots \\ w_P \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & \cdots & D_m \end{array} \\ \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

Past time
(e.g. 1987-1991)

$$\begin{array}{c} \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_Q \end{array} \begin{array}{c} \begin{array}{cccc} \Phi_1 & \Phi_2 & \cdots & \Phi_n \end{array} \\ \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

K Pairs of corresponding terms (anchors)
$\{(w_i, \omega_i), \ldots, (w_j, \omega_j)\}$

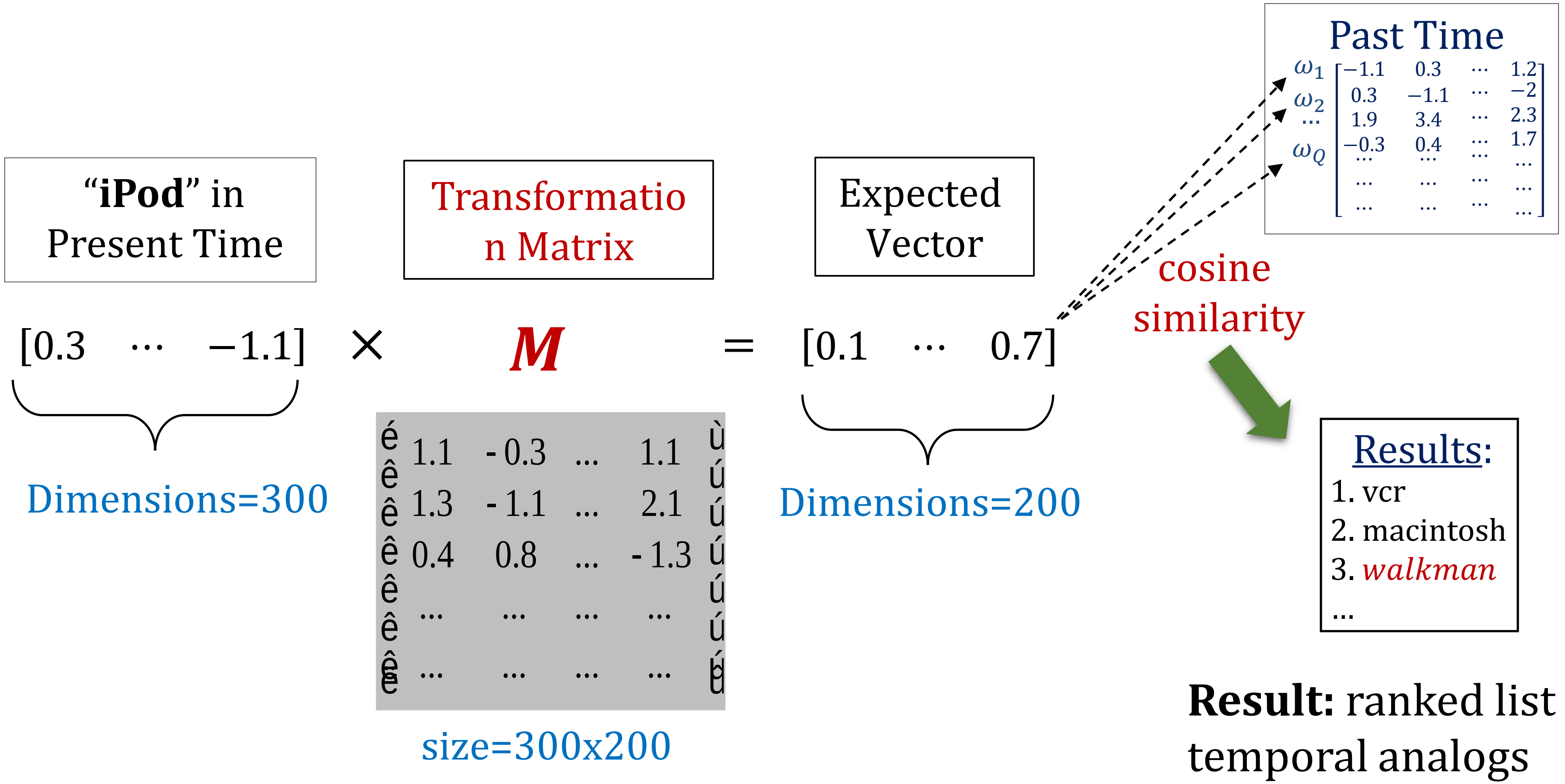
$$\mathbf{M} = \underset{\mathbf{M}}{\operatorname{argmin}} \sum_{i=1}^{u} \left\| \mathbf{M}\mathbf{x}_i^{\mathrm{b}} - \mathbf{x}_i^{\mathrm{t}} \right\|_2^2 + \gamma \left\| \mathbf{M} \right\|_2^2$$

