CS-AD 220 – Spring 2016

Natural Language Processing

Session 26: 3-May-16

Prof. Nizar Habash

Slides from Jurafsky and Martin with modifications

NYUAD Course CS-AD 220 – Spring 2016

Natural Language Processing

Assignment #4

Phrase-based Statistical Machine Translation

Assigned Apr 19, 2016

Due May 10, 2016 (11:59pm)

Introduction[1]

In this laboratory exercise, you will build a complete phrase-based statistical machine translation system from small amounts of training data, evaluate their performance, and identify ways that translation quality can be improved. Resulting systems will be evaluated on test data (released a few days before the deadline). You will build the MT system using Moses, an open-source phrase-based statistical machine translation decoder.

Assignment #4 posted on NYU Classes

START EARLY!

DEADLINE IS May 10 (11:59pm)

MT Assignment Update

1. The issue with the data that cause some problems have been fixed. Please download the new data sets from the files directory on the class website:

   https://sites.google.com/a/nyu.edu/nyuad-cs-ad-220-natural-language-processing-spring-2016/files/Assignment-4-data.zip

2. To be able to put the files on the virtual machine and to take the output out, you need to set up a shared folder. I put some instructions on how to do this on the class website:

   https://sites.google.com/a/nyu.edu/nyuad-cs-ad-220-natural-language-processing-spring-2016/files/SharedFolderVirtualBox.pdf

Update to Syllabus

- May 10 and 12 changed again
- May 12 will include review for final
- **Final is May 16, 1pm to 4pm in CR-002**

26	Tue 3rd May	Lexical Semantics	J+M Chap 20	
	Thu 5th May	**No Class - Isra and Miraj Holiday**		
27	Tue 10th May	Question Answering and Summarization	none / get Assignment 4 finished!	
28	Thu 12th May	Sentiment Analysis / Review for Final	J+M Chap 23	
	Mon 16th May	FINAL EXAM 1pm - 4pm (CR-002)	All previous readings starting with J+M Chapter 5	

Word Similarity

- **Synonymy**: a binary relation
  - Two words are either synonymous or not
- **Similarity** (or **distance**): a looser metric
  - Two words are more similar if they share more features of meaning
- Similarity is properly a relation between **senses**
  - The word "`bank`" is not similar to the word "`slope`"
  - Bank^{1} is similar to fund^{3}
  - Bank^{2} is similar to slope^{5}
- But we'll compute similarity over both words and senses

