Lexical Semantics

Pawan Goyal

CSE, IIT Kharagpur

Week 8, Lecture 1

Lexical Semantics

Definition

Lexical semantics is concerned with the systematic meaning related connections among lexical items, and the internal meaning-related structure of individual lexical items.

Definition

Lexical semantics is concerned with the systematic meaning related connections among lexical items, and the internal meaning-related structure of individual lexical items.

To identify the semantics of lexical items, we need to focus on the notion of **lexeme**, an individual entry in the lexicon.

Lexical Semantics

Definition

Lexical semantics is concerned with the systematic meaning related connections among lexical items, and the internal meaning-related structure of individual lexical items.

To identify the semantics of lexical items, we need to focus on the notion of **lexeme**, an individual entry in the lexicon.

What is a lexeme?

Lexeme should be thought of as a pairing of a particular orthographic and phonological form with some sort of symbolic meaning representation.

- Orthographic form, and phonological form refer to the appropriate form part of a lexeme
- Sense refers to a lexeme's meaning counterpart.

Example

verge [1] | vərj |

noun

an edge or border: *they came down to the verge of the lake.*

- an extreme limit beyond which something specified will happen: *I was* ***on the verge of*** *tears.*
- Brit.a grass edging such as that by the side of a road or path.
- Architecture an edge of tiles projecting over a gable.

verb [no obj.] (**verge on**)

approach (something) closely; be close or similar to (something): *despair verging on the suicidal.*

ORIGIN late Middle English: via Old French from Latin ***virga*** ***'rod.'*** The current verb sense dates from the late 18th cent.

verge [2] | vərj |

noun

a wand or rod carried before a bishop or dean as an emblem of office.

ORIGIN late Middle English: from Latin ***virga*** ***'rod.'***

verge [3] | vərj |

verb [no obj.]

incline in a certain direction or toward a particular state: *his style verged into the art nouveau school.*

ORIGIN early 17th cent. (in the sense ***'descend (to the horizon)'***): from Latin ***vergere*** ***'to bend, incline.'***

Example: meaning related facts?

Definitions from the American Heritage Dictionary (Morris, 1985)

- **right** *adj.* located near the right hand esp. being on the right when facing the same direction as the observer
- **left** *adj.* located near to this side of the body than the right
- **red** *n.* the color of blood or a ruby
- **blood** *n.* the red liquid that circulates in the heart, arteries and veins of animals

Example: meaning related facts?

Definitions from the American Heritage Dictionary (Morris, 1985)

- **right** *adj.* located near the right hand esp. being on the right when facing the same direction as the observer
- **left** *adj.* located near to this side of the body than the right
- **red** *n.* the color of blood or a ruby
- **blood** *n.* the red liquid that circulates in the heart, arteries and veins of animals

- The entries are description of lexemes in terms of other lexemes
- Definitions make it clear that *right* and *left* are similar kind of lexemes that stand in some kind of alternation, or opposition, to one another
- We can glean that *red* is a color, it can be applied to both *blood* and *rubies*, and that *blood* is a liquid.

Relations between word meanings

- Homonymy
- Polysemy
- Synonymy
- Antonymy
- Hypernymy
- Hyponymy
- Meronymy

Homonymy

Definition

Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings.

Homonymy

Definition

Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings.

Examples

- Bat (wooden stick-like thing) vs Bat (flying mammal thing)
- Bank (financial institution) vs Bank (riverside)

Homonymy

Definition

Homonymy is defined as a relation that holds between words that have the same form with unrelated meanings.

Examples

- Bat (wooden stick-like thing) vs Bat (flying mammal thing)
- Bank (financial institution) vs Bank (riverside)

homophones and homographs

homophones are the words with the same pronunciation but different spellings.