$$M = \begin{array}{c} \\ D_1 \\ D_2 \\ \vdots \\ D_m \end{array} \begin{array}{c} \begin{array}{cccc} \Phi_1 & \Phi_2 & \cdots & \Phi_n \end{array} \\ \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix} \end{array}$$

Choose ***common, frequent terms***:
the more frequently the word used,
the harder is to change its meaning [Pargel 2007]
e.g. "man", "woman", "water"

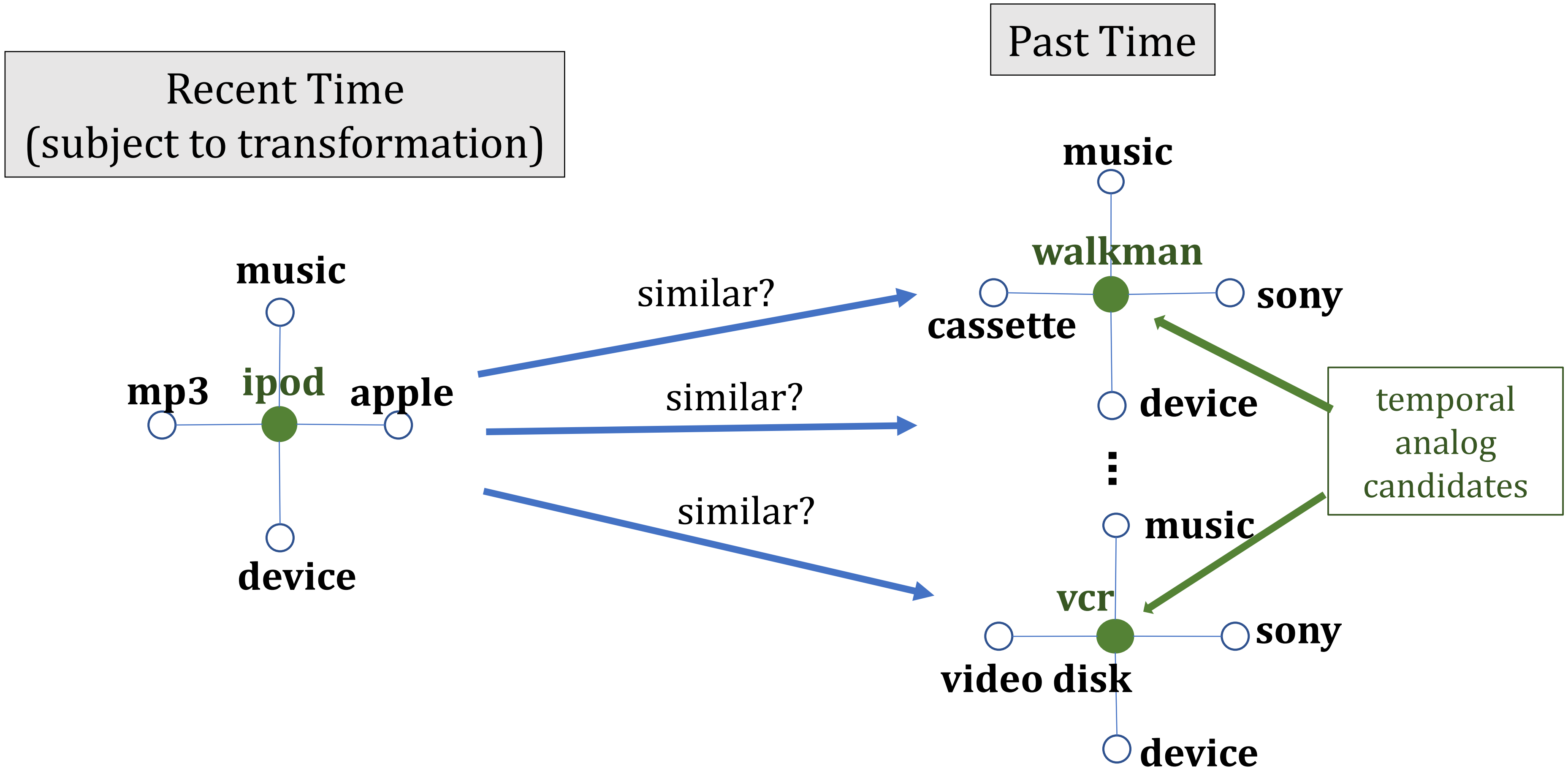
M. Pargel et al. *Frequency of word-use predicts rates of lexical evolution throughout Indo-European history*. Nature, 449, 717-720, 2007

