Computational Lexical Semantics

COSI 114 – Computational Linguistics

James Pustejovsky

March 17, 2015

Brandeis University

Three Perspectives on Meaning

1. **Lexical Semantics**
   - The meanings of individual words
2. **Formal Semantics** (or Compositional Semantics or Sentential Semantics)
   - How those meanings combine to make meanings for individual sentences or utterances
3. **Discourse or Pragmatics**
   - How those meanings combine with each other and with other facts about various kinds of context to make meanings for a text or discourse
     - **Dialog or Conversation** is often lumped together with Discourse

Outline: Comp Lexical Semantics

- Intro to Lexical Semantics
  - Homonymy, Polysemy, Synonymy
  - Online resources: WordNet
- Computational Lexical Semantics
  - Word Sense Disambiguation
    - Supervised
    - Semi-supervised
  - Word Similarity
    - Thesaurus-based
    - Distributional

Preliminaries

- What's a word?
  - Definitions we've used: Types, tokens, stems, roots, inflected forms, etc...
  - Lexeme: An entry in a lexicon consisting of a pairing of a form with a single meaning representation
  - Lexicon: A collection of lexemes

Relationships between word meanings

- Homonymy
- Polysemy
- Synonymy
- Antonymy
- Hypernomy
- Hyponomy
- Meronomy

Homonymy

- Homonymy:
  - Lexemes that share a form
    - Phonological, orthographic or both
  - But have unrelated, distinct meanings
  - Clear example:
    - Bat (wooden stick-like thing) vs
    - Bat (flying scary mammal thing)
    - Or bank (financial institution) versus bank (riverside)
  - Can be homophones, homographs, or both:
    - Homophones:
      - Write and right
      - Piece and peace

Homonymy causes problems for NLP applications

- Text-to-Speech
  - Same orthographic form but different phonological form
    - bass vs bass
- Information retrieval
  - Different meanings same orthographic form
    - QUERY: bat care
- Machine Translation
- Speech recognition
  - Why?

Polysemy

- The **bank** is constructed from red brick
  I withdrew the money from the **bank**
- Are those the same sense?
- Or consider the following WSJ example
  - While some banks furnish sperm only to married women, others are less restrictive
  - Which sense of bank is this?
    - Is it distinct from (homonymous with) the river bank sense?
    - How about the savings bank sense?

Polysemy

- A single lexeme with multiple related meanings (bank the building, bank the financial institution)
- Most non-rare words have multiple meanings
  - The number of meanings is related to its frequency
  - Verbs tend more to polysemy
  - Distinguishing polysemy from homonymy isn't always easy (or necessary)

Metaphor and Metonymy

- Specific types of polysemy
- Metaphor:
  - Germany will **pull** Slovenia **out** of its economic slump.
  - I **spent** 2 hours on that homework.
- Metonymy
  - **The White House** announced yesterday.
  - **This chapter** talks about part-of-speech tagging
  - Bank (building) and bank (financial institution)

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

- ATIS examples
  - Which flights serve breakfast?
  - Does America West serve Philadelphia?
- The "zeugma" test:
  - ?Does United serve breakfast and San Jose?

Synonyms

- Word that have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H_20
- Two lexemes are synonyms if they can be successfully substituted for each other in all situations
  - If so they have the same **propositional meaning**

Synonyms

- But there are few (or no) examples of perfect synonymy.
  - Why should that be?
  - Even if many aspects of meaning are identical
  - Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
- Example:
  - Water and H_20

Some more terminology

- Lemmas and word forms
  - A **lexeme** is an abstract pairing of meaning and form
  - A **lemma** or **citation form** is the grammatical form that is used to represent a **lexeme**.
    - ***Carpet*** is the lemma for ***carpets***
    - ***Dormir*** is the lemma for ***duermes.***
  - Specific surface forms *carpets, sung, duermes* are called **word forms**
- The lemma *bank* has two **senses:**
  - Instead, a **bank** can hold the investments in a custodial account in the client's name
  - But as agriculture burgeons on the east **bank**, the river will shrink even more.
- A **sense** is a discrete representation of one aspect of the meaning of a word

Synonymy is a relation between senses rather than words

- Consider the words *big* and *large*
- Are they synonyms?
  - How **big** is that plane?
  - Would I be flying on a **large** or small plane?
- How about here:
  - Miss Nelson, for instance, became a kind of **big** sister to Benjamin.
  - ?Miss Nelson, for instance, became a kind of **large** sister to Benjamin.
- Why?
  - *big* has a sense that means being older, or grown up
  - *large* lacks this sense

Antonyms

- Senses that are opposites with respect to one feature of their meaning
- Otherwise, they are very similar!
  - dark / light
  - short / long
  - hot / cold
  - up / down
  - in / out
- More formally: antonyms can
  - define a binary opposition or at opposite ends of a scale (*long/short, fast/slow*)
  - Be **reversives**: *rise/fall, up/down*

Hyponymy

- One sense is a **hyponym** of another if the first sense is more specific, denoting a subclass of the other
  - *car* is a hyponym of *vehicle*
  - *dog* is a hyponym of *animal*
  - *mango* is a hyponym of *fruit*
- Conversely
  - *vehicle* is a hypernym/superordinate of *car*
  - *animal* is a hypernym of *dog*
  - *fruit* is a hypernym of *mango*

superordinate	vehicle	fruit	furniture	mammal
hyponym	car	mango	chair	dog

Hypernymy 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)

II. WordNet

- A hierarchically organized lexical database
- On-line thesaurus + aspects of a dictionary
  - Versions for other languages are under development

Category	Unique Forms
Noun	**117,097**
Verb	**11,488**
Adjective	**22,141**
Adverb	**4,601**

Format of Wordnet Entries

The noun "bass" has 8 senses in WordNet.

1. bass[1] - (the lowest part of the musical range)
2. bass[2], bass part[1] - (the lowest part in polyphonic music)
3. bass[3], basso[1] - (an adult male singer with the lowest voice)
4. sea bass[1], bass[4] - (the lean flesh of a saltwater fish of the family Serranidae)
5. freshwater bass[1], bass[5] - (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
6. bass[6], bass voice[1], basso[2] - (the lowest adult male singing voice)
7. bass[7] - (the member with the lowest range of a family of musical instruments)
8. bass[8] - (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

The adjective "bass" has 1 sense in WordNet.

1. bass[1], deep[6] - (having or denoting a low vocal or instrumental range)
   "a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"

WordNet Noun Relations

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

WordNet Verb Relations

Relation	Definition	Example
Hypernym	From events to superordinate events	$fly^9 \rightarrow travel^5$
Troponym	From a verb (event) to a specific manner elaboration of that verb	$walk^1 \rightarrow stroll^1$
Entails	From verbs (events) to the verbs (events) they entail	$snore^1 \rightarrow sleep^1$
Antonym	Opposites	$increase^1 \Longleftrightarrow decrease^1$

WordNet Hierarchies

```
Sense 3
bass, basso --
(an adult male singer with the lowest voice)
=> singer, vocalist, vocalizer, vocaliser
   => musician, instrumentalist, player
      => performer, performing artist
         => entertainer
            => person, individual, someone...
               => organism, being
                  => living thing, animate thing,
                     => whole, unit
                        => object, physical object
                           => physical entity
                              => entity
               => causal agent, cause, causal agency
                  => physical entity
                     => entity


Sense 7
bass --
(the member with the lowest range of a family of
musical instruments)
=> musical instrument, instrument
   => device
      => instrumentality, instrumentation
         => artifact, artefact
            => whole, unit
               => object, physical object
                  => physical entity
                     => entity
```

How is "sense" defined in WordNet?

- The set of near-synonyms for a WordNet sense is called a **synset (synonym set)**; it's their version of a sense or a concept
- Example: chump as a noun to mean
  - 'a person who is gullible and easy to take advantage of'

{chump[1], fool[2], gull[1], mark[9], patsy[1], fall guy[1], sucker[1], soft touch[1], mug[2]}

- Each of these senses share this same gloss
- Thus for WordNet, the meaning of this sense of chump *is* this list.

Word Sense Disambiguation (WSD)

- Given
  - a word in context,
  - A fixed inventory of potential word sense
- decide which sense of the word this is.
  - English-to-Spanish MT
    - Inventory is set of Spanish translations
  - Speech Synthesis
    - Inventory is homographs with different pronunciations like *bass* and *bow*
  - Automatic indexing of medical articles
    - MeSH (Medical Subject Headings) thesaurus entries

Two variants of WSD task

- Lexical Sample task
  - Small pre-selected set of target words
  - And inventory of senses for each word
  - We'll use **supervised machine learning**
- All-words task
  - Every word in an entire text
  - A lexicon with senses for each word
  - Sort of like part-of-speech tagging
    - Except each lemma has its own tagset

Supervised Machine Learning Approaches

- Supervised machine learning approach:
  - a training corpus of words tagged in context with their sense
  - used to train a classifier that can tag words in new text
  - Just as we saw for part-of-speech tagging, statistical MT.
- Summary of what we need:
  - the **tag set** ("sense inventory")
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**

WordNet Bass

The noun ``bass'' has 8 senses in WordNet

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus Micropterus)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass -(nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Inventory of sense tags for *bass*

WordNet Sense	Spanish Translation	Roget Category	Target Word in Context
bass^4	lubina	FISH/INSECT	. . . fish as Pacific salmon and striped **bass** and. . .
bass^4	lubina	FISH/INSECT	. . . produce filets of smoked **bass** or sturgeon. . .
bass^7	bajo	MUSIC	. . . exciting jazz **bass** player since Ray Brown. . .
bass^7	bajo	MUSIC	. . . play **bass** because he doesn't have to solo. . .

Supervised WSD 1: WSD Tags

- What's a tag?
  - A dictionary sense?
- For example, for WordNet an instance of "bass" in a text has 8 possible tags or labels (bass1 through bass8).

Supervised WSD 2: Get a corpus

- Lexical sample task:
  - *Line-hard-serve* corpus - 4000 examples of each
  - *Interest* corpus - 2369 sense-tagged examples
- All words:
  - **Semantic concordance**: a corpus in which each open-class word is labeled with a sense from a specific dictionary/thesaurus.
    - SemCor: 234,000 words from Brown Corpus, manually tagged with WordNet senses
    - SENSEVAL-3 competition corpora - 2081 tagged word tokens

Supervised WSD 3: Extract feature vectors

- Weaver (1955)
  - If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. […]
  - But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. […]
  - The practical question is : ``What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?"

Feature vectors

- A simple representation for each observation (each instance of a target word)
  - Vectors of sets of feature/value pairs
    - I.e. files of comma-separated values
  - These vectors should represent the window of words around the target

Two kinds of features in the vectors

- **Collocational** features and **bag-of-words** features
  - **Collocational**
    - Features about words at **specific** positions near target word
      - Often limited to just word identity and POS
  - **Bag-of-words**
    - Features about words that occur anywhere in the window (regardless of position)
      - Typically limited to frequency counts

Examples

- Example text (WSJ)
  - An electric guitar and **bass** player stand off to one side not really part of the scene, just as a sort of nod to gringo expectations perhaps
  - Assume a window of +/- 2 from the target

Examples

- Example text
  - An electric guitar and **bass** player stand off to one side not really part of the scene, just as a sort of nod to gringo expectations perhaps
  - Assume a window of +/- 2 from the target

Collocational

- Position-specific information about the words in the window
- guitar and bass player stand
  - [guitar, NN, and, CC, player, NN, stand, VB]
  - $Word_{n-2,}$ $POS_{n-2,}$ $word_{n-1,}$ $POS_{n-1,}$ $Word_{n+1}$ POS_{n+1}…
  - In other words, a vector consisting of
  - [position n word, position n part-of-speech…]

Bag-of-words

- Information about the words that occur within the window.
- First derive a set of terms to place in the vector.
- Then note how often each of those terms occurs in a given window.

Co-Occurrence Example

- Assume we've settled on a possible vocabulary of 12 words that includes guitar and player but not and and stand

- guitar and bass player stand
  - [0,0,0,1,0,0,0,0,0,1,0,0]
  - Which are the counts of words predefined as e.g.,
  - [fish,fishing,viol, guitar, double,cello…

Classifiers

- Once we cast the WSD problem as a classification problem, then all sorts of techniques are possible
  - Naïve Bayes (the easiest thing to try first)
  - Decision lists
  - Decision trees
  - Neural nets
  - Support vector machines
  - Nearest neighbor methods…

Classifiers

- The choice of technique, in part, depends on the set of features that have been used
  - Some techniques work better/worse with features with numerical values
  - Some techniques work better/worse with features that have large numbers of possible values
    - For example, the feature **the word to the left** has a fairly large number of possible values

Naïve Bayes

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \; P(s \mid \vec{f})$$

- Rewriting with Bayes

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \; \frac{P(\vec{f} \mid s)\, p(s)}{p(\vec{f})}$$

- Removing denominator

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} \; P(\vec{f} \mid s) P(s)$$

- assuming independence of the features:

$$P(\vec{f} \mid s) \approx \prod_{j=1}^{n} P(f_j \mid s)$$

- Final:

$$\hat{s} \approx \underset{s \in S}{\operatorname{argmax}} \; P(s) \prod_{j=1}^{n} P(f_j \mid s)$$

Naïve Bayes

- P(s) … just the prior of that sense.
  - Just as with part of speech tagging, not all senses will occur with equal frequency
  - P(si) = count(si,wj)/count(wj)
- P(fj|s)… conditional probability of some particular feature/value combination given a particular sense
  - P(fj|s) = count(fj,s)/count(s)
- You can get both of these from a tagged corpus with the features encoded

Naïve Bayes Test

- On a corpus of examples of uses of the word line, naïve Bayes achieved about 73% correct

- Good?

Decision Lists: another popular method

- A case statement….

Rule		Sense
fish within window	⇒	**bass**^{1}
striped bass	⇒	**bass**^{1}
guitar within window	⇒	**bass**^{2}
bass player	⇒	**bass**^{2}
piano within window	⇒	**bass**^{2}
tenor within window	⇒	**bass**^{2}
sea bass	⇒	**bass**^{1}
play/V *bass*	⇒	**bass**^{2}
river within window	⇒	**bass**^{1}
violin within window	⇒	**bass**^{2}
salmon within window	⇒	**bass**^{1}
on bass	⇒	**bass**^{2}
bass are	⇒	**bass**^{1}

Learning Decision Lists

- Restrict the lists to rules that test a single feature (1-decisionlist rules)
- Evaluate each possible test and rank them based on how well they work.
- Glue the top-N tests together and call that your decision list.