Why word similarity

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

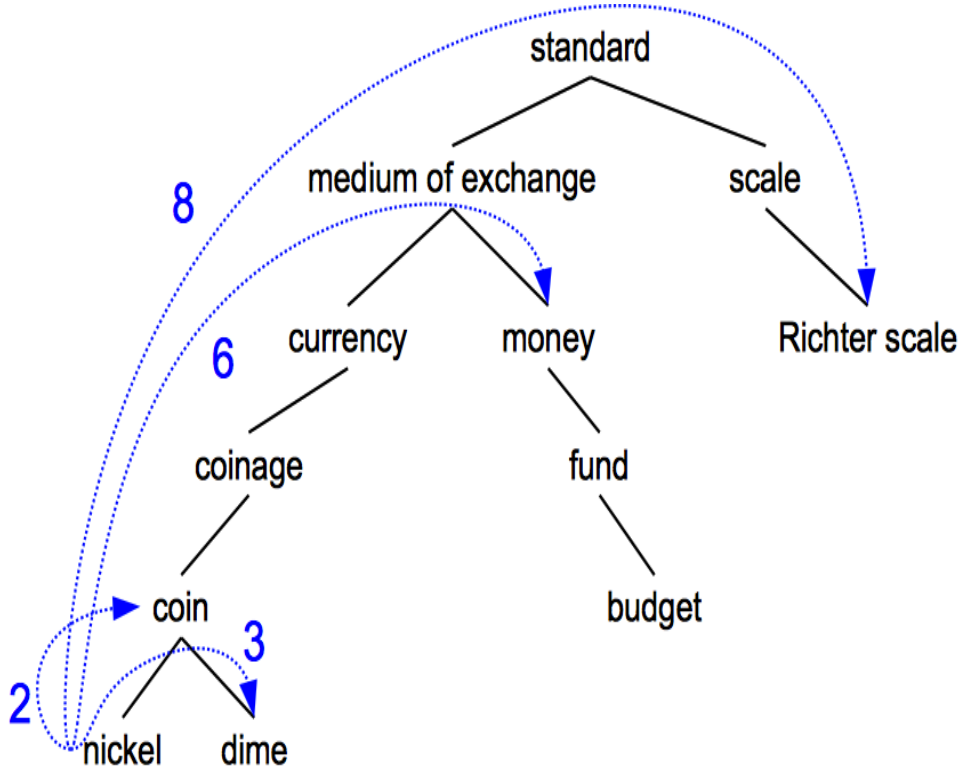
Word similarity and word relatedness

- We often distinguish **word similarity** from **word relatedness**
  - **Similar words**: near-synonyms
  - **Related words**: can be related any way
    - `car, bicycle`: **similar**
    - `car, gasoline`: **related**, not similar

Two classes of similarity algorithms

- Thesaurus-based algorithms
  - Are words “nearby” in hypernym hierarchy?
  - Do words have similar glosses (definitions)?
- Distributional algorithms
  - Do words have similar distributional contexts?

Path based similarity



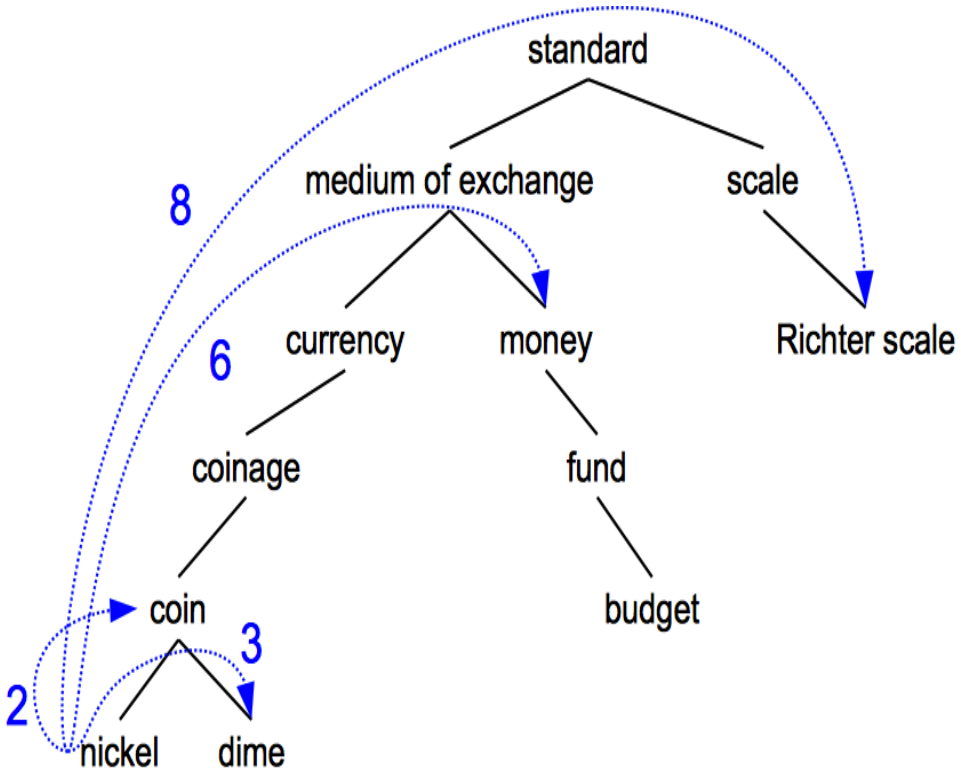
- Two concepts (senses/synsets) are similar if they are near each other in the thesaurus hierarchy
  - have a short path between them
  - concepts have path 1 to themselves

