COMP5046: Lexical Semantics and Distributional Similarity

Joel Nothman

joel.nothman@sydney.edu.au

School of Information Technologies
University of Sydney

2018-03-20

Part I

Vector Space Model

Do two texts (or words) mean the same thing?

- Need a measure of **similarity** between texts (or words)
- Want similarity of meaning, not just similar structure or word choice
- Allows us to:
  - find topical clusters in a collection of documents
  - retrieve documents relevant to a query
  - track topics through the media
  - etc.

Vector space models (VSM)

or meaning

- Represent content as a vector

$\Rightarrow$ d numeric elements per item

- Similarity in vector space implies similarity in meaning

- Language Models: the probability that some text belongs to a particular population
- Vector Space Models: measure the similarity of text pairs

Representing text as a bag of words

- **Bag of words** vector for each document
  - Vector elements correspond to words from the vocabulary
  - d = vocabulary size
  - Each element counts how many times the word appears
  - Disregards word choice, order and ambiguity
- **Term-document matrix**
  - A **sparse** matrix: don't store zeros

rat and rodent are the same but they have individual cols

Zipfs law: most word dont occur too often

(a) the rat bit the cat
(b) the cat ate the rat
(c) the cat ate the rodent

bite	cat	eat	rat	rodent	the
1	1	0	1	0	2
0	1	1	1	0	2
0	1	1	0	1	2

Or sparsely:
{bite: 1, cat: 1, rat: 1, the: 2}
{cat: 1, eat: 1, rat: 1, the: 2}
{cat: 1, eat: 1, rodent: 1, the: 2}

Measuring similarity

- similarity between vectors calculated by **cosine similarity**

normalization: does not matter on the length of document

$$cosine(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \, \|\mathbf{b}\|}$$

i.e. normalise vectors (divide $\mathbf{x}$ by $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$)
then take dot product $\sum_i a_i b_i$

- cosine(the cat sat on the blue mat, the rat sat on a big mat)
- $\mathbf{a} = \{$blue: 1, cat: 1, mat: 1, on: 1, sat: 1, the: 2$\}$
- $\|\mathbf{a}\| = \sqrt{1+1+1+1+1+4} = \sqrt{9} = 3 = \|\mathbf{b}\|$
-

$$cosine(\mathbf{a}, \mathbf{b}) = \underbrace{2(\tfrac{1}{3} \cdot 0)}_{\text{blue, cat}} + \underbrace{3(0 \cdot \tfrac{1}{3})}_{\text{a, big, rat}} + \underbrace{3(\tfrac{1}{3} \cdot \tfrac{1}{3})}_{\text{mat, on, sat}} + \underbrace{1(\tfrac{2}{3} \cdot \tfrac{1}{3})}_{\text{the}}$$

for (a,big,rat), a,big,rat all appears once in the second sentence hence 1/3. There are 3 terms, hence * by 3

Some words are more important than others

counts assume all words are equivalient

- Frequent words have too much weight with raw frequencies
- A classifier over bags of words will determine which are important
- To measure similarity, you need to prescribe importance
  - by adjusting the counts in your vectors
- Which words are most important?
- On the right:

```
brown = nltk.corpus.brown
df = Counter()
for doc in brown.fileids():
  df.update(set(brown.words(doc)))
```

Word in Brown	# docs
a	500
contrary	40
cope	20
prone	12
well-being	8
licensing	6
Sullivan	5
misdeeds	4
clump	3
Waiting	3
advises	2
reorientation	2
Quelch	1
high-gain	1
departmental	1
$26,000,000	1

The TF-IDF principle for weighting vectors

- **term frequency** component
  - count of term in document
  - or square root, or binary, ...
- times **inverse document frequency**
  - DF_t = number of documents where term t appears
  - IDF usually $\log \frac{N}{DF_t}$
  - N is number of documents in collection
- called **TF-IDF**
- raw counts prefer long documents
  - normalise for document length
  - cosine similarity does this for you

how many docs has the word in the collection e.g. a will be in every doc, N=DFt, making log 1 = 0