Yarowsky

- On a binary (homonymy) distinction used the following metric to rank the tests

$$\frac{P(\text{Sense}_1 \mid Feature)}{P(\text{Sense}_2 \mid Feature)}$$

- Ratio tells us how discriminating this feature is
- Order the tests by the log-likelihood ratio
- This gives about 95% on this test…

WSD Evaluations and baselines

- *In vivo* versus *in vitro* evaluation
- In vitro evaluation is most common now
  - Exact match **accuracy**
    - % of words tagged identically with manual sense tags
  - Usually evaluate using held-out data from same labeled corpus
    - Problems?
    - Why do we do it anyhow?
- Baselines
  - Most frequent sense
  - The Lesk algorithm

Most Frequent Sense

- Wordnet senses are ordered in frequency order
- So "most frequent sense" in wordnet = "take the first sense"

Freq	Synset	Gloss
338	plant[1], works, industrial plant	buildings for carrying on industrial labor
207	plant[2], flora, plant life	a living organism lacking the power of locomotion
2	plant[3]	something planted secretly for discovery by another
0	plant[4]	an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience

Ceiling

- Human inter-annotator agreement
  - Compare annotations of two humans
  - On same data
  - Given same tagging guidelines
- Human agreements on all-words corpora with Wordnet style senses
  - 75%-80%

WSD: Dictionary/Thesaurus methods

- The Lesk Algorithm
- Selectional Restrictions and Selectional Preferences

Simplified Lesk

- Count the overlap between the context and the dictionary definition
  - Sentence: "The bank can guarantee deposits will eventually cover future tuition costs because it invest in adjustable-rate mortgage securities

bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

WSD: Dictionary/Thesaurus methods

- The Lesk Algorithm
  - Compare words in the neighborhood of an ambiguous word with words in the definitions of those words
- Selectional Restrictions and Selectional Preferences

Simplified Lesk

The **bank** can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

given the following two WordNet senses:

bank[1]	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
bank[2]	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", "he sat on the bank of the river and watched the currents"

Original Lesk: pine cone

pine	1	kinds of evergreen tree with needle-shaped leaves
	2	waste away through sorrow or illness
cone	1	solid body which narrows to a point
	2	something of this shape whether solid or hollow
	3	fruit of certain evergreen trees

Corpus Lesk

- Add corpus examples to glosses and examples
- The best performing variant

Bootstrapping

- What if you don't have enough data to train a system…
- Bootstrap
  - Pick a word that you as an analyst think will co-occur with your target word in particular sense
  - Grep through your corpus for your target word and the hypothesized word
  - Assume that the target tag is the right one

Bootstrapping

- For bass
  - Assume play occurs with the music sense and fish occurs with the fish sense

Sentences extracting using "fish" and "play"

We need more good teachers – right now, there are only a half a dozen who can **play** the free **bass** with ease.
An electric guitar and **bass play**er stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations perhaps.
When the New Jersey Jazz Society, in a fund-raiser for the American Jazz Hall of Fame, honors this historic night next Saturday, Harry Goodman, Mr. Goodman's brother and **bass play**er at the original concert, will be in the audience with other family members.
The researchers said the worms spend part of their life cycle in such **fish** as Pacific salmon and striped **bass** and Pacific rockfish or snapper.
And it all started when **fish**ermen decided the striped **bass** in Lake Mead were too skinny.
Though still a far cry from the lake's record 52-pound **bass** of a decade ago, "you could fillet these **fish** again, and that made people very, very happy," Mr. Paulson says.

Where do the seeds come from?

1) Hand labeling
2) "One sense per discourse":
   - The sense of a word is highly consistent within a document - Yarowsky (1995)
   - True for topic dependent words
   - Not so true for other POS like adjectives and verbs, e.g. make, take
   - Krovetz (1998) "More than one sense per discourse" argues it isn't true at all once you move to fine-grained senses
3) One sense per collocation:
   - A word reoccurring in collocation with the same word will almost surely have the same sense.

Slide adapted from Chris Manning

Problems

- Given these general ML approaches, how many classifiers do I need to perform WSD robustly
  - One for each ambiguous word in the language
- How do you decide what set of tags/ labels/senses to use for a given word?
  - Depends on the application

WordNet Bass

- Tagging with this set of senses is an impossibly hard task that's probably overkill for any realistic application

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus Micropterus)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass -(nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Senseval History

- ACL-SIGLEX workshop (1997)
  - Yarowsky and Resnik paper
- SENSEVAL-I (1998)
  - Lexical Sample for English, French, and Italian
- SENSEVAL-II (Toulouse, 2001)
  - Lexical Sample and All Words
  - Organization: Kilkgarriff (Brighton)
- SENSEVAL-III (2004)
- SENSEVAL-IV → SEMEVAL (2007)
- → SEMEVAL 2015

WSD Performance

- Varies widely depending on how difficult the disambiguation task is
- Accuracies of over 90% are commonly reported on some of the classic, often fairly easy, WSD tasks (pike, star, interest)
- Senseval brought careful evaluation of difficult WSD (many senses, different POS)
- Senseval 1: more fine grained senses, wider range of types:
  - Overall: about 75% accuracy
  - Nouns: about 80% accuracy
  - Verbs: about 70% accuracy

Word Similarity

- Synonymy is a binary relation
  - Two words are either synonymous or not
- We want a looser metric
  - Word similarity or
  - Word distance
- Two words are more similar
  - If they share more features of meaning
- Actually these are really relations between **senses**:
  - Instead of saying "bank is like fund"
  - We say
    - Bank1 is similar to fund3
    - Bank2 is similar to slope5
- We'll compute them over both words and senses

Why word similarity

- Information retrieval
- Question answering
- Machine translation
- Natural language generation
- Language modeling
- Automatic essay grading

Two classes of algorithms

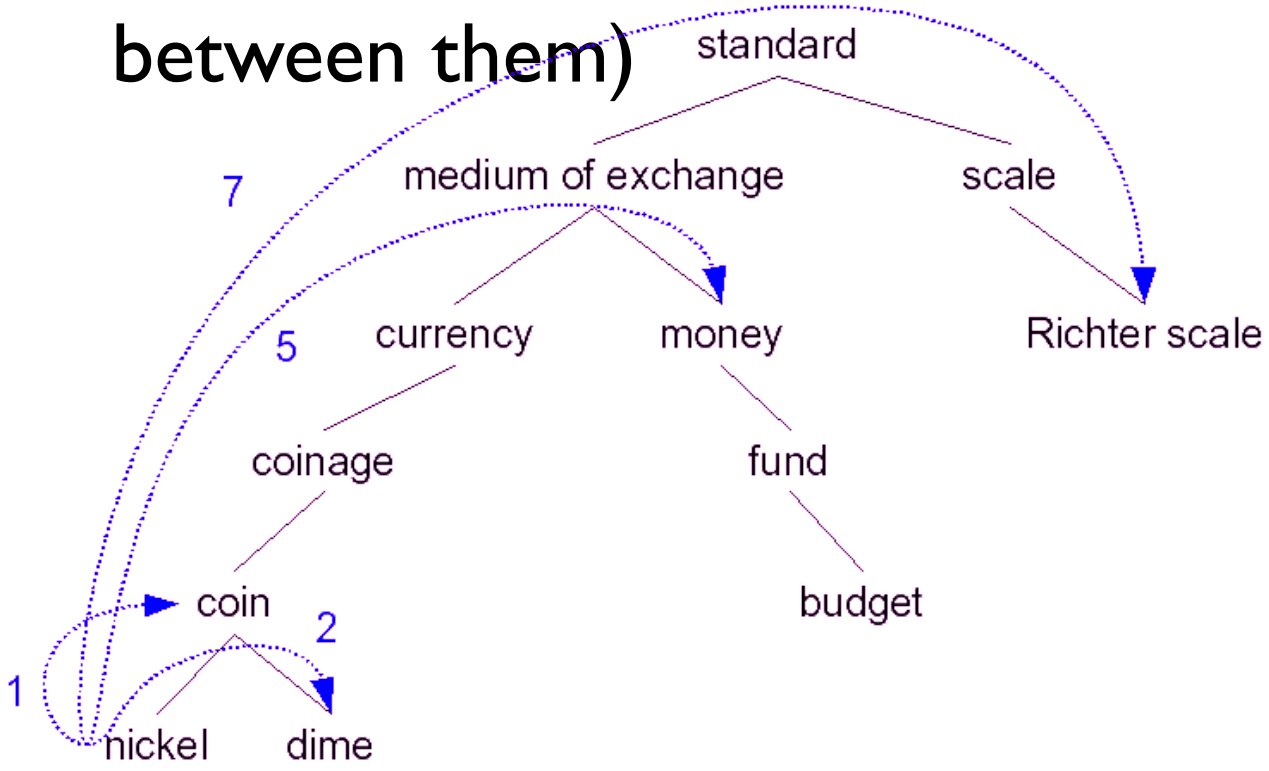
- Thesaurus-based algorithms
  - Based on whether words are "nearby" in Wordnet or MeSH
- Distributional algorithms
  - By comparing words based on their distributional context

Thesaurus-based word similarity

- We could use anything in the thesaurus
  - Meronymy
  - Glosses
  - Example sentences
- In practice
  - By "thesaurus-based" we just mean
    - Using the is-a/subsumption/hypernym hierarchy
- Word similarity versus word relatedness
  - Similar words are near-synonyms
  - Related could be related any way
    - Car, gasoline: related, not similar
    - Car, bicycle: similar

Path based similarity

- Two words are similar if nearby in thesaurus hierarchy (i.e. short path between them)



Refinements to path-based similarity

- pathlen(c1,c2) = number of edges in the shortest path in the thesaurus graph between the sense nodes c1 and c2
- simpath(c1,c2) = -log pathlen(c1,c2)
- wordsim(w1,w2) =
  - $\max_{c1 \in senses(w1), c2 \in senses(w2)}$ sim(c1,c2)

Problem with basic path-based similarity

- Assumes each link represents a uniform distance
- *Nickel* to *money* seem closer than *nickel* to *standard*
- Instead:
  - Want a metric which lets us
  - Represent the cost of each edge independently

Information content similarity metrics

- 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
  - P(root)=1
  - The lower a node in the hierarchy, the lower its probability

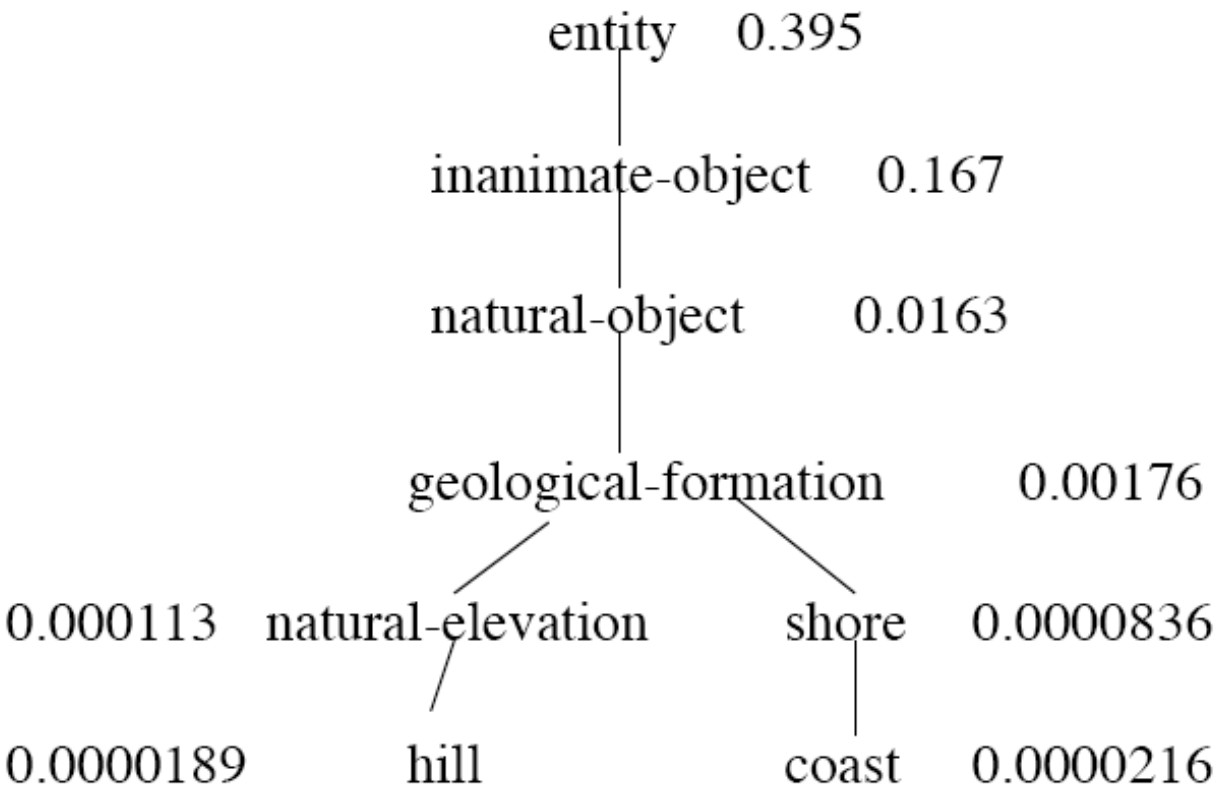
Information content similarity

- Train by counting in a corpus
  - 1 instance of "dime" could count toward frequency of *coin*, *currency*, *standard*, etc
- More formally:

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

Information content similarity

- WordNet hierarchy augmented with probabilities P(C)



Information content: definitions

- Information content:
  - IC(c)=-logP(c)
- Lowest common subsumer
  - LCS(c1,c2) = the lowest common subsumer
    - I.e. the lowest node in the hierarchy that subsumes (is a hypernym of) both c1 and c2
- We are now ready to see how to use information content IC as a similarity metric

Resnik method

- 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 the common information as:
  - The info content of the lowest common subsumer of the two nodes
  - $sim_{resnik}(c1,c2) = -\log P(LCS(c1,c2))$

Dekang Lin method

- Similarity between A and B needs to do more than measure common information
- The more differences between A and B, the less similar they are:
  - Commonality: the more info A and B have in common, the more similar they are
  - Difference: the more differences between the info in A and B, the less similar
- Commonality: IC(Common(A,B))
- Difference: IC(description(A,B)-IC(common(A,B))

Dekang Lin method

- 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

$$\text{simLin}(A,B) = \frac{\log P(\text{common}(A,B))}{\log P(\text{description}(A,B))}$$

- Lin furthermore shows (modifying Resnik) that info in common is twice the info content of the LCS

Lin similarity function

- $\text{SimLin}(c1,c2) = \frac{2 \times \log P(\text{LCS}(c1,c2))}{\log P(c1) + \log P(c2)}$
- $\text{SimLin}(\text{hill},\text{coast}) = \frac{2 \times \log P(\text{geological-formation}))}{\log P(\text{hill}) + \log P(\text{coast})}$
- $= .59$

Extended Lesk

- 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 occurs in both glosses
  - Add a score of n^2
  - *Paper* and *specially prepared* for 1 + 4 = 5…

Summary: thesaurus-based similarity

$$\text{sim}_{\text{path}}(c_1, c_2) = -\log \text{pathlen}(c_1, c_2)$$

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

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

$$\text{sim}_{\text{jc}}(c_1, c_2) = \frac{1}{2 \times \log P(\text{LCS}(c_1, c_2)) - (\log P(c_1) + \log P(c_2))}$$

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

Problems with thesaurus-based methods

- We don't have a thesaurus for every language
- Even if we do, many words are missing
- They rely on hyponym info:
  - Strong for nouns, but lacking for adjectives and even verbs
- Alternative
  - Distributional methods for word similarity

Distributional methods for word similarity

- Firth (1957): "You shall know a word by the company it keeps!"
- Nida example noted by Lin:
  - A bottle of ***tezgüino*** is on the table
  - Everybody likes ***tezgüino***
  - ***Tezgüino*** makes you drunk
  - We make ***tezgüino*** out of corn.
- Intuition:
  - just from these contexts a human could guess meaning of tezguino
  - So we should look at the surrounding contexts, see what other words have similar context.