Refinements to 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
- $$\text{simpath}(c_1,c_2) = \frac{1}{\text{pathlen}(c_1,c_2)}$$
  - ranges from 0 to 1 (identity)
- $$\text{wordsim}(w_1,w_2) = \max_{\substack{c_1 \in \text{senses}(w_1),\\ c_2 \in \text{senses}(w_2)}} \text{sim}(c_1,c_2)$$

Example: path-based similarity

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



simpath(*nickel,coin*) =

simpath(*fund,budget*) =

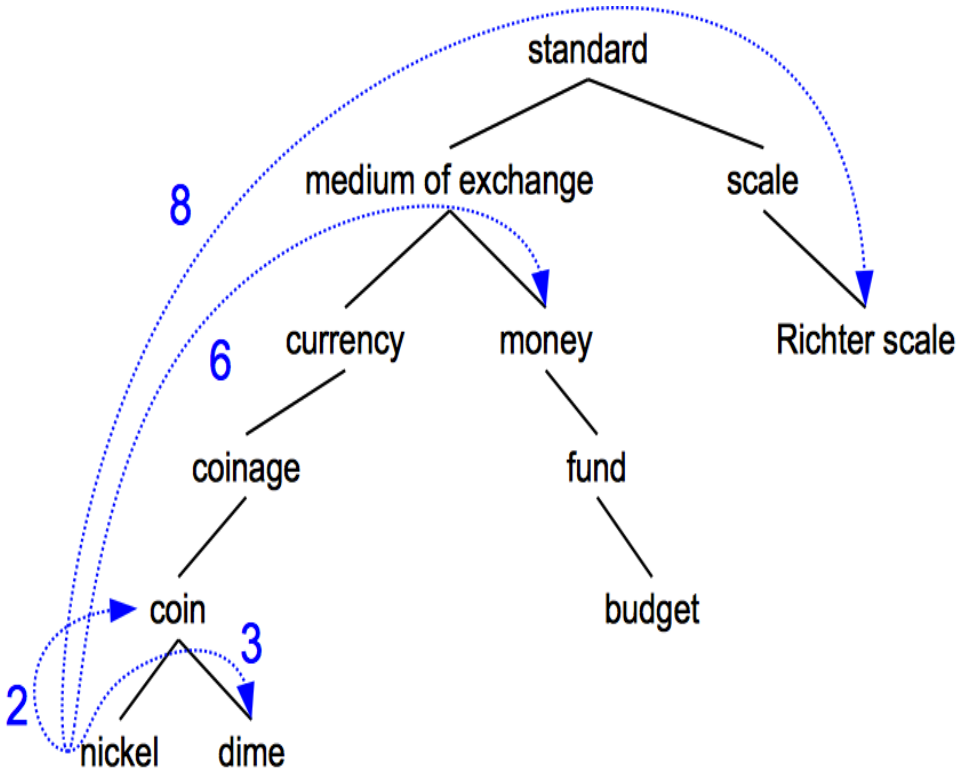
simpath(*nickel,currency*) =

simpath(*nickel,money*) =

simpath(*coinage,Richter scale*) =

Example: path-based similarity

simpath(c_1,c_2) = 1/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

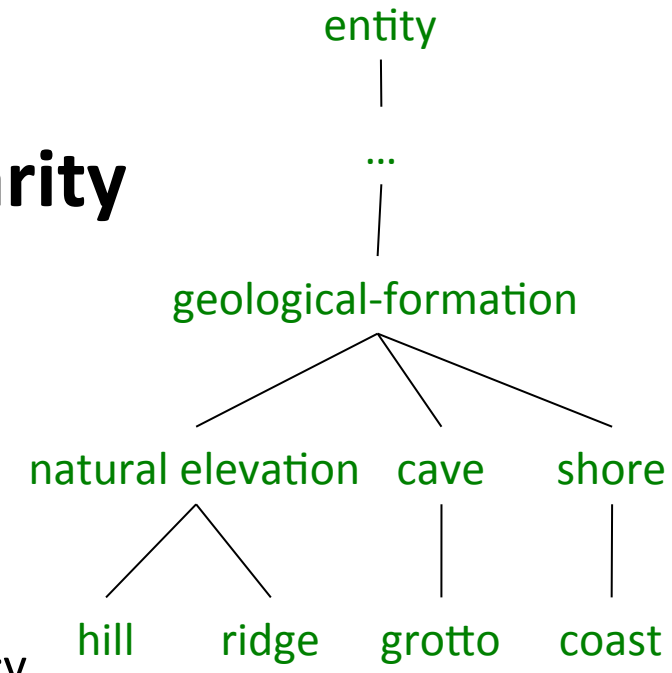
- Assumes each link represents a uniform distance
  - But *nickel* to *money* seems to us to be closer than *nickel* to *standard*
  - Nodes high in the hierarchy are very abstract
- We instead want a metric that
  - Represents the cost of each edge independently
  - Words connected only through abstract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