- write vs right
- piece vs peace

homographs are the lexemes with the same orthographic form but different meaning. Ex: bass

Problems for NLP applications

Text-to-Speech

Same orthographic form but different phonological form

Problems for NLP applications

Text-to-Speech

Same orthographic form but different phonological form

Information Retrieval

Different meaning but same orthographic form

Problems for NLP applications

Text-to-Speech

Same orthographic form but different phonological form

Information Retrieval

Different meaning but same orthographic form

Speech Recognition

to, two, too

Perfect homonyms are also problematic

Polysemy

*Multiple **related** meanings within a single lexeme.*

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

Polysemy

*Multiple **related** meanings within a single lexeme.*

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

Are those the same sense?

- Sense 1: “The building belonging to a financial institution”
- Sense 2: “A financial institution”

Another example

- Heavy snow caused the roof of the *school* to collapse.
- The *school* hired more teachers this year than ever before.

Polysemy: multiple related meanings

Often, the relationships are systematic

E.g., building vs. organization

school, university, hospital, church, supermarket

Polysemy: multiple related meanings

Often, the relationships are systematic

E.g., building vs. organization

school, university, hospital, church, supermarket

More examples:

- Author (Jane Austen wrote Emma) ↔ Works of Author (I really love Jane Austen)
- Animal (The chicken was domesticated in Asia) ↔ Meat (The chicken was overcooked)
- Tree (Plums have beautiful blossoms) ↔ Fruit (I ate a preserved plum yesterday)

Polysemy: multiple related meanings

Zeugma test

- Which of these flights *serve* breakfast?
- Does Midwest Express *serve* Philadelphia?

Polysemy: multiple related meanings

Zeugma test

- Which of these flights *serve* breakfast?
- Does Midwest Express *serve* Philadelphia?

**Does Midwest Express serve breakfast and San Jose?*

Polysemy: multiple related meanings

Zeugma test

- Which of these flights *serve* breakfast?
- Does Midwest Express *serve* Philadelphia?

**Does Midwest Express serve breakfast and San Jose?*

Combine two separate uses of a lexeme into a single example using conjunction

Since it sounds weird, we say that these are two different senses of *serve*.

Synonymy

Words that have the same meaning in some or all contexts.

- filbert / hazelnut
- couch / sofa
- big / large
- automobile / car
- vomit / throw up
- water / H_2O

Two lexemes are synonyms if they can be substituted for each other in all situations.

Synonymy: A relation between senses

Consider the words *big* and *large*.

Are they synonyms?

- How **big** is that plane?
- Would I be flying on a **large** or small plane?

Synonymy: A relation between senses

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.

Synonymy: A relation between senses

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

Synonyms

Shades of meaning

- What is the cheapest first class *fare*?
- *What is the cheapest first class *price*?

Synonyms

Shades of meaning

- What is the cheapest first class *fare*?
- *What is the cheapest first class *price*?

Collocational constraints

- We frustate 'em and frustate 'em, and pretty soon they make a *big* mistake.
- *We frustate 'em and frustate 'em, and pretty soon they make a *large* mistake.

Antonyms

- Senses that are opposites with respect to one feature of their meaning
- Otherwise, they are similar!
  - dark / light
  - short / long
  - hot / cold
  - up / down
  - in / out

Antonyms

- Senses that are opposites with respect to one feature of their meaning
- Otherwise, they are 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*

Hyponymy and Hypernymy

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*

Hyponymy and Hypernymy

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*

Hypernymy

Conversely

- *vehicle* is a hypernym/superordinate of *car*
- *animal* is a hypernym of *dog*
- *fruit* is a hypernym of *mango*

Hyponymy more formally

Entailment

Sense A is a hyponym of sense B if being an A entails being a B.
Ex: dog, animal

Transitivity

A hypo B and B hypo C entails A hypo C

Meronyms and holonyms

Definition

Meronymy: an asymmetric, transitive relation between senses.
X is a **meronym** of Y if it denotes a part of Y.
The inverse relation is **holonymy**.