Context vector

- Consider a target word *w*
- Suppose we had one binary feature f_i for each of the *N* words in the lexicon v_i
- Which means "word v_i occurs in the neighborhood of *w*"
- w=(f1,f2,f3,…,fN)
- If w=tezguino, v1 = bottle, v2 = drunk, v3 = matrix:
- w = (1,1,0,…)

Intuition

- Define two words by these sparse features vectors
- Apply a vector distance metric

	arts	boil	data	function	large	sugar	summarized	water
apricot	0	1	0	0	1	1	0	1
pineapple	0	1	0	0	1	1	0	1
digital	0	0	1	1	1	0	1	0
information	0	0	1	1	1	0	1	0

Distributional similarity

- So we just need to specify 3 things
  1. How the co-occurrence terms are defined
  2. How terms are weighted
     - (frequency? Logs? Mutual information?)
  3. What vector distance metric should we use?
     - Cosine? Euclidean distance?

Defining co-occurrence vectors

- Just as for WSD
- We could have windows
  - Bag-of-words
  - We generally remove **stopwords**
- But the vectors are still very sparse
- So instead of using ALL the words in the neighborhood
- How about just the words occurring in particular relations

Defining co-occurrence vectors

- Zellig Harris (1968)
  - The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entitites relative to other entities

I discovered dried tangerines:

discover (subject I)	I (subj-of discover)
tangerine (obj-of discover)	tangerine (adj-mod dried)
dried (adj-mod-of tangerine)	

Co-occurrence vectors based on dependencies

- For the word "cell": vector of NxR features
  - R is the number of dependency relations

	subj-of, absorb	subj-of, adapt	subj-of, behave	...	pobj-of, inside	pobj-of, into	...	nmod-of, abnormality	nmod-of, anemia	nmod-of, architecture	...	obj-of, attack	obj-of, call	obj-of, come from	obj-of, decorate	...	nmod, bacteria	nmod, body	nmod, bone marrow
cell	1	1	1		16	30		3	8	1		6	11	3	2		3	2	2

2. Weighting the counts
("Measures of association with context")

- We have been using the frequency of some feature as its weight or value
- But we could use any function of this frequency
- Let's consider one feature
- f=(r,w') = (obj-of,*attack*)
- P(f|w)=count(f,w)/count(w)

- $Assoc_{prob}(w,f)=p(f|w)$

Intuition: why not frequency

Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

- "drink it" is more common than "drink wine"
- But "wine" is a better "drinkable" thing than "it"
- Idea:
  - We need to control for change (expected frequency)
  - We do this by normalizing by the expected frequency we would get assuming independence

Weighting: Mutual Information

- **Mutual information:** between 2 random variables X and Y

$$I(X,Y) = \sum_x \sum_y P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

Essential Information Theory

- Developed by Shannon in the 40s
  - Maximizing the amount of information that can be transmitted over an imperfect communication channel
  - Data compression (entropy)
  - Transmission rate (channel capacity)
- In Computational Linguistics
  - Underlies perplexity: measure of how well a particular grammar matches a particular language

Entropy

- X: discrete RV, p(X)
- Entropy (or self-information)

$$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Entropy measures the amount of information in a RV; it's the average length of the message needed to transmit an outcome of that variable using the optimal code
- Lower bound on the number of bits needed to encode a decision or piece of information

Example

- There are 8 horses in a race. You want to send a bet to your bookie. How many bits do you need?
  - Simplest is 3: 000,001,011, …
  - If we bet all day, the average number of bits is 3
- But assume this distribution of priors on the horse

Horse 1	$\frac{1}{2}$	Horse 5	$\frac{1}{64}$
Horse 2	$\frac{1}{4}$	Horse 6	$\frac{1}{64}$
Horse 3	$\frac{1}{8}$	Horse 7	$\frac{1}{64}$
Horse 4	$\frac{1}{16}$	Horse 8	$\frac{1}{64}$

Example (cont)

- Entropy of the random variable X that ranges over the horses

$$H(p) = H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

= -1/2log1/2 - 1/4log1/4 …

= 2 bits

- For example, we could encode the most likely horse with the code 0, the next with 10, the next with 110, 1110, …
- One bit is the most frequent. If we bet all day, the average would be 2 bits.

Joint Entropy

- The joint entropy of 2 RV X,Y is the amount of the information needed on average to specify both their values

$$H(X,Y) = -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(X,Y)$$

Conditional Entropy

- The conditional entropy of a RV Y given another X, expresses how much extra information one still needs to supply on average to communicate Y given that the other party knows X

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X=x)$$

$$= -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log p(y|x)$$

$$= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log p(y|x) \; = -E(\log p(Y|X))$$

Chain Rule

$$H(X,Y) = H(X) + H(Y \mid X)$$

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + \ldots. + H(X_n \mid X_1, \ldots X_{n-1})$$

Mutual Information

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$$

$$H(X) - H(X|Y) = H(Y) - H(Y|X) = I(X,Y)$$

- I(X,Y) is the mutual information between X and Y. It is the reduction of uncertainty of one RV due to knowing about the other, or the amount of information one RV contains about the other

Mutual Information (cont)

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- I is 0 only when X,Y are independent: H(X|Y)=H(X)

- H(X)=H(X)-H(X|X)=I(X,X) Entropy is the self-information

Entropy and Linguistics

- Entropy is measure of uncertainty. The more we know about something the lower the entropy.
- If a language model captures more of the structure of the language, then the entropy should be lower.
- We can use entropy as a measure of the quality of our models

Weighting: Mutual Information

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- PMI between a target word *w* and a feature *f* :

$$\text{assoc}_{\text{PMI}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

Mutual information intuition

- Objects of the verb *drink*

Object	Count	PMI assoc	Object	Count	PMI assoc
bunch beer	2	12.34	wine	2	9.34
tea	2	11.75	water	7	7.65
Pepsi	2	11.75	anything	3	5.15
champagne	4	11.75	much	3	5.15
liquid	2	10.53	it	3	1.25
beer	5	10.20	<SOME AMOUNT>	2	1.22

Lin is a variant on PMI

- **Pointwise mutual information**: measure of how often two events x and y occur, compared with what we would expect if they were independent:

$$I(x,y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

- PMI between a target word *w* and a feature *f* :

$$\text{assoc}_{\text{PMI}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(f)}$$

- Lin measure: breaks down expected value for P(f) differently:

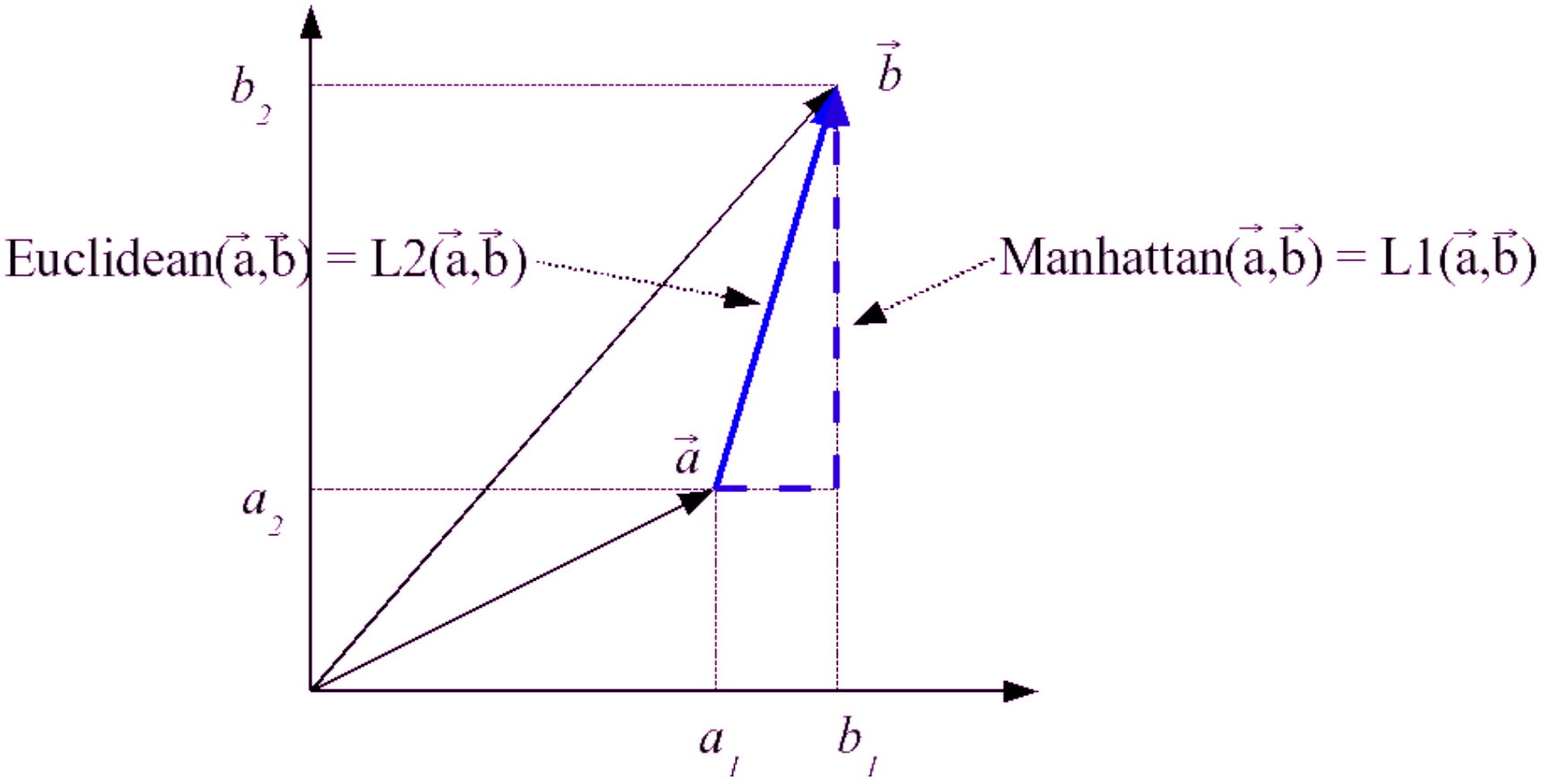
$$\text{assoc}_{\text{Lin}}(w,f) = \log_2 \frac{P(w,f)}{P(w)P(r|w)P(w'|w)}$$

Summary: weightings

- See Manning and Schuetze (1999) for more

$$\begin{aligned}
\text{assoc}_{\text{prob}}(w,f) &= P(f|w) \\
\text{assoc}_{\text{PMI}}(w,f) &= \log_2 \frac{P(w,f)}{P(w)P(f)} \\
\text{assoc}_{\text{Lin}}(w,f) &= \log_2 \frac{P(w,f)}{P(w)P(r|w)P(w'|w)} \\
\text{assoc}_{\text{t-test}}(w,f) &= \frac{P(w,f)-P(w)P(f)}{\sqrt{P(f)P(w)}}
\end{aligned}$$

3. Defining similarity between vectors



Summary of similarity measures

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

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

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

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

Evaluating similarity

- Intrinsic Evaluation:
  - Correlation coefficient between algorithm scores
    - And word similarity ratings from humans
- Extrinsic (task-based, end-to-end) Evaluation:
  - Malapropism (spelling error) detection
  - WSD
  - Essay grading
  - Taking TOEFL multiple-choice vocabulary tests
  - Language modeling in some application

An example of detected plagiarism

MAINFRAMES

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

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

MAINFRAMES

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

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

Detecting hyponymy and other relations

- Could we discover new hyponyms, and add them to a taxonomy under the appropriate hypernym?
- Why is this important?
  - "**insulin**" and "**progesterone** are in WN 2.1, but "**leptin**" and "**pregnenolone**" are not.
  - "**combustibility**" and "**navigability**", but not "**affordability**", "**reusability**", or "**extensibility**".
  - "**HTML**" and "**SGML**", but not "**XML**" or "**XHTML**".
  - "**Google**" and "**Yahoo**", but not "**Microsoft**" or "**IBM**".
- This **unknown word problem** occurs throughout NLP

Hearst Approach

- Agar is a substance prepared from a mixture of red algae, such as Gelidium, for laboratory or industrial use.
- What does Gelidium mean? How do you know?