Information content similarity metrics

Resnik 1995. Using information content to evaluate semantic similarity in a taxonomy. IJCAI

- Let's define $P(c)$ as:
  - The probability that a randomly selected word in a corpus is an instance of concept c
  - Formally: there is a distinct random variable, ranging over words, associated with each concept in the hierarchy
    - for a given concept, each observed noun is either
      - a member of that concept with probability $P(c)$
      - not a member of that concept with probability $1\text{-}P(c)$
  - All words are members of the root node (Entity)
    - $P(root)=1$
  - The lower a node in hierarchy, the lower its probability

Information content similarity



- Train by counting in a corpus
  - Each instance of `hill` counts toward frequency

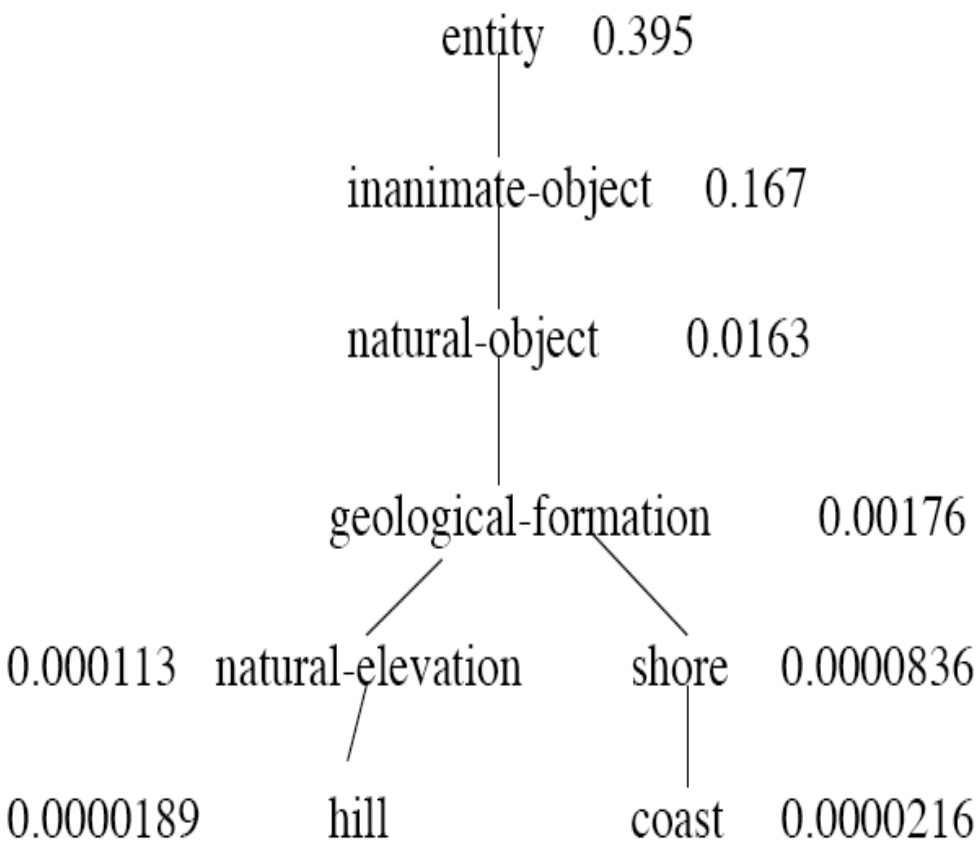
  of *natural elevation*, *geological formation*, *entity*, etc
  - Let $\text{words}(c)$ be the set of all words that are children of node c
    - words("geo-formation") = {hill,ridge,grotto,coast,cave,shore,natural elevation}
    - words("natural elevation") = {hill, ridge}