meronym	holonym
porch	house
wheel	car
leg	chair
nose	face

Lexical Semantics - WordNet

Pawan Goyal

CSE, IIT Kharagpur

Week 8, Lecture 2

WordNet

https://wordnet.princeton.edu/wordnet/

- A hierarchically organized lexical database
- A machine-readable thesaurus, and aspects of a dictionary
- Versions for other languages are under development

part of speech	no. synsets
noun	82,115
verb	13,767
adjective	18,156
adverb	3,621

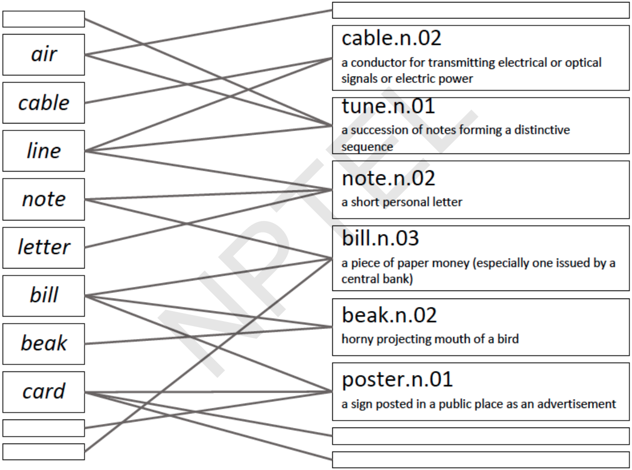
Synsets in WordNet

- A **synset** is a set of synonyms representing a sense
- 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.
- For WordNet, the meaning of this sense of chump is this list.

lemma vs. synsets



All relations in WordNet

```
searchtype is at least one of the following:
        -ants{n|v|a|r}          Antonyms
        -hype{n|v}              Hypernyms
        -hypo{n|v}, -tree{n|v}  Hyponyms & Hyponym Tree
        -entav                  Verb Entailment
        -syns{n|v|a|r}          Synonyms (ordered by estimated frequency)
        -smemn                  Member of Holonyms
        -ssubn                  Substance of Holonyms
        -sprtn                  Part of Holonyms
        -membn                  Has Member Meronyms
        -subsn                  Has Substance Meronyms
        -partn                  Has Part Meronyms
        -meron                  All Meronyms
        -holon                  All Holonyms
        -causv                  Cause to
        -pert{a|r}              Pertainyms
        -attr{n|a}              Attributes
        -deri{n|v}              Derived Forms
        -domn{n|v|a|r}          Domain
        -domt{n|v|a|r}          Domain Terms
        -faml{n|v|a|r}          Familiarity & Polysemy Count
        -framv                  Verb Frames
        -coor{n|v}              Coordinate Terms (sisters)
        -simsv                  Synonyms (grouped by similarity of meaning)
        -hmern                  Hierarchical Meronyms
        -hholn                  Hierarchical Holonyms
        -grep{n|v|a|r}          List of Compound Words
        -over                   Overview of Senses
```

Wordnet noun and verb relations

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

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

```
Synonyms/Hypernyms (Ordered by Estimated Frequency) of noun mouse

4 senses of mouse

Sense 1
mouse
       => rodent, gnawer
           => placental, placental mammal, eutherian, eutherian mammal
               => mammal, mammalian
                   => vertebrate, craniate
                       => chordate
                           => animal, animate being, beast, brute, creature, fauna
                               => organism, being
                                   => living thing, animate thing
                                       => whole, unit
                                           => object, physical object
                                               => physical entity
                                                   => entity


      Sense 4
      mouse, computer mouse
             => electronic device
                 => device
                     => instrumentality, instrumentation
                         => artifact, artefact
                             => whole, unit
                                 => object, physical object
                                     => physical entity
                                         => entity
```

Word Similarity

- Synonymy is a binary relation
  - Two words are either synonymous or not

Word Similarity

- Synonymy is a binary relation
  - Two words are either synonymous or not
- We want a looser metric
  - Word similarity or
  - Word distance