$$NP_0 \textit{ such as } NP_1\{,NP_2\ldots,(and|or)NP_i\}, i \geq 1$$

implies the following semantics

$$\forall NP_i, i \geq 1, \text{hyponym}(NP_i, NP_0)$$

allowing us to infer

$$\text{hyponym}(\text{Gelidium}, \text{red algae})$$

Hearst's hand-built patterns

$NP\{,NP\}*\{,\}$ (and|or) other NP_H	. . . temples, treasuries, and other important civic buildings.
NP_H such as $\{NP,\}$* (or|and) NP	red algae such as Gelidium
such NP_H as $\{NP,\}$* (or|and) NP	works by such authors as Herrick, Goldsmith, and Shakespeare
NP_H $\{,\}$ including $\{NP,\}$* (or|and) NP	All common-law countries, including Canada and England
NP_H $\{,\}$ especially $\{NP,\}$* (or|and) NP	. . . most European countries, especially France, England, and Spain

Distributional Semantics: Word Association and Similarity

Word Association Measures

- Goal: measure the statistical strength of word (term) co-occurrence in corpus
  - How strongly two words are associated?
  - Also called 1st-order similarity
- Based on contingency table (next slide)

	x	~x
y	a	b
~y	c	d

Term Co-occurrence Representation

- Surface form vs. *lemma*
  - Lemma: base form of a word, as in dictionary
  - Produced as output by *lemmatizer*
    - Input: surface form, POS
    - Example tool: http://nltk.org/api/nltk.stem.html
- Co-occurrence in a narrow/wide window
- Co-occurrence in a syntactic relationships
  - E.g. Subj-verb
  - Typically based on dependency structure

Point-wise Mutual Information

- Simple co-occurrence measures:
  - For two co-occurring words *x,y*: *freq(x,y), log(freq(x,y)+1)*
  - Do not normalize for word frequency

- PMI normalizes for word frequency:

$$PMI(x,y) = \log \frac{P(x,y)}{P(x)P(y)} = \log \frac{P(x|y)}{P(x)} = \log \frac{P(y|x)}{P(y)}$$

- **Estimation** – according to the ***space of co-occurrences!***
  - Out of all counted co-occurrences, what is the probability of *x-y, x-*, *-y ?* (more detail in PMI for 2nd-order similarity)
- Disadvantage: the PMI value is inflated for low *freq(x,y)*

- *Chapter about PMI: http://nlp.stanford.edu/fsnlp/promo/colloc.pdf, 5.4 סעיף*

Dice association measure

- Dice formula: $$\frac{2 * count(c,t)}{count(c) + count(t)}$$
  - Associated words for: baseball
    - Co-occurring unigrams: pitcher, league, bat, Yankees.
    - Co-occurring bigrams: baseball player/team, major league, Jackie Robinson, Ted Williams.
  - Capturing topical co-occurrence

The distributional hypothesis (Firth, Harris)

- "You shall know a word by the company it keeps" (Firth)
- Similar words tend to occur in similar contexts

- What does "similar" mean?
  - We'll get back to this later
  - Partly still an open question

The distributional hypothesis in real life

McDonald & Ramscar 2001

He filled the wampimuk, passed it around and we all drunk some

We found a little, hairy wampimuk sleeping behind the tree

What word can appear in the context of all these words?

Word 1: drown, bathroom, shower, fill, fall, lie, electrocute, toilet, whirlpool, iron, gin

Word 2: eat, fall, pick, slice, peel, tree, throw, fruit, pie, bite, crab, grate

Word 3: advocate, overthrow, establish, citizen, ideal, representative, dictatorship, campaign, bastion, freedom

Word 4: spend, enjoy, remember, last, pass, end, die, happen, brighten, relive

What word can appear in the context of all these words?

bathtub

Word 1: drown, bathroom, shower, fill, fall, lie, electrocute, toilet, whirlpool, iron, gin

apple

Word 2: eat, fall, pick, slice, peel, tree, throw, fruit, pie, bite, crab, grate

democracy

Word 3: advocate, overthrow, establish, citizen, ideal, representative, dictatorship, campaign, bastion, freedom

day

Word 4: spend, enjoy, remember, last, pass, end, die, happen, brighten, relive

What can you say about word number 5?
Distributional Similarity (2nd-order)

bathtub

Word 1: drown, bathroom, shower, fill, fall, lie, electrocute, toilet, whirlpool, iron, gin

apple

Word 2: eat, fall, ripe, slice, peel, tree, throw, fruit, pie, bite, crab, grate

day

Word 4: spend, enjoy, remember, last, pass, end, die, happen, brighten, relive

democracy

Word 3: advocate, overthrow, establish, citizen, ideal, representative, dictatorship, campaign, bastion, freedom

Word 5: eat, paint, peel, apple, fruit, juice, lemon, blue, grow

What can you say about word number 5?
Distributional Similarity (2nd-order)

bathtub

Word 1: drown, bathroom, shower, fill, fall, lie, electrocute, toilet, whirlpool, iron, gin

apple

Word 2: eat, fall, ripe, slice, peel, tree, throw, fruit, pie, bite, crab, grate

day

Word 4: spend, enjoy, remember, last, pass, end, die, happen, brighten, relive

democracy

Word 3: advocate, overthrow, establish, citizen, ideal, representative, dictatorship, campaign, bastion, freedom

orange

Word 5: eat, paint, peel, apple, fruit, juice, lemon, blue, grow

Counting context words

- They **picked up red** apples **that had fallen** to the ground
- **Eating** apples **is healthy**
- She **ate a red** apple
- **Pick an** apple.

Word count, 3-word context window,

a	be	eat	fall	have	healthy	pick	red	that	up
2	1	2	1	1	1	2	2	1	1

Distributional semantics

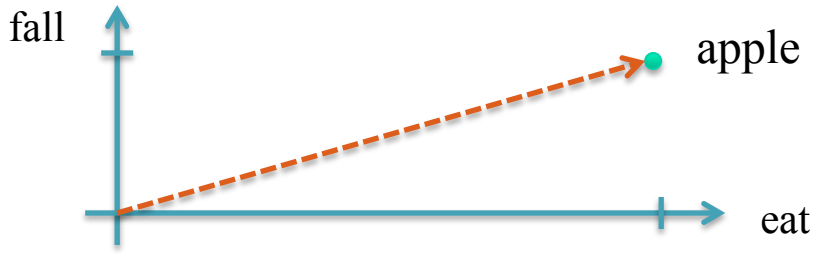
- Comparing two words:
  - Look at all context words for word1
  - Look at all context words for word2
  - How similar are those two context collections in their entirety?
- Compare distributional representations of two words

How can we compare two context collections in their entirety?

Count how often “apple” occurs close to other words in a large text collection (corpus):

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
794	244	47	221	208	160	145	156	109	104	88

Interpret counts as coordinates:



Every context word becomes a dimension.

How can we compare two context collections in their entirety?

Count how often “apple” occurs close to other words in a large text collection (corpus):

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
794	244	47	221	208	160	145	156	109	104	88

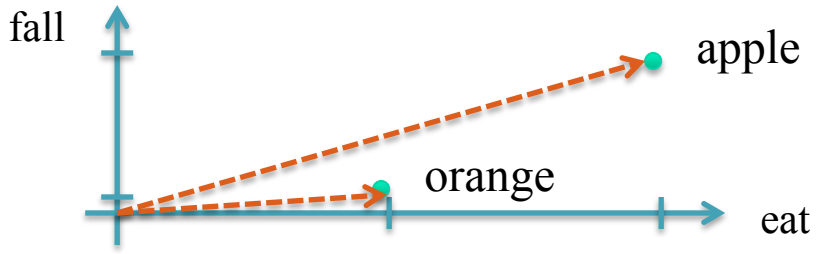
Do the same for “orange”:

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
265	22	25	62	220	64	74	111	4	4	8

How can we compare two context collections in their entirety?

Then visualize both count tables as vectors in the same space:

eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
eat	fall	ripe	slice	peel	tree	throw	fruit	pie	bite	crab
265	22	25	62	220	64	74	111	4	4	8



Similarity between two words as proximity in space

Using distributional models

- Finding (near-)synonyms: automatically building a thesaurus
- Related: use distributional similarity of documents (containing similar words) in Information Retrieval

Where can we find texts to use for making a distributional model?

- Text in electronic form!
- Newspaper articles
- Project Gutenberg: older books available for free
- Wikipedia
- Text collections prepared for language analysis:
  - Balanced corpora
  - WaC: Scrape all web pages in a particular domain
  - ELRA, LDC hold corpus collections
    - For example, large amounts of newswire reports
  - Google n-grams, Google books

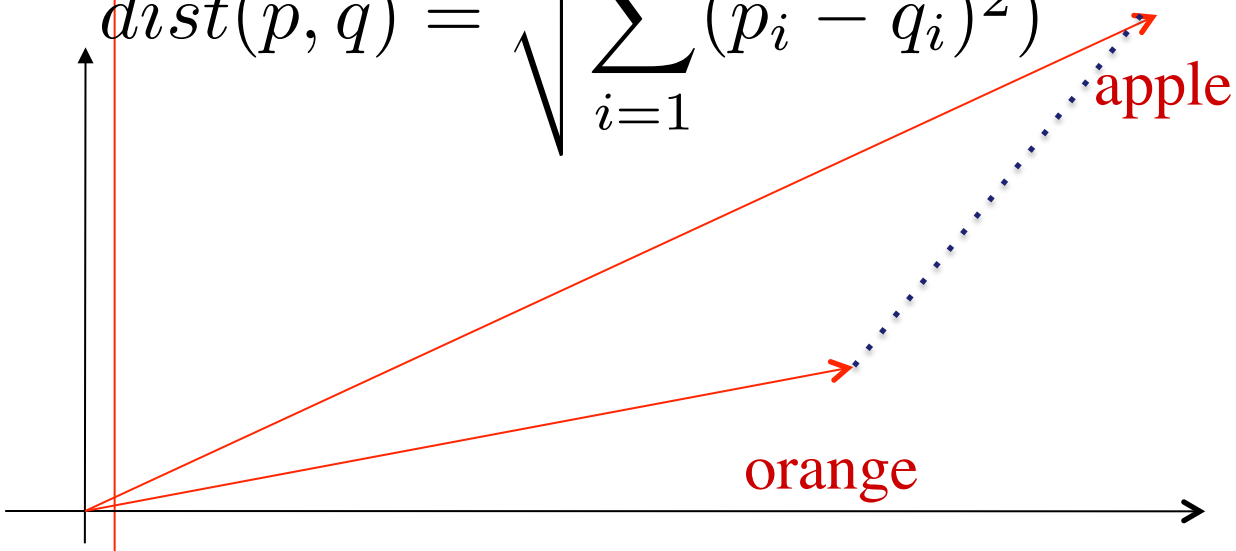
How much text do we need?

- At least:
  British National Corpus, 100 million words
- Better: add
  - UKWaC (2 billion words)
  - Wikipedia (2 billion words)

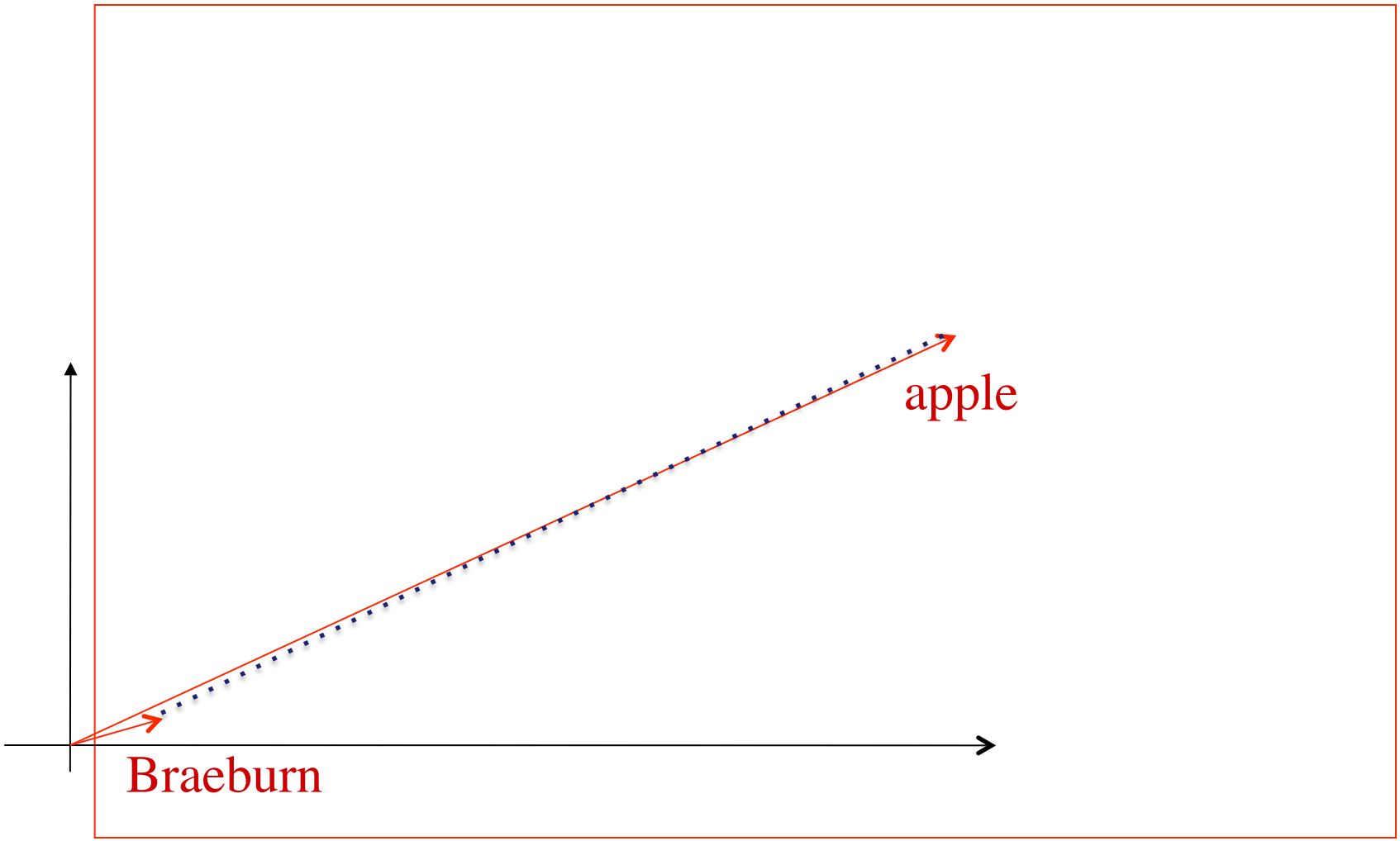
What do we mean by "similarity" of vectors?