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

Information content similarity

- WordNet hierarchy augmented with probabilities P(c)

D. Lin. 1998. An Information-Theoretic Definition of Similarity. ICML 1998



Information content: definitions

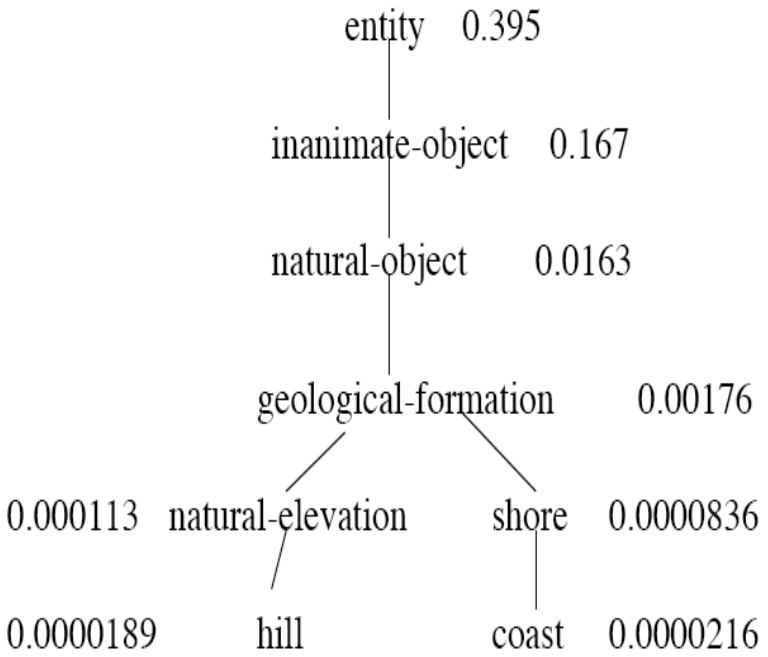
- Information content:

  $IC(c) = -\log P(c)$

- Most informative subsumer (Lowest common subsumer)

  $LCS(c_1,c_2) =$

  The most informative (lowest) node in the hierarchy subsuming both c_1 and c_2



Using information content for similarity: the Resnik method

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. IJCAI 1995.
Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. JAIR 11, 95-130.

- The similarity between two words is related to their common information
- The more two words have in common, the more similar they are
- Resnik: measure common information as:
  - The information content of the most informative (lowest) subsumer (MIS/LCS) of the two nodes
  - $\text{sim}_{\text{resnik}}(c_1,c_2) = -\log P(\text{LCS}(c_1,c_2))$

Dekang Lin method

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. ICML

- Intuition: Similarity between A and B is not just what they have in common
- The more **differences** between A and B, the less similar they are:
  - Commonality: the more A and B have in common, the more similar they are
  - Difference: the more differences between A and B, the less similar
- Commonality: IC(common(A,B))
- Difference: IC(description(A,B)-IC(common(A,B))