Word in Brown	IDF
a	0
contrary	3.6
cope	4.6
prone	5.4
well-being	6.0
licensing	6.4
Sullivan	6.6
misdeeds	7.0
clump	7.4
Waiting	7.4
advises	8.0
reorientation	8.0
Quelch	9.0
high-gain	9.0
departmental	9.0
$26,000,000	9.0

Information retrieval (IR)

- Rank documents according to their relevance to a query
  - web search
  - email archive search
  - ...
- Baseline approach:
  cosine similarity over TF-IDF-weighted bag of words
- See work of Gerry Salton

- Sometimes frequent words are important, e.g. The Who

Meaning as a vector

- We want to compare the meanings of texts
- Represent a text as a vector of **features**
- Similarity in vector space should mean similarity in meaning

- Features = count of each word is a good start
- For similarity, feature weighting is important
- Use a sparse vector/matrix representation

- Soon: representing word meaning as a vector

Part II

Lexical Semantics

The Meaning of Words

- Lexical semantics studies the meaning of words or **lexemes**
- lexemes are usually single 'words' in English:
  - the **dog**
  - the **dog**s
- but sometimes are more or less than one 'word':
  - the **hot dog**s (more than one word)
  - perhaps is**n't** (less than one word)

What's the meaning of 'meaning'?

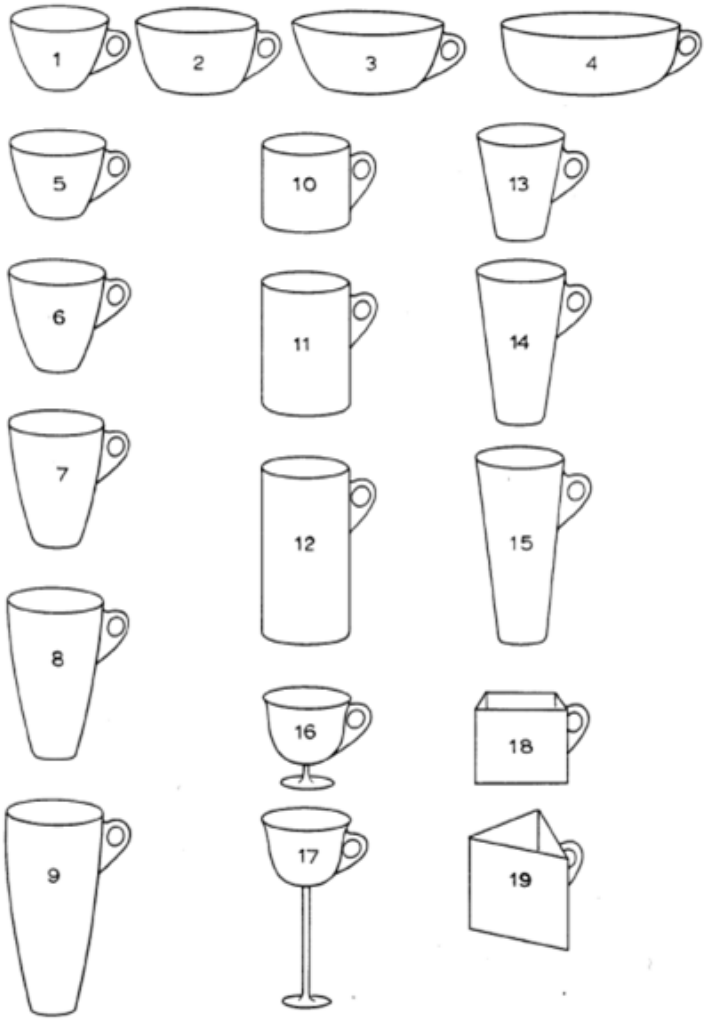
- a meaning is modelled in terms of other meanings

Example (Definitions from the American Heritage Dictionary)

red n. the color of blood or a ruby.
blood n. the red liquid that circulates in the heart, arteries and veins of animals.