Euclidean distance (a dissimilarity measure!):

$$dist(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2)}$$



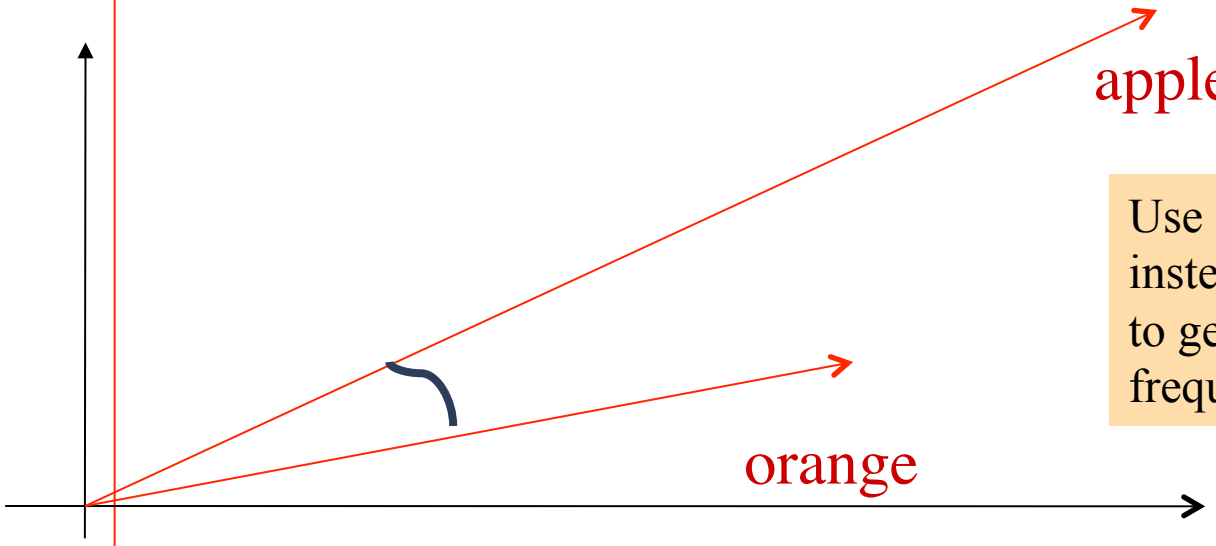
Problem with Euclidean distance: very sensitive to word frequency!



What do we mean by "similarity" of vectors?

Cosine similarity:

$$cos(\vec{p}, \vec{q}) = \frac{\sum_{i=1}^{n} p_i \cdot q_i}{\sqrt{\sum_{i=1}^{n} p_i^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}}$$



Use angle between vectors instead of point distance to get around word frequency issues

Some counts for “letter” in “Pride and Prejudice”. What do you notice?

the	to	of	and	a	her	she	his	is	was	in	that
102	75	72	56	52	50	41	36	35	34	34	33

had	i	from	you	as	this	mr	for	not	on	be	he
32	28	28	25	23	23	22	21	21	20	18	17

but	elizabeth	with	him	which	by	when	jane
17	17	16	16	16	15	14	12

Some counts for “letter” in “Pride and Prejudice”. What do you notice?

the	to	of	and	a	her	she	his	is	was	in	that
102	75	72	56	52	50	41	36	35	34	34	33

had	i	from	you	as	this	mr	for	not	on	be	he
32	28	28	25	23	23	22	21	21	20	18	17

but	elizabeth	with	him	which	by	when	jane
17	17	16	16	16	15	14	12

All the most frequent co-occurring words are function words.

Some words are more informative than others

- Function words co-occur frequently with all words
  - That makes them less informative
- They have much higher co-occurrence counts than content words
  - They can "drown out" more informative contexts

Using association rather than co-occurrence counts

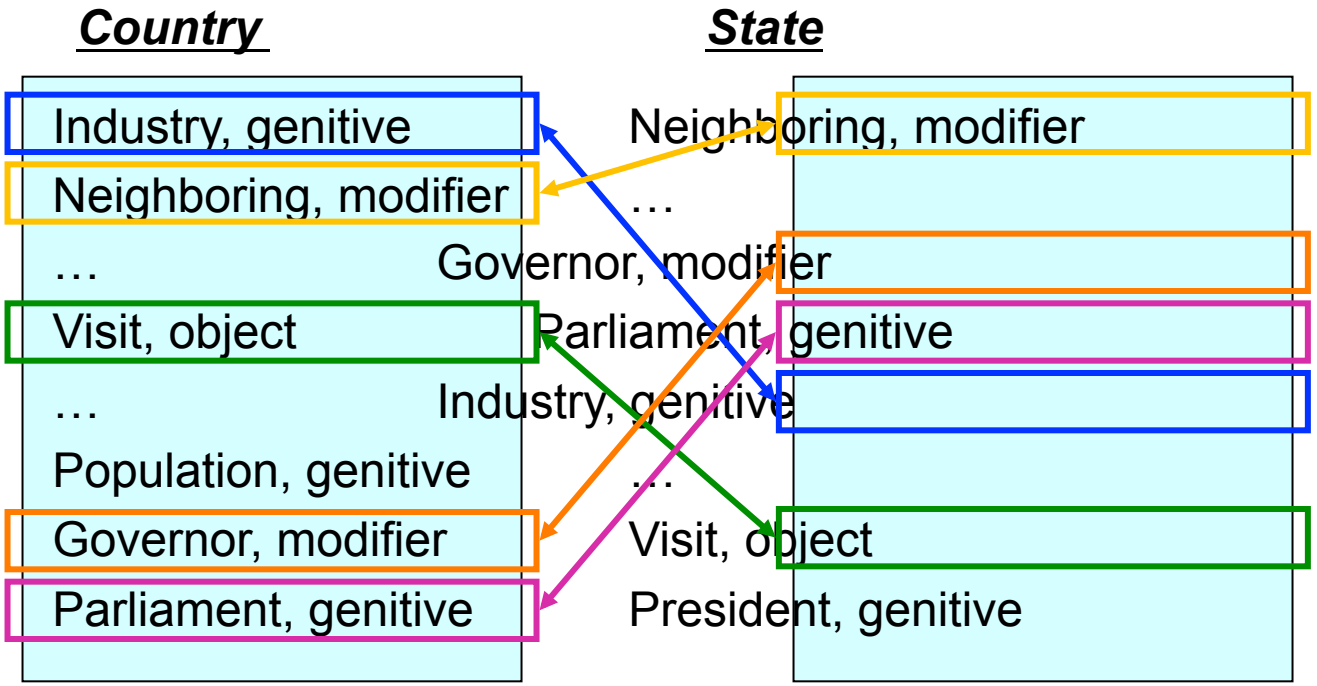
- Degree of association between target and context:
  - High association: high co-occurrence with "letter", lower with everything else
  - Low association: lots of co-occurrence with all words
- Many ways of implementing this
- For example Pointwise Mutual Information between target a and context b:

$$PMI(a,b) = \log \frac{P(a,b)}{P(a) \cdot P(b)}$$

Alternative Co-occurrence Represesntations

- Types of labels on dimensions:
  - Word forms or lemmas
  - Bag-of-words context words:
    - Wide context: topical similarity
    - Narrow context: Mostly dog-animal, not so much dog-leash
      - Good approximation of syntactic-based contexts
  - Syntactic parse relations
    - Mostly dog-animal, not so much dog-leash

Syntactic-based Co-occurrences



Same corpus (BNC), different contexts (window sizes)

Nearest neighbours of *dog*

2-word window

- cat
- horse
- fox
- pet
- rabbit
- pig
- animal
- mongrel
- sheep
- pigeon

30-word window

- kennel
- puppy
- pet
- bitch
- terrier
- rottweiler
- canine
- cat
- to bark
- Alsatian

Similarity Measure Computation

Computing Various Similarity Measures

- Cosine:

$$sim(u,v) = \frac{\sum_{att} log(freq(u,att)) \cdot log(freq(v,att))}{\sqrt{\left(\sum_{att} log(freq(u,att))^2\right) \cdot \left(\sum_{att} log(freq(v,att))^2\right)}}$$

- Weighted Jaccard (Min/Max):

  I: PMI

$$sim(u,v) = \frac{\sum_{att} min(I(u,att), I(v,att))}{\sum_{att} max(I(u,att), I(v,att))}$$

Lin's (1998) Similarity Measure

- Commonly used
  - Feature weights - PMI

$$w_u(f) = \log_2 \frac{P(u,f)}{P(u)P(f)}$$

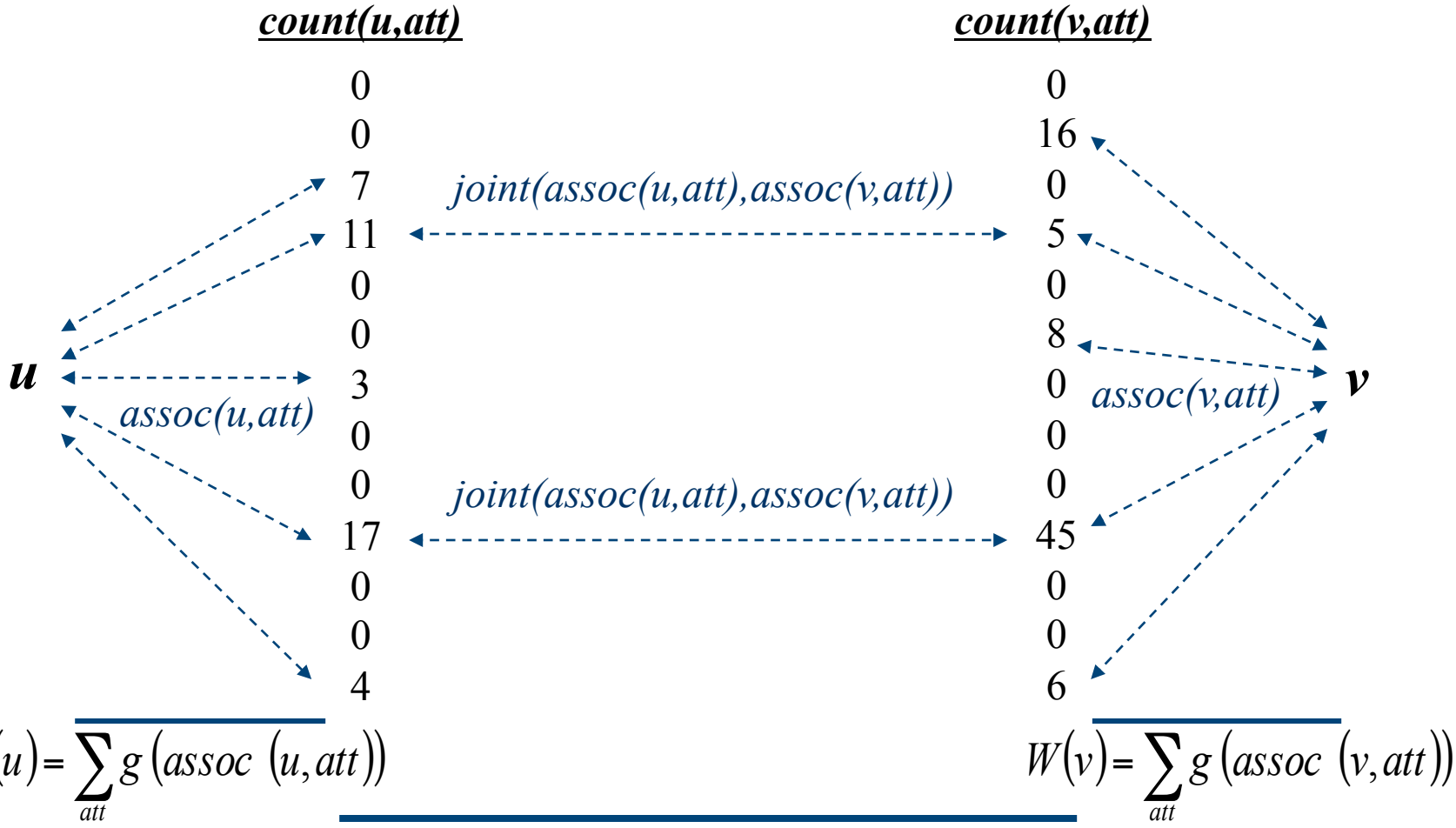
  - Similarity metric

$$sim(u,v) = \frac{\sum_{f \in Fu \cap Fv} (w_u(f) + w_v(f))}{\sum_{f \in Fu} w_u(f) + \sum_{f \in Fv} w_v(f)}$$

A Unifying Schema of Similarity

- A general schema encoding most measures
- Identifies explicitly the important factors that determine (word) similarity
- Provides the basis for:
  - a general and efficient similarity computation procedure
  - evaluating and comparing alternative measures and components

Mapping to Unified Similarity Scheme



$$W(u) = \sum_{att} g(assoc(u, att))$$

$$W(v) = \sum_{att} g(assoc(v, att))$$

$$SJ(u,v) = \sum_{att \in Both(u,v)} joint(assoc(u, att), assoc(v, att))$$

$$f_{SIM}(u,v) = \frac{SJ(u,v)}{norm(SJ(u,v), W(u), W(v))}$$

Association and Joint Association

- *assoc(u,att)*: quantify association strength
  - mutual information, weighted log frequency, conditional probability (orthogonal to scheme)
- *joint*(*assoc*(*u*,*att*),*assoc*(*v*,*att*)): quantify the "similarity" of the two associations
  - ratio, difference, min, product
- $$SJ(u,v) = \sum_{att \in Both(u,v)} joint\left(assoc\left(u,att\right),assoc\left(v,att\right)\right)$$

$$Both(u,v) = \left\{ att \mid freq(u,att) > 0 \,,\, freq(v,att) > 0 \right\}$$

Normalization

- Global weight of a word vector:

$$W(u) = \sum_{att \in Just(u)} g(assoc(u, att)) \qquad Just(u) = \{ att \mid freq(u, att) > 0 \}$$