Dekang Lin similarity theorem

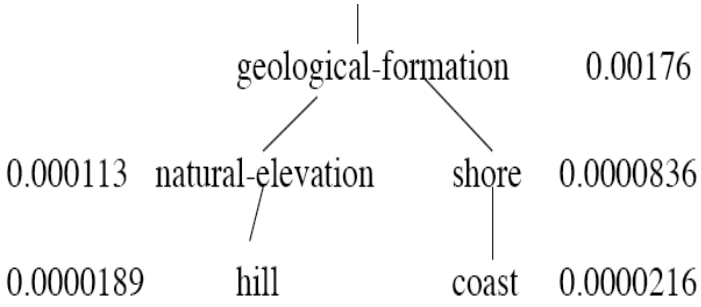
- The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are

$$sim_{Lin}(A,B) \propto \frac{IC(common(A,B))}{IC(description(A,B))}$$

- Lin (altering Resnik) defines IC(common(A,B)) as 2 x information of the LCS

$$sim_{Lin}(c_1,c_2) = \frac{2\log P(LCS(c_1,c_2))}{\log P(c_1)+\log P(c_2)}$$

Lin similarity function



$$sim_{Lin}(A,B) = \frac{2\log P(LCS(c_1,c_2))}{\log P(c_1)+\log P(c_2)}$$

$$sim_{Lin}(\text{hill},\text{coast}) = \frac{2\log P(\text{geological-formation})}{\log P(\text{hill})+\log P(\text{coast})}$$

$$= \frac{2\ln 0.00176}{\ln 0.0000189+\ln 0.0000216}$$

$$= .59$$

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 overlap also for other relations (RELS)
    - glosses of hypernyms and hyponyms

Summary: thesaurus-based similarity

$$\text{sim}_{\text{path}}(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)}$$

$$\text{sim}_{\text{resnik}}(c_1, c_2) = -\log P(LCS(c_1, c_2))$$

$$\text{sim}_{\text{lin}}(c_1, c_2) = \frac{2 \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)}$$

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

Libraries for computing thesaurus-based similarity

- NLTK
  - http://nltk.github.com/api/nltk.corpus.reader.html?highlight=similarity - nltk.corpus.reader.WordNetCorpusReader.res_similarity

- WordNet::Similarity
  - http://wn-similarity.sourceforge.net/
  - Web-based interface:
    - http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi
    - http://ws4jdemo.appspot.com/

WS4J Demo

WS4J (WordNet Similarity for Java) measures semantic similarity/relatedness between words.

Type in texts below, or use: example words | example sentences

1.	Input mode	(•) Word ( ) Sentence
2.	Word 1	mouse#n#1
3.	Word 2	bat#n#1
4.	Submit	Calculate Semantic Similarity

Summary

wup(mouse#n#1 , bat#n#1) = 0.8889

jcn(mouse#n#1 , bat#n#1) = 0.1167

lch(mouse#n#1 , bat#n#1) = 2.3026

lin(mouse#n#1 , bat#n#1) = 0.5671

res(mouse#n#1 , bat#n#1) = 5.6130

path(mouse#n#1 , bat#n#1) = 0.2500

lesk(mouse#n#1 , bat#n#1) = 46

hso(mouse#n#1 , bat#n#1) = 4

Evaluating similarity

- Intrinsic Evaluation:
  - Correlation between algorithm and human word similarity ratings
    - Edge counting → r=0.66
    - Resnik metric → r=0.79
    - Lin metric → r=0.83
    - Human agreement (upper bound) → r=0.90
- Extrinsic (task-based, end-to-end) Evaluation:
  - Malapropism (spelling error) detection
  - WSD
  - Essay grading
  - Taking TOEFL multiple-choice vocabulary tests

    Levied is closest in meaning to:

    imposed, believed, requested, correlated

Problems with thesaurus-based meaning

- We don't have a thesaurus for every language
- Even if we do, they have problems with **recall**
  - Many words are missing
  - Most (if not all) phrases are missing
  - Some connections between senses are missing
  - Thesauri work less well for verbs, adjectives
    - Adjectives and verbs have less structured hyponymy relations

Distributional models of meaning

- Also called vector-space models of meaning
- Offer much higher recall than hand-built thesauri
  - Although they tend to have lower precision
- Zellig Harris (1954): "**oculist** and **eye-doctor** … occur in almost the same environments…. **If A and B have almost identical environments we say that they are synonyms**.