- can any set of information really capture 'dogness'?
  - $\Rightarrow$ description can never equate to experiencing

Putting bounds on a meaning



cup? bowl? vase?

- handle or none
- relative diameter
- cross-section
- stem
- function

Labov (1973)

Baseline lexical semantics model

- assume text is composed of words
- each word is a semantic atom with one meaning unique to that word
- as in bag of words
- text meaning is logical composition of word meanings

- Reality is a long way from this:
  - meaning can be evoked by a multi-token span
    e.g. kicked the bucket
  - single 'word' can have many different meanings
  - meanings are related to each other
    (part/whole, is-a, similarity, etc)

Lexical semantics in NLP

- word sense disambiguation working out which meaning of word is evoked in text
- word sense induction group occurances of word into different meanings
- automatic lexicon construction building a collection of definitions of each word
- word meanings as vectors vectorization

One word can mean many things

- some words are historically unrelated, but spelled (*homograph*) or pronounced (*homophone*) the same, e.g. bass
- others may have historic links, but are now very different, e.g. bank
- others have a tangle of closely related meanings

One word can mean many things

- some words are historically unrelated, but spelled (*homograph*) or pronounced (*homophone*) the same, e.g. bass
- others may have historic links, but are now very different, e.g. bank
- others have a tangle of closely related meanings, e.g.
  - "a **set** of books"; "a **set** of golf clubs"; "a **set** of teeth"
  - "the **set** of prime numbers is infinite"
  - "he did four **set**s of the incline bench press"
  - "the production's **set**s were meticulously authentic"
  - "in the rise and **set** of the sun"
  - "the **set** of his mind was obvious"
  - "he gave a final **set** to his hat"
  - "they played two **set**s of tennis after dinner"
  - "he waited for the concrete to **set**"

Terminology

- We say that a lexeme has multiple **senses** each sense has a different meaning
- It is **polysemous**

Word Sense Disambiguation

- **resolve** a term in context to **one of a set** of meanings
- for all words, all nouns/verbs, or specific words
- Lesk (1986): match **context** to dictionary definition
  - Have you put your money in the **bank**?
  - The rabbit climbed up the **bank**

find similarity between sentence with word and meaning with tf-idf, cosine or bag of words

- Macquarie Dictionary:

bank[1] *n* 1. a long pile or mass: bank of earth; bank of snow; bank of clouds. 2. a slope or acclivity. 3. *Physical Geography* the slope immediately bordering the course of a river along which the water normally runs. 4. *Oceanography* a broad submarine elevation . . .

bank[2] *n* 1. an institution for receiving and lending money (in some cases, issuing notes or holding current accounts that serve as money) and transacting other financial business. 2. the office or quarters of such an institution. 3. *(in games)* a. the stock or fund of pieces . . .

bank[3] *n* 1. an arrangement of objects in line. 2. *Music* a row of keys . . .

Methodology for determining word senses

1. collect instances of the word in context;
2. divide the instances into clusters, so that, as far as possible, members of each cluster have much in common with each other, and little in common with members of other clusters;
3. for each cluster, work out what it is that makes its members belong together, and write a definition

- The boundaries between clusters are often very blurred
- Exercise: Look up a common word, e.g. speak, in multiple dictionaries

Not all variations of meaning are considered polysemy

- **Polysemy**: a context **selects** a different sense
  - Have you put your money in the **bank**?
  - The rabbit climbed up the **bank**
- **Modulation**: a context refers to different **aspect** of a meaning:
  - He doesn't often oil his **bike**.
  - Madeline dried off her **bike**.
  - Boris's **bike** goes like the wind.