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

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
    - ★ Bank^1 is similar to fund^3
    - ★ Bank^2 is similar to slope^5

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
    - Bank^1 is similar to fund^3
    - Bank^2 is similar to slope^5
- We will compute similarity over both words and senses

Two classes of algorithms

Distributional algorithms

By comparing words based on their distributional context in the corpora

Thesaurus-based algorithms

Based on whether words are “nearby” in WordNet

Thesaurus-based Word Similarity

- We could use anything in the thesaurus:
  - Meronymy, hyponymy, troponymy
  - Glosses and example sentences

Thesaurus-based Word Similarity

- We could use anything in the thesaurus:
  - Meronymy, hyponymy, troponymy
  - Glosses and example sentences
- In practice, "thesaurus-based" methods usually use:
  - the is-a/subsumption/hypernymy hierarchy
  - and sometimes the glosses too

Thesaurus-based Word Similarity

- We could use anything in the thesaurus:
  - Meronymy, hyponymy, troponymy
  - Glosses and example sentences
- In practice, “thesaurus-based” methods usually use:
  - the is-a/subsumption/hypernymy hierarchy
  - and sometimes the glosses too
- Word similarity vs. word relatedness
  - Similar words are near-synonyms
  - Related words could be related any way
    - ★ car, gasoline : related, but nor similar
    - ★ car, bicycle: similar

Path-based similarity

Basic Idea

- Two words are similar if they are nearby in the hypernym graph

Path-based similarity

Basic Idea

- Two words are similar if they are nearby in the hypernym graph
- $pathlen(c_1, c_2)$ = number of edges in shortest path (in hypernym graph) between senses c_1 and c_2

Path-based similarity

Basic Idea

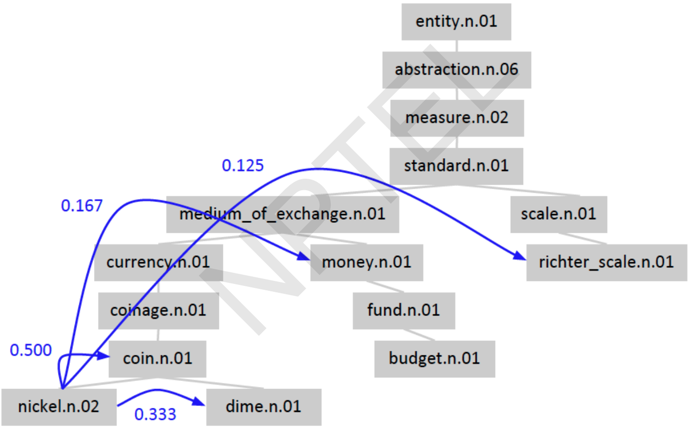
- Two words are similar if they are nearby in the hypernym graph
- $pathlen(c_1, c_2)$ = number of edges in shortest path (in hypernym graph) between senses c_1 and c_2
- $sim_{path}(c_1, c_2) = \frac{1}{1+pathlen(c_1, c_2)}$

Path-based similarity

Basic Idea

- Two words are similar if they are nearby in the hypernym graph
- $pathlen(c_1, c_2)$ = number of edges in shortest path (in hypernym graph) between senses c_1 and c_2
- $sim_{path}(c_1, c_2) = \frac{1}{1+pathlen(c_1,c_2)}$
- $sim(w_1, w_2) = max_{c_1 \in senses(w_1), c_2 \in senses(w_2)} sim(c_1, c_2)$

Shortest path in the hierarchy



Leacock-Chodorow (L-C) Similarity

L-C similarity

$$sim_{LC}(c_1, c_2) = -log(pathlen(c_1, c_2)/2d)$$

d: maximum depth of the hierarchy

Leacock-Chodorow (L-C) Similarity

L-C similarity

$$sim_{LC}(c_1, c_2) = -log(pathlen(c_1, c_2)/2d)$$

d: maximum depth of the hierarchy

Problems with L-C similarity

- Assumes each edge represents a uniform distance
- 'nickel-money' seems closer than 'nickel-standard'

Leacock-Chodorow (L-C) Similarity

L-C similarity

$$sim_{LC}(c_1, c_2) = -log(pathlen(c_1, c_2)/2d)$$

d: maximum depth of the hierarchy

Problems with L-C similarity

- Assumes each edge represents a uniform distance
- 'nickel-money' seems closer than 'nickel-standard'
- We want a metric which lets us assign different "lengths" to different edges - but how?