- Firth (1957): "You shall know a word by the company it keeps!"

Intuition of distributional word similarity

- Example:

  A bottle of ***tesgüino*** is on the table
  Everybody likes ***tesgüino***
  Tesgüino makes you drunk
  We make ***tesgüino*** out of corn.

- From context words humans can guess ***tesgüino*** means
  - an alcoholic beverage like **beer**
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

Term-document matrix

- Each cell: count of term *t* in a document *d*: $tf_{t,d}$:
  - Each document is a count vector in $\mathbb{N}^{|V|}$: a column below

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

Term-document matrix

- Two documents are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

The words in a term-document matrix

- Each word is a count vector in $\mathbb{N}^D$: a row below

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

The words in a term-document matrix

- Two **words** are similar if their vectors are similar

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

The Term-Context matrix

- Instead of using entire documents, use smaller contexts
  - Paragraph
  - Window of 10 words
- A word is now defined by a vector over counts of context words

Sample contexts: 20 words (Brown corpus)

- equal amount of sugar, a sliced lemon, a tablespoonful of ?????? preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first ?????? and another fruit whose taste she likened to that of

- of a recursive type well suited to programming on the ?????? computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and ?????? necessary for the study authorized in the first section of this

Sample contexts: 20 words (Brown corpus)

- equal amount of sugar, a sliced lemon, a tablespoonful of **apricot** preserve or jam, a pinch each of clove and nutmeg,
- on board for their enjoyment. Cautiously she sampled her first **pineapple** and another fruit whose taste she likened to that of

- of a recursive type well suited to programming on the **digital** computer. In finding the optimal R-stage policy from that of
- substantially affect commerce, for the purpose of gathering data and **information** necessary for the study authorized in the first section of this

Term-context matrix for word similarity

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

	aardvark	computer	data	pinch	result	sugar	…
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0	2	1	0	1	0	
information	0	1	6	0	4	0	

Should we use raw counts?

- For the term-document matrix
  - We used TF-IDF instead of raw term counts
  - TF = Term Frequency
  - IDF = Inverse Document Frequency
- For the term-context matrix
  - Positive Pointwise Mutual Information (PPMI) is common

Pointwise Mutual Information

- **Pointwise mutual information**:
  - Do events x and y co-occur more than if they were independent?

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

- **PMI between two words**: (Church & Hanks 1989)
  - Do words x and y co-occur more than if they were independent?

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

- **Positive PMI between two words** (Niwa & Nitta 1994)
  - Replace all PMI values less than 0 with zero

Computing PPMI on a term-context matrix

- Matrix F with W rows (words) and C columns (contexts)
- f_{ij} is # of times w_i occurs in context c_j

	aardvark	computer	data	pinch	result	sugar
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \qquad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w=information,c=data) = 6/19 = .32

p(w=information) = 11/19 = .58

p(c=data) = 7/19 = .37

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

p(w,context) **p(w)**

	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

	p(w,context)					p(w)
	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

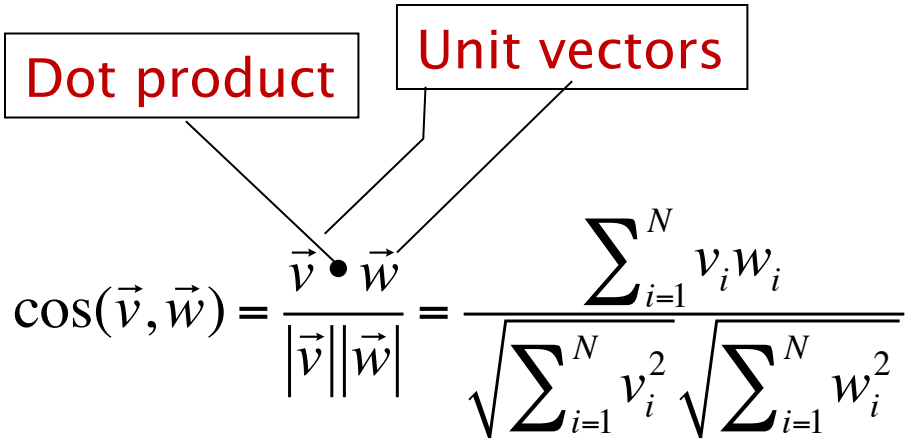
- pmi(information,data) = $\log_2$ (.32 / (.37*.58)) = .58