Zeugma: are two senses distinct? use "and"

Zeugma - a test

- Which of those flights serve breakfast?
- Does Midwest Express serve Philadelphia?
- ?Does Midwest Express serve breakfast and Philadelphia?
  (from Jurafsky and Martin)

- An example of a **test** used in linguistics to distinguish categories

Conclusions on word sense disambiguation

- sense inventories are subjective and genre-specific
- an inventory is a useful body of human knowledge about words and meanings
- the truth is closer to soft clusters: clusters overlap, and are often co-activated

- taking the **most frequent sense** often performs well
- for specific applications, e.g. MT or IR, it is better to take a task-driven approach

machine translation

info retrival

Relating meanings to each other

- We can account for polysemy by mapping each word to multiple meanings
- ... but what about sparse data problems? Can we generalise from one of these to the other:
  - color vs. colour
  - resume vs. CV
  - cosmonaut vs. astronaut
  - diplodocus vs. dinosaur
  - leaves vs. tree

Hyponymy/Hypernymy: the IS-A relation

- a *hypernym* is a super-set of a *hyponym*
  (hyper- → large, hypo → small)
- hypernymy is a good way to make taxonomies/ontologies:
  dog → canid → mammal → ... → physical thing

- animals, plants etc are easy to taxonomise
- abstract concepts and man-made things are more difficult
- verbs and adjectives are pretty much impossible

Other relations

- a *meronym* is a part of a whole
  e.g. leaf, branch, trunk are meronyms of tree
- a *holonym* is the inverse (whole of the part)
  e.g. car is a holonym of engine, wheel
- *antonyms* are opposite meanings
  e.g. rich vs. poor. Often have similar distributions
- *troponyms* denote a specific manner of something
  e.g. chuckle to laugh, march to walk

- see WordNet (Miller, 1995); `nltk.corpus.wordnet`

Synonymy: merging two meanings

- two senses are **synonyms** if they are interchangeable in all contexts
- ... by this definition, almost no senses are synonymous!
- more useful to think of degrees of synonymy: the number of contexts the senses are interchangeable in
- also useful to think of *similarity*: how similar the contexts the two words inhabit are

Part III

Distributional Similarity

Do you know what a *blag* is?

Do you know what a *blag* is?

- The **blag** bit the postman.

Do you know what a *blag* is?

- The **blag** bit the postman.
- The big hairy **blag**...

Do you know what a *blag* is?

- The **blag** bit the postman.
- The big hairy **blag**...
- He was walking his **blag**.

Do you know what a *blag* is?

- The **blag** bit the postman.
- The big hairy **blag**...
- He was walking his **blag**.
- The **blag** barked.

Do you know what a *blag* is?

- The **blag** bit the postman.
- The big hairy **blag**...
- He was walking his **blag**.
- The **blag** barked.

- Now do you know what a blag is?

We can learn word meaning from context

- all there was available to learn blag was the context
- needed no *grounding* or *definition*
- e.g. no real world example, photograph or other representation

Distributional Hypothesis (Harris, 1954)

Similar terms

appear

in similar contexts

Distributional Hypothesis (Harris, 1954)

Similar terms

appear

in similar contexts

- what do we mean by *terms*?

- single words
- multi-word expressions (e.g. verb particle constructions)
- relationships between words in a sentence
  e.g. paths in a dependency graph (Lin and Pantel, 2001)

Distributional Hypothesis (Harris, 1954)

Similar terms

appear

in similar contexts

- what do we mean by *similar terms*?

- we are interested in finding synonyms
- and what about antonyms?
- but hypernyms/hyponyms are important too
- except they are *asymmetric* relationships

Distributional Hypothesis (Harris, 1954)

Similar terms

appear

in similar contexts

- what do we mean by *contexts*?
- there are many different definitions of context

Distributional Hypothesis (Harris, 1954)

Similar terms

appear

in similar contexts

- what do we mean by *similar contexts*?

- compare vectors (or distributions) of contexts

Distributional Similarity $\neq$ Co-occurrence

- distributional similarity requires shared context
- the terms themselves don't have to appear together
- i.e. distributionally similar terms need not co-occur
- this is important since *synonyms* don't always co-occur
  e.g. color vs colour

Thesaurus construction can be broken into stages

extract	extract contextual information
collect	vector of context counts for each term
compare	measure similarity between vectors
rank/cluster	return words with most similar contexts

Thesaurus construction can be broken into stages

extract	extract contextual information
collect	vector of context counts for each term
compare	measure similarity between vectors
rank/cluster	return words with most similar contexts

But how do we know if we've done a good job?