  - For cosine:

$$W(u) = \sum_{att \in Just(u)} assoc(u, att)^2$$

- Normalization factor:

$$Norm\_Factor(u, v) = norm(SJ(u, v), W(u), W(v))$$

  - For cosine:

$$Norm\_Factor(u, v) = \sqrt{W(u) \cdot W(v)}$$

The General Similarity Scheme

$$sim(u,v) = \frac{SJ(u,v)}{Norm\_Factor(u,v)} = \frac{SJ(u,v)}{norm(SJ(u,v),W(u),W(v))}$$

where

$$SJ(u,v) = \sum_{att \in Both(u,v)} joint\left(assoc\left(u,att\right),assoc\left(v,att\right)\right)$$

- For example - cosine :

$$sim(u,v) = \frac{SJ(u,v)}{\sqrt{W(u)\cdot W(v)}}$$

Min/Max Measures

$$sim(u,v) = \frac{\sum_{att} \min(assoc(u,att), assoc(v,att))}{\sum_{att} \max(assoc(u,att), assoc(v,att))}$$

- May be viewed as (assuming non negative assoc):

$$\text{joint}(\cdot,\cdot) = \min(assoc(u,att), assoc(v,att)$$

- What about norm?

$$\sum_{att \in Either(u,v)} max(assoc(u,att), assoc(v,att)) =$$

$$\sum_{att} assoc(u,att) + \sum_{att} assoc(v,att) - \sum_{att \in Both(u,v)} min(assoc(u,att), assoc(v,att))$$

Associations Used with *Min/Max*

- Point-wise mutual information (used by Dagan et al., 1993/5):

$$assoc(u, att) = \log \frac{P(u, att)}{P(u)P(att)} = \log \frac{P(att|u)}{P(att)} = \log \frac{P(u|att)}{P(u)}$$

Cosine Measure

$$\cos(u,v) = \frac{\sum_{att} assoc(u,att) \cdot assoc(v,att)}{\sqrt{\sum_{att} assoc(w_1,att)^2} \cdot \sqrt{\sum_{att} assoc(w_2,att)^2}}$$

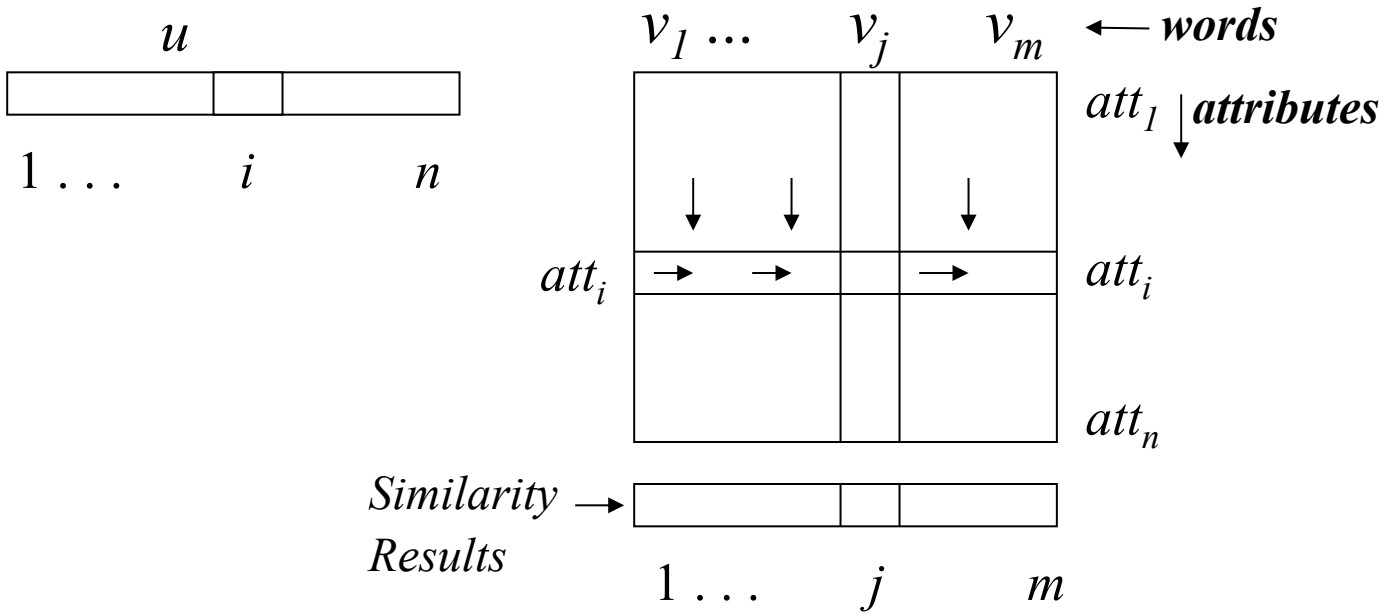
- Used for word similarity (Ruge, 1992) with: $assoc(u,att) = \ln(freq(u,att))$
- Popular for document ranking (vector space)

$$assoc(doc,term) = tf \cdot idf$$

$$tf = \frac{freq(doc,term)}{\max freq(doc,\cdot)} \qquad idf = \log \frac{\max docfreq(\cdot)}{docfreq(term)} + 1$$

Efficient Computation

- Efficient implementation through sparse matrix indexing
  - By computing over common attributes only (*both*)



- *Pseudocode – next slide*
- *Complexity reduced by "sparseness" factor –*
  #non-zero cells / total #cells
  - *Two orders of magnitude in corpus data*

Computing SJ for a word *u*

(1) For each *att* in *ATT*(*u*)
(2) For each *v* in W(*att*)
(3) *SJ*(*u*,*v*) = *SJ*(*u*,*v*) + joint(assoc(u,att),assov(v,att))

PMI for Distributional Vectors

$$PMI(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

- *PMI(x,y)* is defined for a *joint distribution* of two random variables *X* and *Y,* measuring the association between the pair of values *x* and *y*
  - <*x,y*> is an *ordered* pair
- *PMI(u,att)* is defined for a space of co-occurrences of words (*X*) with attributes/features (*Y*)
  - Co-occurrence in a window – *att* is a word *v*
  - Co-occurrence in a syntactic relation – *att* is a pair of a word and the *directional* dependency relation
    - E.g.: <*read,* <*book, obj*↓>> vs. <*book,* <*read, obj*↑>>
  - Notice: each such word co-occurrence is counted *twice*

Estimating *PMI* probabilities

- Probability estimation is based on *#(u,att)* – number of times *u* counted (occurred) with *att*
  - For "symmetric" word co-occurrence features: *#(*, *)* is twice the number of co-occurrences observed in corpus
- *p(u,att)* – the probability that a random word-attribute co-occurrence will have *u* as the word and *att* as the attribute
- *p(u)* [*p(att)*] – the probability that a random word-attribute co-occurrence will have *u* as the word [*att* as the attribute]

$$p(u,att) = \frac{\#(u,att)}{\#(*,*)} \quad p(u) = \frac{\#(u,*)}{\#(*,*)} \quad p(att) = \frac{\#(*,att)}{\#(*,*)}$$

Estimation Sanity Checks

- Each distribution (for each *u,att*) sums to 1:

$$\sum_{u,att} p(u,att) = \sum_{u} p(u) = \sum_{att} p(att) = 1$$

- Definition of marginal probabilities:

$$p(u) = \sum_{att} p(u,att)$$

- *PMI* is symmetric: *PMI(u,v)=PMI(v,u)*

$$p(att) = \sum_{u} p(u,att)$$

Distributional Inclusion for Directional Similarity

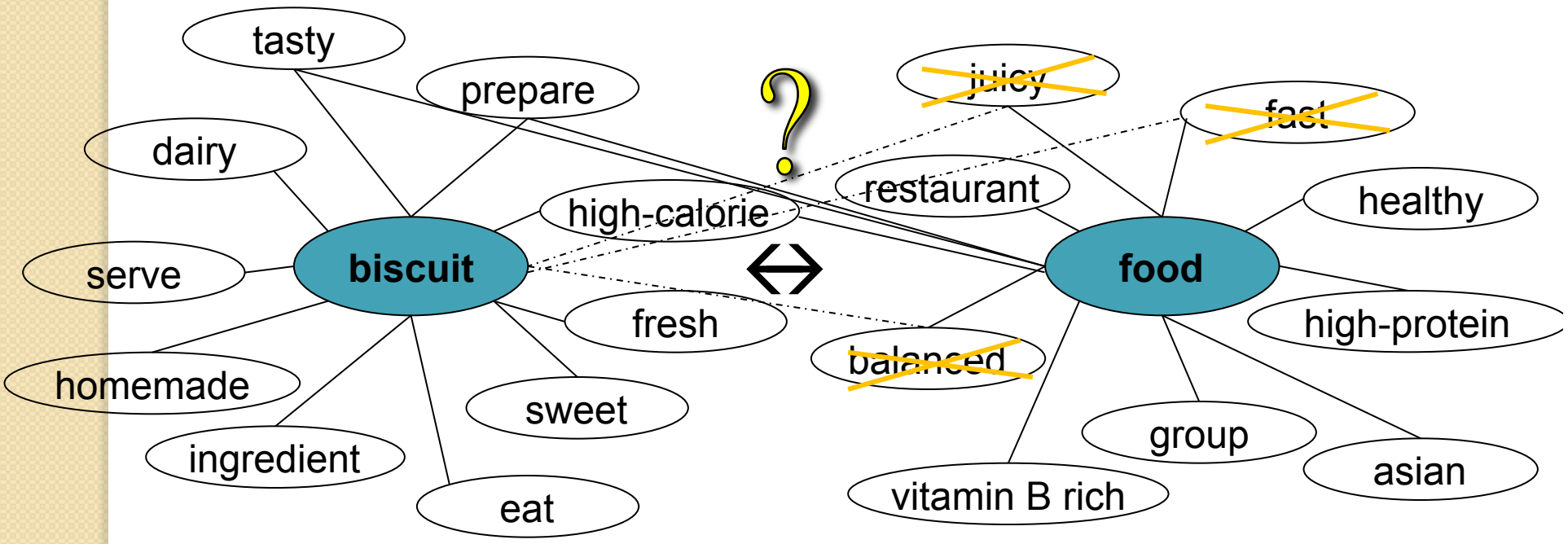
Symmetric Similarity

- The top most similar words for ***food***

meat	clothing	water	sugar
beverage	foodstuff	coffee	material
goods	textile	meal	chemical
medicine	fruit	tobacco	equipment
drink	feed	fuel	rice

Directional Similarity: $u \rightarrow v$

- ***Distributional inclusion***:
  - If $u \rightarrow v$, then all the characteristic contexts of u are expected to be possible contexts for v
    (Weeds et al., 2004; Geffet and Dagan, 2005)



Inclusion measures : $u \rightarrow v$

- Weeds and Weir (2003)

$$WeedsPrecision(u \rightarrow v) = \frac{\sum_{f \in Fu \cap Fv} w_u(f)}{\sum_{f \in Fu} w_u(f)}$$

- Szpektor and Dagan (2008) *"pierce X"* → *"marry X"*

$$balPrecision(u \rightarrow v) = sim_{Lin}(u,v) \cdot WeedsPrecision(u \rightarrow v)$$

Evaluations of Word Similarity

Empirical Evaluation

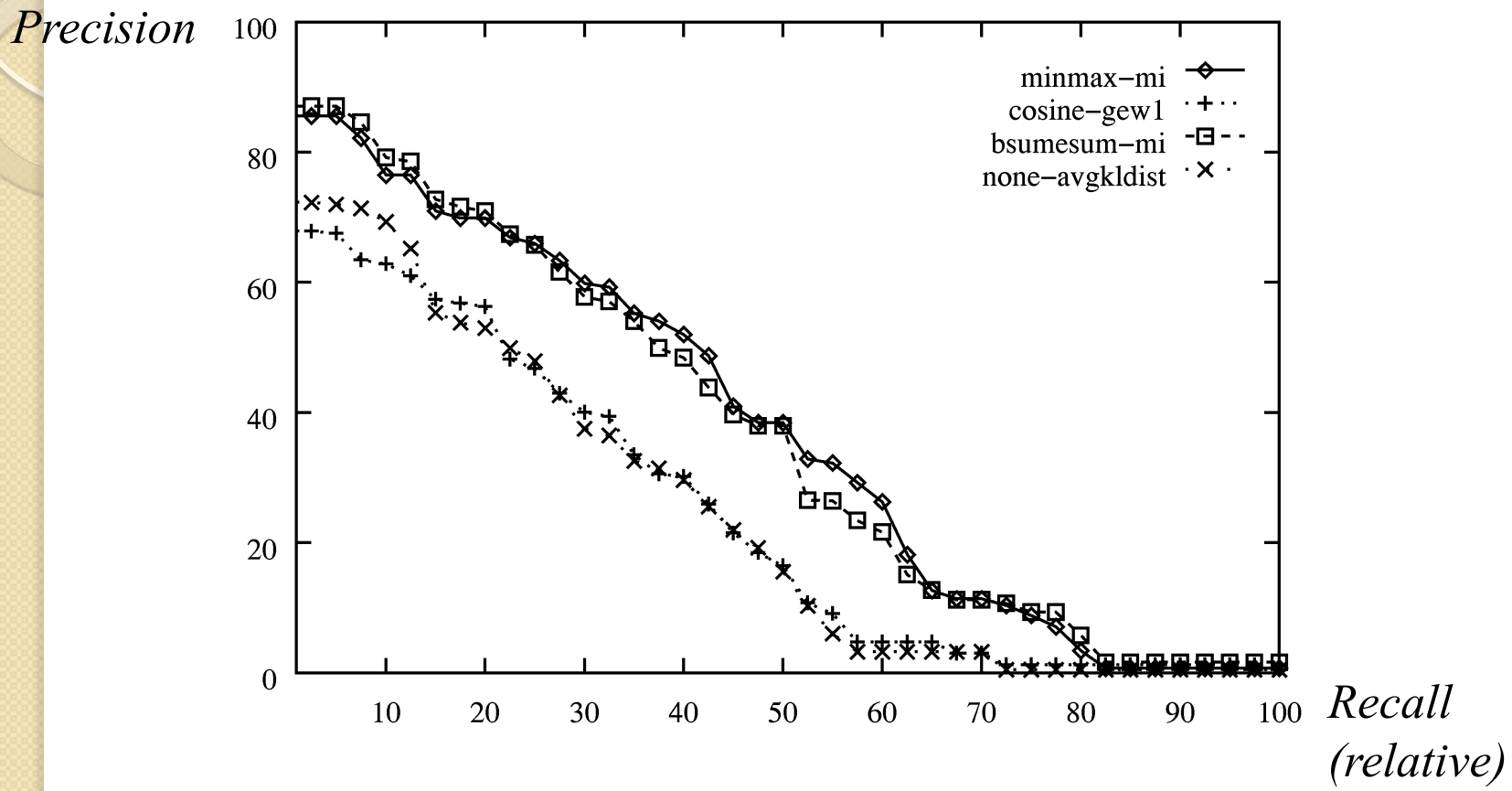
- Thesaurus for query expansion (e.g. *"insurance laws"*):