(.57 using full precision)

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Reminder: cosine for computing similarity

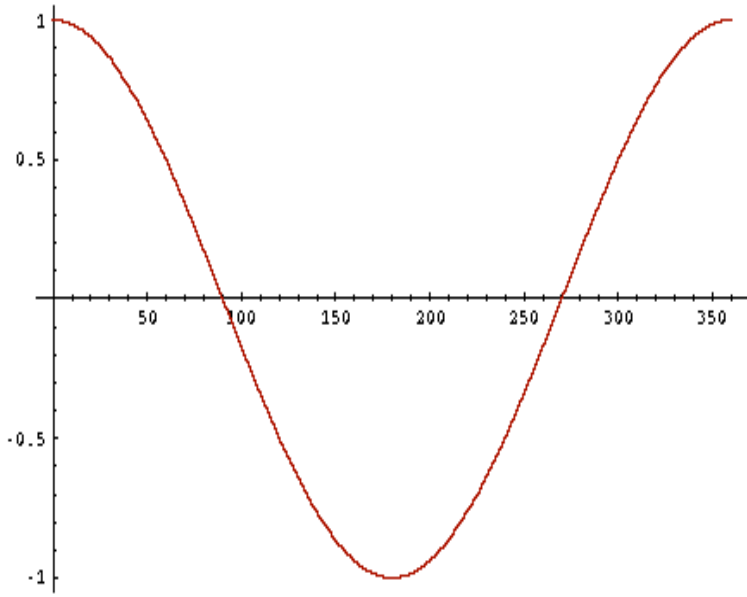


v_i is the PPMI value for word v in context i
w_i is the PPMI value for word w in context i.

Cos($\vec{v}$,$\vec{w}$) is the cosine similarity of $\vec{v}$ and $\vec{w}$

Cosine as a similarity metric

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



- Raw frequency or PPMI are non-negative, so cosine range 0-1

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

	large	data	computer
apricot	1	0	0
digital	0	1	2
information	1	6	1

Which pair of words is more similar?

cosine(apricot,information) = $\frac{1+0+0}{\sqrt{1+0+0}\sqrt{1+36+1}} = \frac{1}{\sqrt{38}} = .16$

cosine(digital,information) = $\frac{0+6+2}{\sqrt{0+1+4}\sqrt{1+36+1}} = \frac{8}{\sqrt{5}\sqrt{38}} = .58$

cosine(apricot,digital) = $\frac{0+0+0}{\sqrt{1+0+0}\sqrt{0+1+4}} = 0$

Other possible similarity measures

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

$$\text{sim}_{\text{Jaccard}}(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} \max(v_i,w_i)}$$

$$\text{sim}_{\text{Dice}}(\vec{v},\vec{w}) = \frac{2\times\sum_{i=1}^{N} \min(v_i,w_i)}{\sum_{i=1}^{N} (v_i+w_i)}$$

$$\text{sim}_{\text{JS}}(\vec{v}||\vec{w}) = D\left(\vec{v}\middle|\frac{\vec{v}+\vec{w}}{2}\right) + D\left(\vec{w}\middle|\frac{\vec{v}+\vec{w}}{2}\right)$$

Next Time

- May 5th → No Class! (Israa' & Miraaj Holiday)
- May 10th → Assignment #4
- May 12th →
  - J+M Chap 23
  - Come with questions about final!