Application-based (extrinsic) evaluation

- smoothing language models (Dagan et al., 1994, 1995)
- word sense disambiguation (Dagan et al., 1997; Lee, 1999)
- information retrieval (Grefenstette, 1994)
- many others for other techniques, e.g.
  - malapropism detection (Budanitsky and Hirst, 2001)
  - collocation extraction (Pearce, 2001)
- can be biased by the properties of application
- can be obscured by other parts of the system

Gold standard-based (intrinsic) evaluation

gold standard = truth

- collect lists of known synonyms
  compare to top ranked predictions returned by system
- psycholinguistic evidence:
  - free word association (e.g. Deese, 1962)
  - distance judgements (e.g. Rubenstein and Goodenough, 1965)
  - lexical priming (McDonald, 2000)
- TOEFL vocabulary tests (Landauer and Dumais, 1997)
- paper and electronic thesauri
  (Grefenstette, 1994; Curran, 2004; Gorman, 2005)
- intrinsic evaluation alone does not measure how useful the technology is

. . . but polysemy makes using thesauri messy

New Oxford Thesaurus of English: **company ▶ noun**

❶ *he works for the world's biggest oil company* **firm**, business, corporation, house, establishment, agency, office, bureau, institution, organization, operation, concern, enterprise, venture, undertaking, practice; conglomerate, consortium, syndicate, group, chain, combine, multiple, multinational; *informal* outfit, set-up.

❷ *I was greatly looking forward to the pleasure of his company* **companionship**, presence, friendship, fellowship, closeness, amity, camaraderie, comradeship; society, association.

❸ *I'm expecting company* **guests**, a guest, visitors, a visitor, callers, a caller, people, someone; *archaic* visitants.

❹ *he disentangled himself from the surrounding company of poets* **group**, crowd, body, party, band, collection, assembly, assemblage, cluster, flock, herd, horde, troupe, swarm, stream, mob, throng, congregation, gathering, meeting, convention; *informal* bunch, gang, gaggle, posse, crew, pack;

❺ *he recognized the company of infantry as French* **unit**, section,

Measures of retrieval performance

G is set of gold standard synonyms; S is set of system-proposed synonyms

Precision $P = \frac{|G \cap S|}{|S|}$ measures:

- what proportion of system synonyms are correct?
- spurious / false positive / type I errors

Recall $R = \frac{|G \cap S|}{|G|}$ measures:

- what proportion of gold synonyms are recovered?
- missing / false negative / type II errors

F-score combines these: $F_1 = \frac{2PR}{P+R}$
Or prefer precision/recall: $F_\beta = (1 + \beta^2)\frac{PR}{\beta^2 P+R}$

Evaluation practice

- choose your primary objective metric in advance
- compare to a baseline
  - a trivial system
  - a random prediction
  - a standard approach
  - a state of the art system

Context: Locality

- window of *n* words around term
  - terms within the window are the features
  - window sizes vary (± 1 to ± 250)
  - phrase, sentence and document boundaries
- sentence and document level co-occurrence
  - this is typical in Information Retrieval and in Topic Modelling
  - IDs are the features
  - called a *term-document matrix*
- there are precision/recall trade-offs with locality

i.e. big window

  - more general contexts allow more collisions (recall)
  - more specific contexts allow for finer distinctions (precision)

Context: Linguistic Structure

- *n*-word window contexts may include relative position
- filtering on stop words or syntactic category (e.g. nouns only)
- normalising words by case folding, stemming, etc.
- grammatical relations
  - many variant schemas and dependency grammars
  - Universal Dependencies (McDonald et al., 2013)
- extracting linguistic structure affects precision: . . . how?