Finding Temporal Analogy: Global Term Transformation Approach



Result: ranked list of temporal analogs

Finding Temporal Analogy: Transformation Using Local Graph by Using Reference Points



Finding Temporal Analogy: Desired Characteristics of Reference Points

- Reference Points - context terms which help to build effective across-time connection
- Desired criteria:
  a) have **high relation** with the query
  b) be sufficiently **general**
  c) **independent** from each other

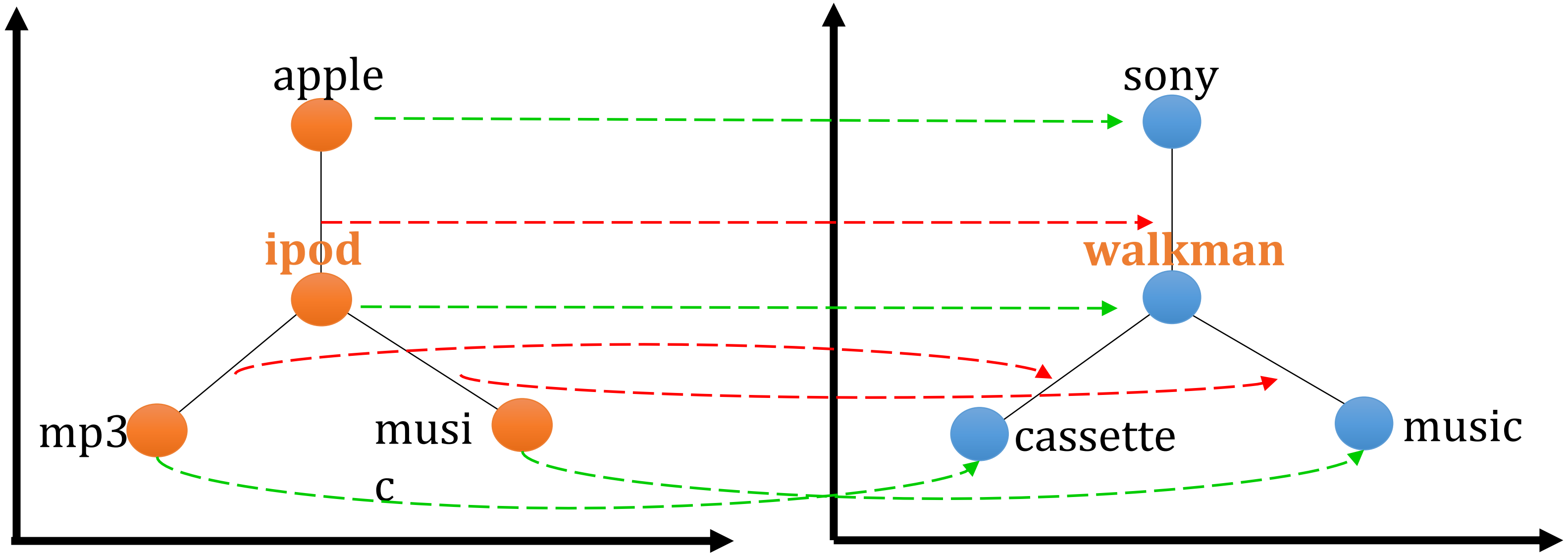
Finding Temporal Analogy: Local Graph Similarity Measurement

Approach:

Measure by concept similarity and relational similarity

absolute position

difference between vector representations



Recent Time
(e.g. 2003-2007)