Top similar words for *law* :

Word	***Similarity***	***Judgment (for QE)***
regulation	0.050242	+
rule	0.048414	+
legislation	0.038251	+
guideline	0.035041	+
commission	0.034499	–
bill	0.033414	+
budget	0.031043	–
regulator	0.031006	+
code	0.030998	+
circumstance	0.030534	–

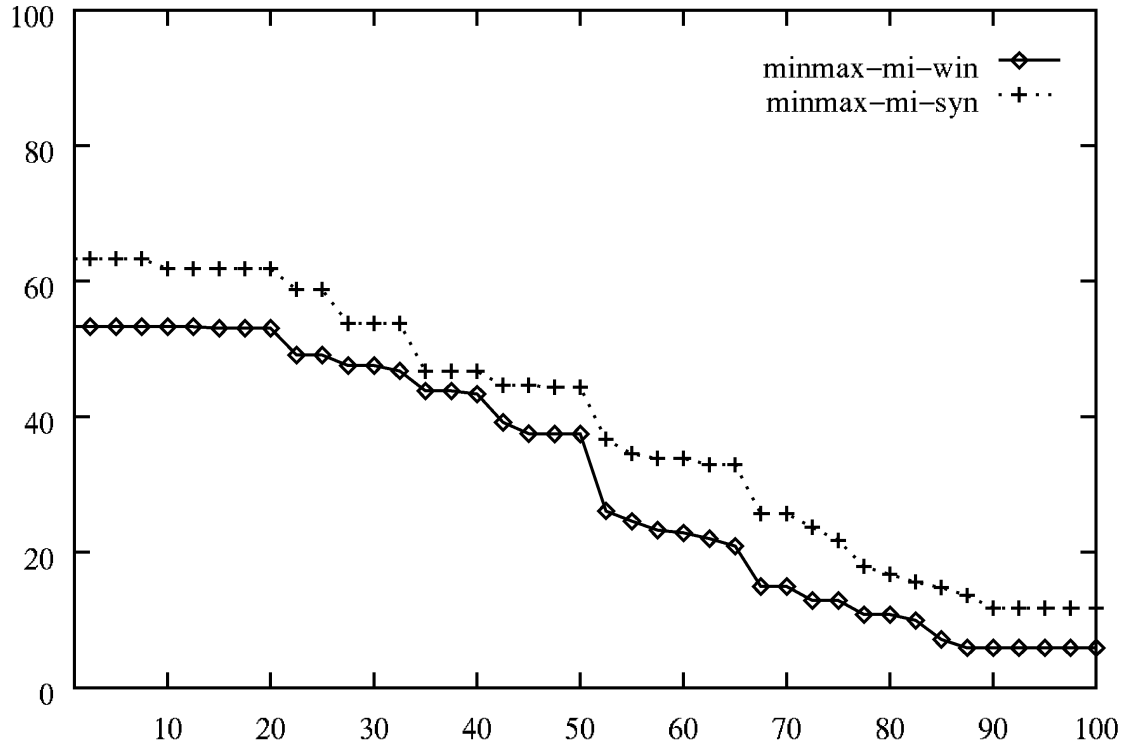
- **Precision** and **relative Recall** at each point in the list
  - Post-hoc evaluation

Comparing Measure Combinations



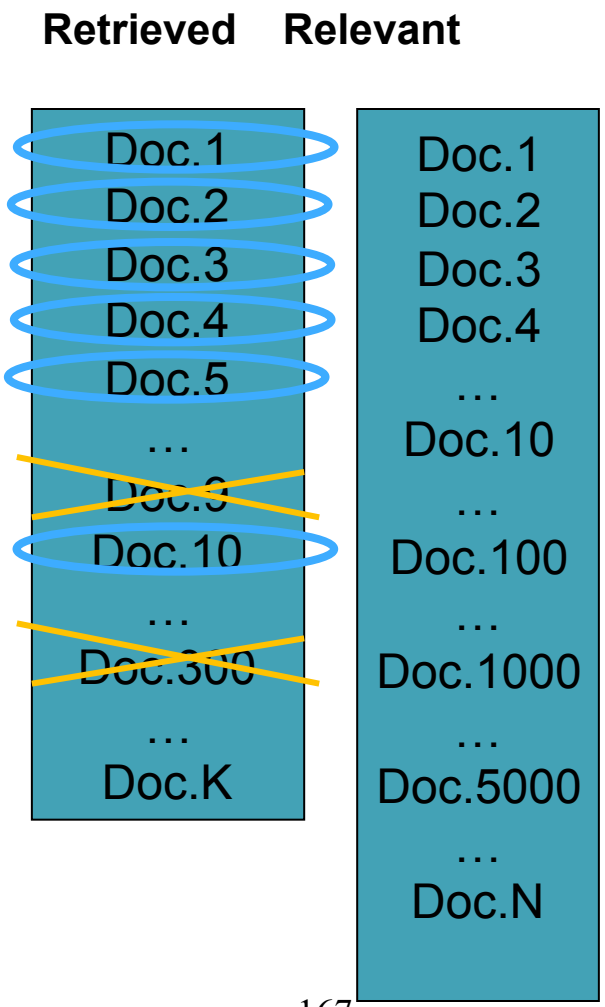
- Min/Max schemes worked better than cosine and Jensen-Shannon (almost by 20 points); stable over association measures

Effect of Co-occurrence Type on Semantic Similarity (R/P curve)



Average Precision Evaluation Measure

- Apply *Average Precision* measure, common in *IR* for comparison of different search engines
- A good engine should:
  - find many relevant documents
  - find no or not many irrelevant documents
  - place relevant documents at high ranks in the list



$$AP = \frac{\sum_{R=1}^{K} [Prec_R \cdot rel_R]}{N}; \qquad rel \text{ – } indicator \ function$$

$$Prec_R = \frac{1}{R} \sum_{i=1}^{R} rel_i$$

AP For Word Similarity

- Retrieved:
  - Similar words proposed by similarity measure
  - Top ranked – by top-*k* or threshold (as in IR)
- Relevant:
  - Words judged as similar
    - From the union retrieved by all evaluated measures (as in relative recall, for post-hoc evaluation)
    - From a pre-defined gold standard (e.g. BLESS)
      - Judge only retrieved appearing in gold standard (pos/neg)
- Mean Average Precision (MAP)
  - Average AP values over all target words (queries)

How we BLESSed distributional semantic evaluation

Marco Baroni[1], Alessandro Lenci[2]

[1]Università di Trento, CIMeC
[2]Università di Pisa, Dipartimento di Linguistica

GEMS-2011, GEometrical Models of Natural Language Semantics
EMNLP, Edinburgh, 31st July 2011

Evaluation in distributional semantics

- Current approaches to the evaluation of Distributional Semantic Models (DSMs) are task-oriented
- Model performance is evaluated in "semantic tasks"
  - detecting synonyms
  - recognizing analogies
  - modeling verb selectional preferences
  - categorization
  - etc.
- These tasks represent a form of extrinsic evaluation of DSMs (Spark Jones & Galliers 1996) since they provide indirect tests of the ability of DSMs to capture lexical meaning

BLESS

Baroni and Lenci Evaluation of Semantic Spaces

- BLESS is a new data set specifically geared towards the intrinsic evaluation of DSMs (Spark Jones & Galliers 1996)
  - singles out a particular aspect of meaning to be focused on for the evaluation of DSMs
  - presents a new design that is able to explicitly and reliably encode the target semantic information
  - proposes evaluation criteria of the system performance on the data set
- The goal is to perform direct tests of the semantic spaces produced by DSMs
- BLESS has been used in the GEMS - 2011 Shared Task

A snapshot of BLESS

- BLESS is formed by 26,554 tuples expressing a relation between a (target) concept and a relatum (concept)
  - 200 basic-level nominal concrete concepts, 8 relation types, each instantiated by multiple relata (nouns, verbs or adjectives)
  - a number of relata are random, i.e. not semantically related to the concept

target concept	relation	relata
rabbit	HYPER	*animal, chordate, mammal, . . .*
guitar	COORD	*violin, trumpet, piano, . . .*
beaver	MERO	*fur, head, tooth, . . .*
sword	ATTRI	*dangerous, long, heavy, . . .*
butterfly	EVENT	*fly, catch, flutter, . . .*
villa	RAN.N	*disease, assistance, game, . . .*
donkey	RAN.V	*coincide, express, vent, . . .*
hat	RAN.J	*quarterly, massive, obvious, . . .*

Relations

COORD the relatum is a co-hyponym (coordinate) of the concept
- *guitar* COORD *violin*

HYPER the relatum is a hypernym of the concept
- *rabbit* HYPER *animal*

MERO the relatum is a noun referring to a part / organ / member of the concept, or something that the concept contains or is made of
- *beaver* MERO *fur*

ATTRI the relatum is an adjective expressing an attribute of the concept
- *sword* ATTRI *dangerous*

EVENT the relatum is a verb referring to an action / activity / happening / event the concept is involved in or is performed by / with it
- *butterfly* EVENT *fly*

RAN.K the relatum is a random noun (k=n), adjective (k =j) and verb (k=v) semantically unrelated to the target
- *donkey* RAN.V *coincide*

"True" relata selection

- The 14,440 relata of the non-random relations are English nouns, verbs and adjectives selected and validated by both authors
  - semantic sources - the McRae Norms (McRae *et al.*2005), WordNet (Fellbaum 1998) and ConceptNet (Liu & Singh 2004)
  - text sources - Wikipedia and ukWaC corpus
- The BLESS relata represent a wide spectrum of features of the target concepts:
  - domain-specific features (e.g., *gymnosperm* for *pine*)
  - commonsense features (e.g., the events *park* and *steal* for *car*)
  - very distinctive features of a concept (e.g., *hoot* for *owl*)
  - attributes and events shared by a whole class of concepts (e.g., all animals have relata such as *eat*, *feed*, and *live*)
  - prototypical features (e.g. *red* for *apple*)
  - statistically salient events and attributes (e.g. *hunt*, *kill* for *elephant*)
- Phrasal relata have always been reduced to their head
  - *lay eggs* $\rightarrow$ *lay*

“Random” relata selection

- For each “true” relatum, another lemma was randomly picked from the BLESS corpus with the following constraints:
  - same pos
  - frequency within 1 absolute log unit from the frequency of the corresponding “true” relatum
- The 15K extracted words were filtered with Amazon Mechanical Turk via the CrowdFlower interface (CF)
  - for each pair, workers checked a YES radio button if they thought there is a relation between the words, NO otherwise
  - a minimum of 2 workers rated each pair
- BLESS contains only the 12,154 random relata unanimously rated as unrelated to their target concepts

Target concepts

- Target concepts are 200 English concrete nouns (100 living and 100 non-living) grouped into 17 broader classes
  - AMPHIBIAN_REPTILE (including amphibians and reptiles: *alligator*)
  - APPLIANCE (*toaster*)
  - BIRD (*crow*)
  - BUILDING (*cottage*)
  - CLOTHING (*sweater*)
  - CONTAINER (*bottle*)
  - FRUIT (*banana*)
  - FURNITURE (*chair*)
  - GROUND_MAMMAL (*beaver*)
  - INSECT (*cockroach*)
  - MUSICAL_INSTRUMENT (*violin*)
  - TOOL (i.e., manipulable tools or devices: *hammer*)
  - TREE (*birch*), VEGETABLE (*cabbage*)
  - VEHICLE (*bus*)
  - WATER_ANIMAL (including fish and sea mammals: *herring*)
  - WEAPON (*dagger*)

Conclusions

- BLESS is the first data set specifically designed for the intrinsic evaluation of DSMs
- It contains tuples instantiating different, explicitly typed semantic relations, plus a number of controlled random tuples
- Experiments have shown that is able to highlight interesting differences in the semantic spaces produced by various models
- The extension of BLESS to verbs and other class of nouns is ongoing
- Distribution
  - freely downloadable from:
    `http://clic.cimec.unitn.it/distsem`

Essential references

- Agirre E., Alfonseca, E., Hall, K., Kravalova, J., Pasça, M. and Soroa, A. (2009), "A study on similarity and relatedness using distributional and WordNet-based Approaches", *Proceedings of HLT-NAACL*, 19–27.
- Baroni, M. and Lenci A. (2010), "Distributional memory: A general framework for corpus-based semantics". *Computational Linguistics*, 36(4): 673–721.
- Baroni, M. and Lenci A. (forthcoming), *Distributional Semantics*, Cambridge, Cambridge University Press
- McRae, K., Cree, G., Seidenberg, M. and McNorgan C. (2005), "Semantic feature production norms for a large set of living and nonliving things" *Behavior Research Methods*, 37(4):547–559.
- Sparck Jones, K. and Galliers, J.R. (1996), *Evaluating Natural Language Processing Systems: An Analysis and Review*, Berlin, Springer Verlag.

Learning Inference Rules between Templates by Distributional Similarity

- Lin and Pantel, JNLE 2001

DIRT: Inference Rules for Predicate Templates

Table 3. The top-50 most similar paths to "*X solves Y*".