Context: Linguistic Structure

- *n*-word window contexts may include relative position
- filtering on stop words or syntactic category (e.g. nouns only)
- normalising words by case folding, stemming, etc.
- grammatical relations
  - many variant schemas and dependency grammars
  - Universal Dependencies (McDonald et al., 2013)
- extracting linguistic structure affects precision:
  - grammatical relations allow finer distinctions (higher precision)
  - parsing errors introduce noise (lower precision)

Example contexts for idea

Window Context	Grammatical Context
(-1, good)	(ADJ, good)
(-1, faintest)	(ADJ, faintest)
(-2, have)	(DOBJ, have) direct object
(-3, toy)	(IOBJ, toy) indirect object
(-1, preconceived)	(ADJ, preconceived)
(-1, foggiest)	(ADJ, foggiest)

Sextant (Grefenstette, 1994)

`The man wearing bell-bottom trousers spoke .`

- raw text from the 100 million word British National Corpus

Sextant (Grefenstette, 1994)

The man wearing bell-bottom trousers spoke .

The_{DT} man_{NN} $\text{wearing}_{\text{VBG}}$ $\text{bell-bottom}_{\text{JJ}}$ $\text{trousers}_{\text{NN}}$ $\text{spoke}_{\text{VBD}}$.
$[\text{The man}]_{\text{NP}}$ $[\text{wearing}]_{\text{VP}}$ $[\text{bell-bottom trousers}]_{\text{NP}}$ $[\text{spoke}]_{\text{VP}}$.

- POS tag and identify phrasal chunks

Sextant (Grefenstette, 1994)

`The man wearing bell-bottom trousers spoke .`

`(man, SUBJ, wearing)` `(trousers, ADJ, bell-bottom)`
`(trousers, SUBJ, spoke)`* `(trousers, DOBJ, wearing)`

- rule-based detection of some grammatical relations

Sextant (Grefenstette, 1994)

`The man wearing bell-bottom trousers spoke .`

`(man, SUBJ, wear)` `(trousers, ADJ, bell-bottom)`
`(trousers, SUBJ, speak)` `(trousers, DOBJ, wear)`

- normalise to lemmas
- using the Sussex morphological analyser (Minnen et al., 2000)

Sextant (Grefenstette, 1994)

`The man wearing bell-bottom trousers spoke .`

- feature vector for trousers includes:

(SUBJ, wear) 133	(NN-APPOS, corduroy) 33
(NN-APPOS, shirt) 59	(IOBJ, dress) 32
(ADJ, black) 51	(DOBJ, get) 32
(ADJ, baggy) 42	(NN-APPOS, jacket) 32

- represents the distribution of syntactic contexts a word appears in

Efficiency is an important factor

- window methods are extremely fast (minutes)
- linguistic methods can me much much slower (hours to days)
- but can produce much better quality context information

Efficiency is an important factor

- window methods are extremely fast (minutes)
- linguistic methods can me much much slower (hours to days)
- but can produce much better quality context information
- however, more data *can* trump better quality
  (Curran and Moens, 2002)
- given (near) unlimited raw text, efficiency becomes important

Measures of similarity

- similarity calculated by pairwise vector comparison

$\Rightarrow$ rank candidates by similarity

- factorize the similarity metric into:
  - a vector comparison function (the measure)
  - an feature weighting function (the weight)

Measures from information retrieval

- comparing w_1 to w_2 word1 vs word2
- as a function of context term w' with relation r

- Cosine

$$\frac{\sum \text{weight}(w_1, *_r, *_{w'}) \times \text{weight}(w_2, *_r, *_{w'})}{\sqrt{\sum \text{weight}(w_1, *_r, *_{w'})^2 \times \sum \text{weight}(w_2, *_r, *_{w'})^2}}$$

dont worry about the notations, just need to know when to use which

- Weighted versions of Dice and Jaccard similarity (Grefenstette, 1994)