Past Time
(e.g. 1987-1991)

Example Results

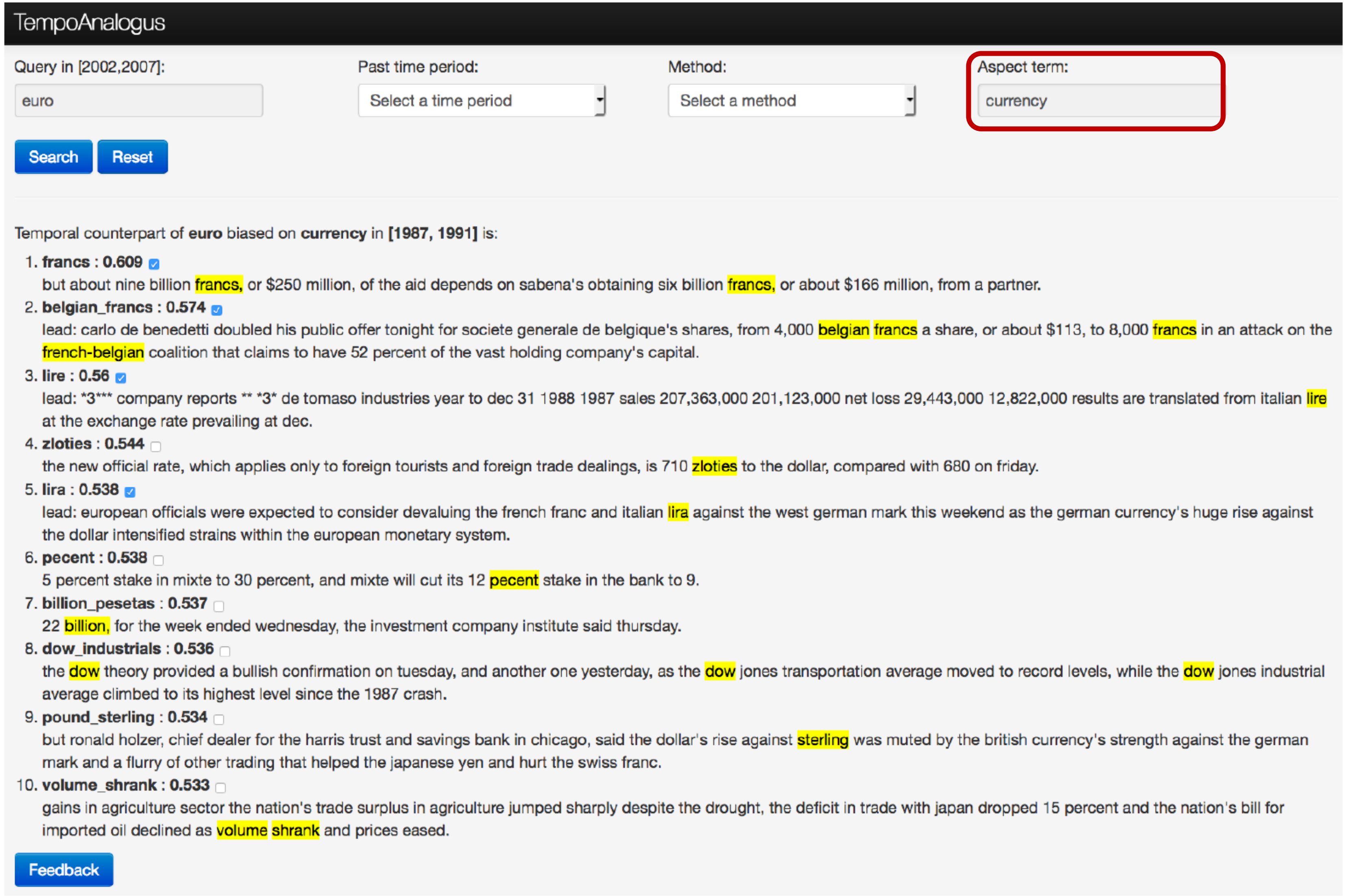
Lexico-Syntactic Pattern used to detect reference points