1.	Y is solved by X	26.	X clears up Y
2.	X resolves Y	27.	*X creates Y
3.	X finds a solution to Y	28.	*Y leads to X
4.	X tries to solve Y	29.	Y is eased between X
5.	X deals with Y	30.	X gets down to Y
6.	Y is resolved by X	31.	X worsens Y
7.	X addresses Y	32.	X ends Y
8.	X seeks a solution to Y	33.	*X blames something for Y
9.	X do something about Y	34.	X bridges Y
10.	X solution to Y	35.	X averts Y
11.	Y is resolved in X	36.	*X talks about Y
12.	Y is solved through X	37.	X grapples with Y
13.	X rectifies Y	38.	*X leads to Y
14.	X copes with Y	39.	X avoids Y
15.	X overcomes Y	40.	X solves Y problem
16.	X eases Y	41.	X combats Y
17.	X tackles Y	42.	X handles Y
18.	X alleviates Y	43.	X faces Y
19.	X corrects Y	44.	X eliminates Y
20.	X is a solution to Y	45.	Y is settled by X
21.	X makes Y worse	46.	*X thinks about Y
22.	X irons out Y	47.	X comes up with a solution to Y
23.	*Y is blamed for X	48.	X offers a solution to Y
24.	X wrestles with Y	49.	X helps somebody solve Y
25.	X comes to grip with Y	50.	*Y is put behind X

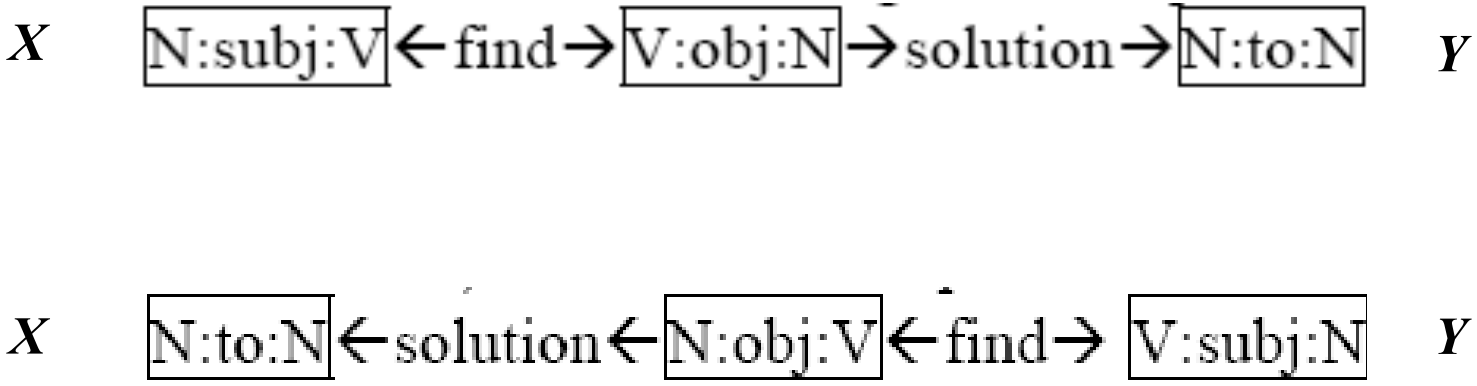
Distributional Hypothesis for Paths

Extended Distributional Hypothesis:
If two paths tend to occur in similar contexts, the meanings of the paths tend to be similar.

Table 2. Sample slot fillers for two paths extracted from a newspaper corpus.

"X finds a solution to Y"		*"X solves Y"*	
SLOTX	*SLOTY*	*SLOTX*	*SLOTY*
commission	strike	committee	problem
committee	civil war	clout	crisis
committee	crisis	government	problem
government	crisis	he	mystery
government	problem	she	problem
he	problem	petition	woe
I	situation	researcher	mystery
legislator	budget deficit	resistance	crime
sheriff	dispute	sheriff	murder

Dependency Paths (in both directions)



Path Similarity

- Between X (Y) slots of two paths:

$$sim(slot_1, slot_2) = \frac{\sum_{w \in T(p_1,s) \cap T(p_2,s)} mi(p_1, s, w) + mi(p_2, s, w)}{\sum_{w \in T(p_1,s)} mi(p_1, s, w) + \sum_{w \in T(p_2,s)} mi(p_2, s, w)}$$

- Between two paths:

$$S(p_1, p_2) = \sqrt{sim(SlotX_1, SlotX_2) \times sim(SlotY_1, SlotY_2)}$$

- Simple efficiency technique: compute for paths that share at least one feature, and at least 1% of the union of their features.

Table 7. Paths found for five of the 15 questions in TREC-8 and the variations discovered manually and by DIRT.

Q	PATHS	MANUAL VARIATIONS	DIRT VARIATIONS
Q_1	*X* is author of *Y*	*Y* is the work of *X*; *X* is the writer of *Y*; *X* penned *Y*; *X* produced *Y*; *X* authored *Y*; *X* chronicled *Y*; *X* wrote *Y*	*X* co-authors *Y*; *X* is co-author of *Y*; *X* writes Y; *X* edits *Y*; *Y* is co-authored by *X*; *Y* is authored by *X*; *X* tells story in *Y*; *X* translates *Y*; *X* writes in *Y*; *X* notes in *Y*; …
Q_3	*X* manufactures *Y*	*X* makes *Y*; *X* produce *Y*; *X* is in *Y* business; *Y* is manufactured by *X*; *Y* is provided by *X*; *Y* is *X*'s product; *Y* is product from *X*; *Y* is *X* product; *Y* is product made by *X*; *Y* is example of *X* product; *X* is manufacturer of *Y*; find *Y* in *X*'s product line; find *Y* in *X* atalog	*X* produces *Y*; *X* markets *Y*; *X* develops *Y*; *X* is supplier of *Y*; *X* ships *Y*; *X* supplies *Y*; *Y* is manufactured by *X*; *X* is maker of *Y*; *X* introduces *Y*; *X* exports *Y*; *X* makes *Y*; *X* builds *Y*; *X*'s production of *Y*; *X* unveils *Y*; *Y* is bought from *X*; *X*'s line of *Y*; *X* assembles *Y*; *X* is *Y* maker; *X*'s *Y* factory; *X*'s *Y* production; *X* is manufacturer of *Y*; *X*'s *Y* division; *X* meets demand for *Y*; …
Q_4	*X* spend *Y*	*X* put *Y* into marketing; at *X*, *Y* was spent; *X* invest *Y*; *X* pay *Y*; *Y* is *X*'s budget; *Y* is *X*'s outlay; *Y* is *X*'s spending	*X* invests *Y*; *X* pays *Y*; *X* pays somebody *Y*; *X* contributes *Y*; *Y* is spent by *X*; *X* allocates *Y*; *X* wastes *Y*; *X* pours *Y*; *X* puts up *Y*; …
	spend *X* on *Y*	put *X* into *Y*; *X* was spent on *Y*; invest *X* in *Y*; pay *X* for *Y*; *X* is *Y* budget; *X* is *Y* outlay; *X* is spending for *Y*; *X* is *Y* spending	*X* pays for *Y*; *X* spends something for *Y*; *X*'s *Y* budget; *X* finances *Y*; *X* purchases *Y*; *X* goes ahead with *Y*; …

Table 5. Evaluation of Top-40 most similar paths.

QUESTION	PATHS	MANUAL	DIRT (CORRECT)	INTERSECTION	ACCURACY
Q_1	X is author of Y	7	21	2	52.5%
Q_2	X is monetary value of Y	6	0	0	0%
Q_3	X manufactures Y	13	37	4	92.5%
Q_4	X spend Y	7	16	2	40.0%
	spend X on Y	8	15	3	37.5%
Q_5	X is managing director of Y	5	14	1	35.0%
Q_6	X asks Y	2	23	0	57.5%
	asks X for Y	2	14	0	35.0%
	X asks for Y	3	21	3	52.5%
Q_7	X leave Y	4	0	0	0%
Q_8	X is disease with Y	5	0	0	0%
Q_9	∅	N/A	N/A	N/A	N/A
Q_{10}	X is designer of Y	5	7	2	17.5%
Q_{11}	∅	N/A	N/A	N/A	N/A
Q_{12}	∅	N/A	N/A	N/A	N/A
Q_{13}	rent X for Y	14	16	1	40.0%
Q_{14}	X is producer of Y	10	31	3	77.5%
Q_{15}	∅	N/A	N/A	N/A	N/A

Distributional Similarity Tool

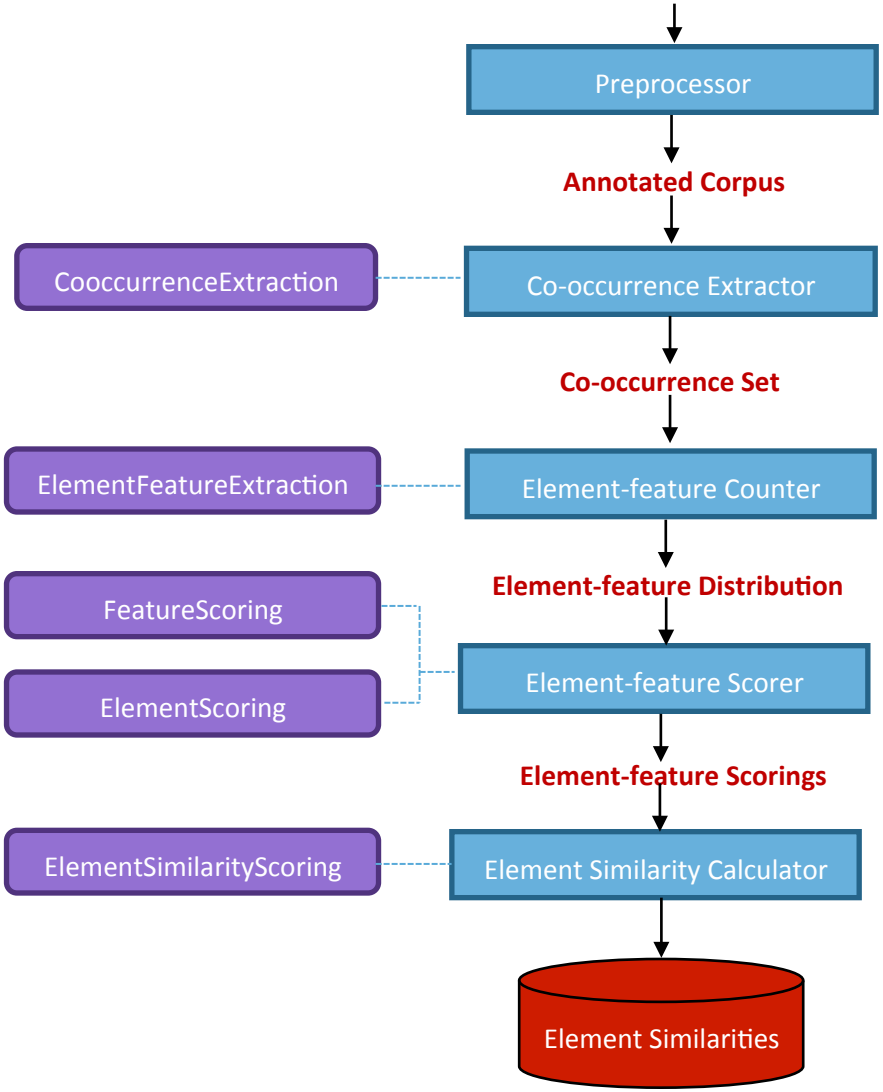
EXCITEMENT Open Platform

Meni Adler, Bar-Ilan University

Quick-start resource generation guide:
https://github.com/hltfbk/Excitement-Open-Platform/wiki/Resource-generation-guide

User guide documentation:
https://github.com/hltfbk/Excitement-Open-Platform/wiki/Distsim-user-guide

Architecture – Single Measure



Selected Interfaces

```
public interface FeatureScoring {
    double score(Element element, Feature feature,
        final double totalElementCount,
        final double jointCount)
            throws ScoringException;
}
```

- Implementations
  - Count
  - TFIDF
  - Dice
  - PMI
  - Prob

Selected Interfaces

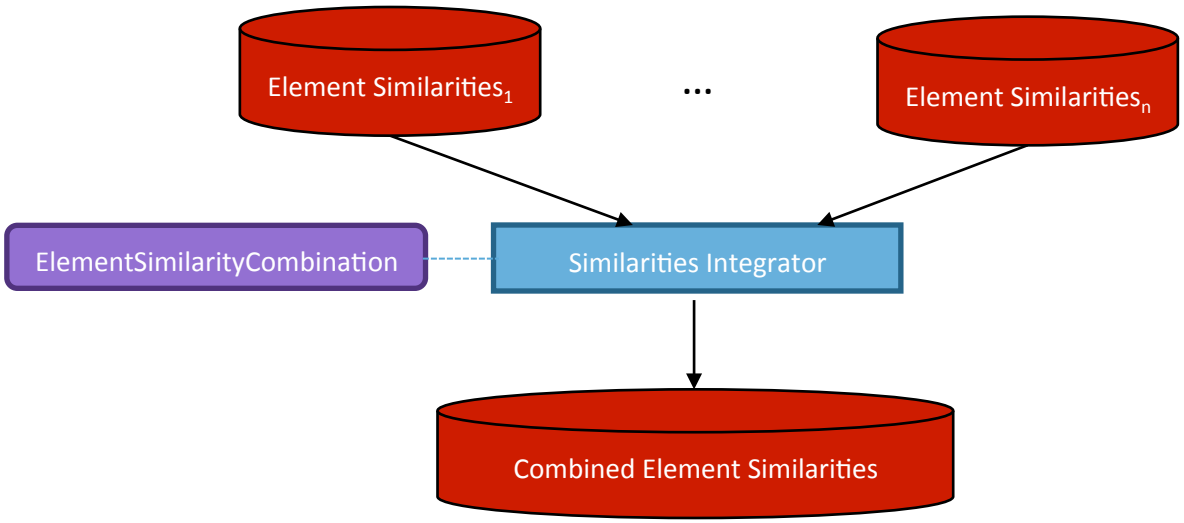
```
public interface ElementSimilarityScoring  {

   void addElementFeatureScore(
            double leftElementFeatureScore,
            double rightElementFeatureScore);

   double getSimilarityScore(double leftDenominator,
                             double rightDenominator);
}
```

- Implementations
  - Cosine
  - Lin
  - Cover
  - APinc

Architecture – Integration of Multiple Measures



Examples

- Lin
  - Co-occurrences: pair of lemmas with their dependency relations
  - Elements: lemmas
  - Features: lemmas and their dependency relations
  - Feature scoring: PMI
  - Vector similarities: Lin
- DIRT
  - Co-occurrences: dependency paths and their arguments
  - Elements: dependency paths
  - Features: X arguments, Y arguments
  - Feature scoring: PMI
  - Vector similarities: Lin
  - Scoring integration: Geometric mean of X-based and Y-based scores