Concept probability models

Cencept probabilities

- For each concept (synset) c, let $P(c)$ be the probability that a randomly selected word in a corpus is an instance (hyponym) of c

Concept probability models

Cencept probabilities

- For each concept (synset) c, let $P(c)$ be the probability that a randomly selected word in a corpus is an instance (hyponym) of c
- $P(ROOT) = 1$
- The lower a node in the hierarchy, the lower its probability

Concept probability models

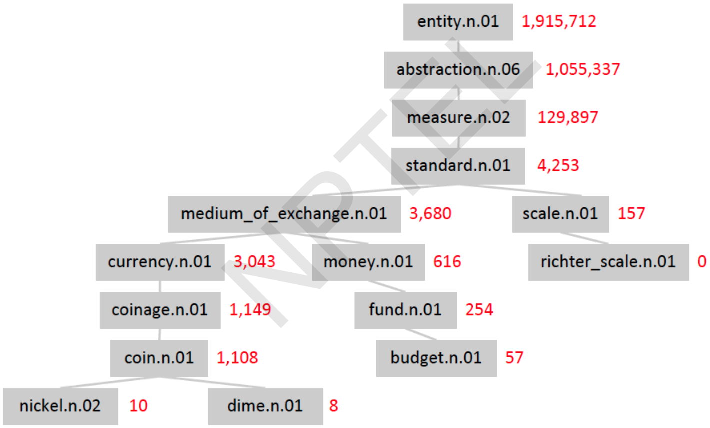
Cencept probabilities

- For each concept (synset) c, let $P(c)$ be the probability that a randomly selected word in a corpus is an instance (hyponym) of c
- $P(ROOT) = 1$
- The lower a node in the hierarchy, the lower its probability

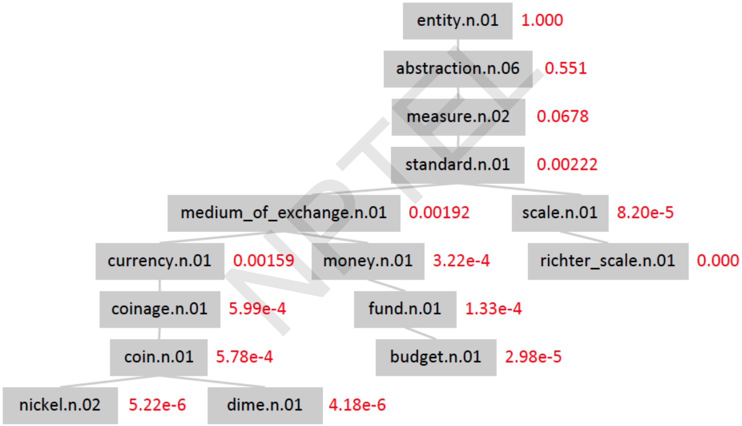
Estimating concept probabilities

- Train by counting "concept activations" in a corpus
- Each occurrence of *dime* also increments counts for *coin, currency, standard*, etc.