queries | correct answers | baselines | methods

	[2002,2007]	[1987,1991]	BOW (baseline)	LSI+Com (baseline)	Global_Tran	Local_Tran (Lex)
1	Putin	Yeltsin	1000+	51	24	2
2	Chirac	Mitterrand	1000+	6	7	2
3	iPod	Walkman	1000+	6	3	1
4	Facebook	Usenet	1000+	1000+	1	1
5	Linux	Unix	1000+	5	20	1
6	spam	junk mail	1000+	1000+	5	1
7	spreadsheet	database	1000+	395	3	1
9	email	messages	1000+	1	2	7
10	email	letters	1000+	1000+	1	1
11	email	mail	1000+	119	7	6
12	email	fax	1000+	1000+	3	4
14	superman	batman	1000+	46	5	2
15	Pixar	Tristar	1000+	110	1	1
16	Pixar	Disney	1000+	1	3	2
17	Euro	Mark	1000+	1000+	2	1
19	Euro	Franc	1000+	1000+	7	3
20	Myanmar	Burma	1000+	3	64	46
21	Koizumi	Kaifu	1000+	66	2	1
22	NATO	NATO	1000+	1	304	141
24	fridge	freezer	1000+	7	1	1
25	fridge	refrigerator	1000+	4	2	2
27	Serbia	Yugoslavia	1000+	12	1	1
28	Kosovo	Yugoslavia	1000+	27	14	10
30	mp3	compact disk	1000+	44	58	19
…	…	…	…	…	…	…

Rank of correct answers

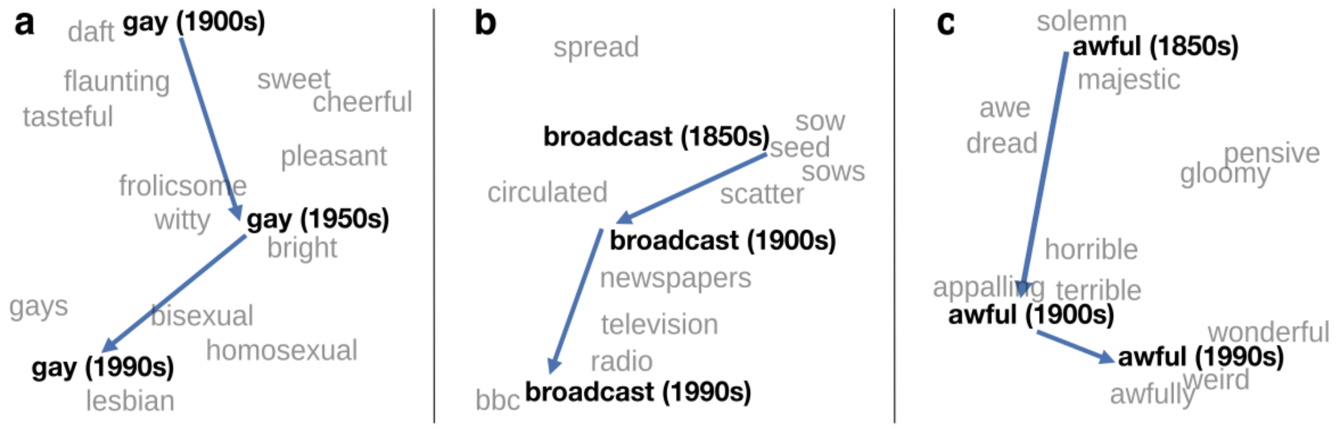
Finding Temporal Analogy: Aspect-based Retrieval System for Temporal Analogs



Y. Zhang, A. Jatowt, S. Bhowmick and Y. Matsumoto: *ATAR: Aspect-based Temporal Analog Retrieval System for Document Archives*, WSDM 2019

Embeddings as a window onto historical semantics

~30 million books, 1850-1990, Google Books data



Trained embeddings on different decades of historical text to see meanings shift

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. Proceedings of ACL.