Information Theoretic measures

- consider task as comparing two conditional distributions:

$$p = P(*|w_1) \text{ and } q = P(*|w_2)$$

* is context of word

- Kullback-Leibler divergence (or relative entropy)

how likely one distribution represents a sample from another distribution

$$D(p||q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$$

- Jensen-Shannon divergence

$$A(p,q) = D(p||\frac{p+q}{2}) + D(q||\frac{p+q}{2})$$

Weight functions: a principle

- the weight function should encode:
  - association strength between word and feature
  - typicality of the feature over all feature vectors

feature i.e. context

(NN-APPOS, corduroy)	33	0.0630
(ADJ, baggy)	42	0.0512
(ADJ, moleskin)	10	0.0508
(NN-APPOS, shirt)	59	0.0444
(ADJ, bell-bottom)	4	0.0380
(ADJ, flared)	15	0.0331

Weight functions from IR

- simple weight functions
  identity, frequency, relative frequency
- term frequency $\times$ inverse document frequency (TF-IDF)

$$f(w, r, w') \cdot log \frac{n(*, *, *)}{n(*, r, w')}$$

Weight functions: association measures

- informative features are strongly correlated with the term
- this is like the collocation extraction problem earlier
- use statistics from Manning and Schütze (1999) ch. 5: t-test, χ^2, likelihood ratio, mutual information

- t-test

$$\frac{p(w, r, w') - p(*, r, w')p(w, *_r, *_{w'})}{\sqrt{p(*, r, w')p(w, *_r, *_{w'})}}$$

- pointwise mutual information

$$\log\left(\frac{p(w, r, w')}{p(w, *_r, *_{w'})p(*, r, w')}\right)$$

Efficiency

- determining similarity between all pairs is costly
- matrix is (# distinct terms) $\times$ (# distinct contexts) $=: n \times m$
- sparse matrix: multiplication in $O(nk^2)$
  where $k = \max_{w=1,\ldots,n}(\#\text{ contexts for } w)$
- billion word datasets, highly discriminative contexts

$\Rightarrow$ comparisons become rather inefficient

Therefore:

- compress the matrix (we'll cover this later)
  to dense $n \times m'$ for $m' \ll m$ i.e. reduce number of features/contexts
- reduce the number of comparisons

Approximate matching

- used to eliminate unlikely similar terms quickly
- reduce the $O(n^2)$ vector comparisons
- preferably to linear or log-linear time complexity
- if similarity is reasonably close then do full vector comparison

important

- **Approximate Nearest Neighbors** search
- e.g. locality sensitive hashing (LSH)
  - near vectors more likely to have same hash value
  - index vectors with many hash functions
  - only compare vector to others which share a hash

not important

Where to make a cutoff?

- in some tasks we may use the vector representation directly
- but often we need hard sets
- we can only rely on the first few synonyms
- we don't know how to select a cutoff

Other relations are often identified

- other related concepts with similar contextual distributions
- this is called the tennis problem:
  ball, racquet, net, ...
- the worst case is antonyms antonyms could have very similar contexts
- less problematic are hypernyms/hyponyms
  (particularly problematic for symmetric measures)

Polysemy muddies the water

- polysemy merges contexts for two/more (latent) concepts
- WSD not good enough to use reliably
- some work attempts to overcome these problems:
  - identify dominant sense (McCarthy et al., 2004a)
  - finding the other senses (McCarthy et al., 2004b)
- clustering contexts or hypothesised synonyms
  - Word Sense Induction
  - multi-prototype neural network embeddings

- need to identify **semantic shifts** and new senses

  e.g. *mouse* and keyboard
  has 200 *likes*

Take away

- Measuring meaning similarity in vector space
- Terms: lexeme, sense, polysemy, synonymy, hyponymy, precision, recall, F_1, bag of words, word sense disambiguation, word embedding
- Distributional hypothesis of meaning
- Principle behind reweighting (e.g. TF-IDF)
- Need for data structures and algorithms for efficiency (e.g. sparse vectors, LSH, compression)
- Intrinsic and extrinsic evaluation
- Trade-off between precision and recall