Embeddings reflect cultural and other kinds of biases

- Ask "Paris : France :: Tokyo : x"
  - x = Japan
- Ask "father : doctor :: mother : x"
  - x = nurse
- Ask "man : computer programmer :: woman : x"
  - x = homemaker

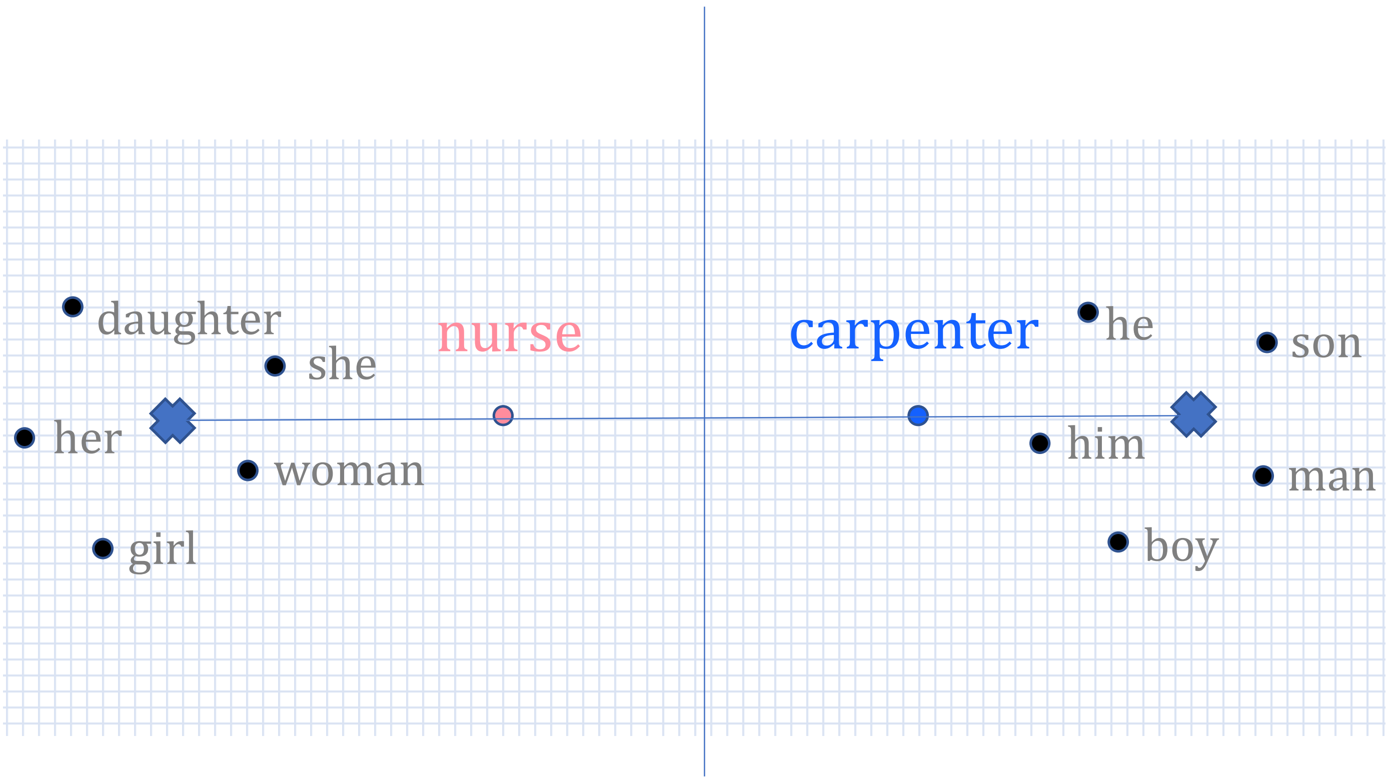
Algorithms that use embeddings as part of e.g., hiring searches for programmers, might lead to bias in hiring…

Hispanic	Asian	White
housekeeper	professor	smith
mason	official	blacksmith
artist	secretary	surveyor
janitor	conductor	sheriff
dancer	physicist	weaver
mechanic	scientist	administrator
photographer	chemist	mason
baker	tailor	statistician
cashier	accountant	clergy
driver	engineer	photographer

The top ten occupations most closely associated with each ethnic group in the Google News embedding (Garg, 2018)

Bolukbasi, Tolga, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. "Man is to computer programmer as woman is to homemaker? debiasing word embeddings." In *NeurIPS*, pp. 4349-4357. 2016.
Garg, N., Schiebinger, L., Jurafsky, D., and Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. Proceedings of the National Academy of Sciences 115(16), E3635–E3644.

Computing the gender bias of a word



How much closer a word is to "woman" synonyms than "man" synonyms?