Example : concept count



Example : concept probabilities

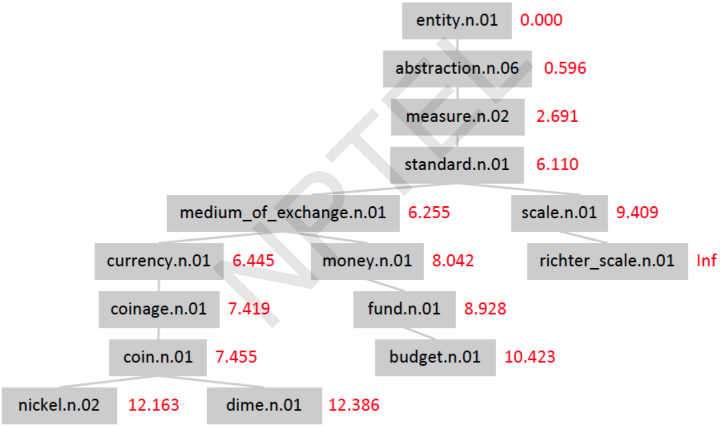


Information content

Information content

- Information content: $IC(c) = -logP(c)$
- Lowest common subsumer : $LCS(c_1, c_2)$: the lowest node in the hierarchy that subsumes (is a hypernym of) both c_1 and c_2
- We are now ready to see how to use information content (IC) as a similarity metric.

Example : Information content

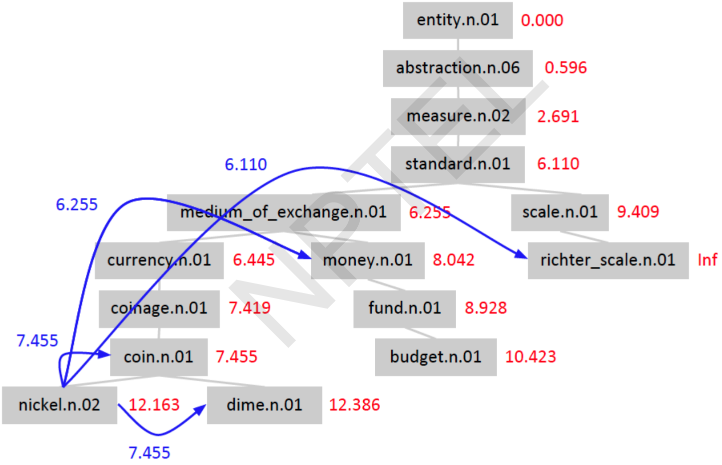


Resnik Similarity

Resnik Similarity

- Intuition: how similar two words are depends on how much they have in common
- It measures the commonality by the information content of the lowest common subsumer
- $sim_{resnik}(c_1, c_2) = IC(LCS(c_1, c_2)) = -logP(LCS(c_1, c_2))$

Example: Resnik similarity



Lin similarity

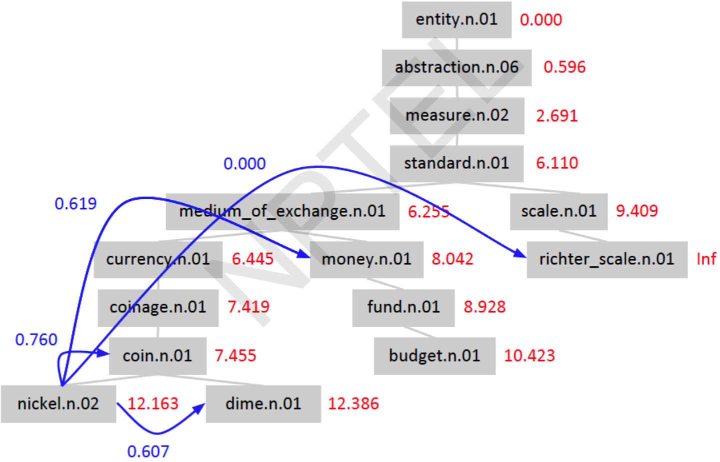
Proportion of shared information

- It's not just about commonalities - it's also about differences!
- **Resnik:** The more information content they share, the more similar they are
- **Lin:** The more information content they don't share, the less similar they are
- Not the *absolute* quantity of shared information but the *proportion* of shared information

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

The information content common to c_1 and c_2, normalized by their average information content.

Example: Lin similarity



Jiang-Conrath distance

JC similarity

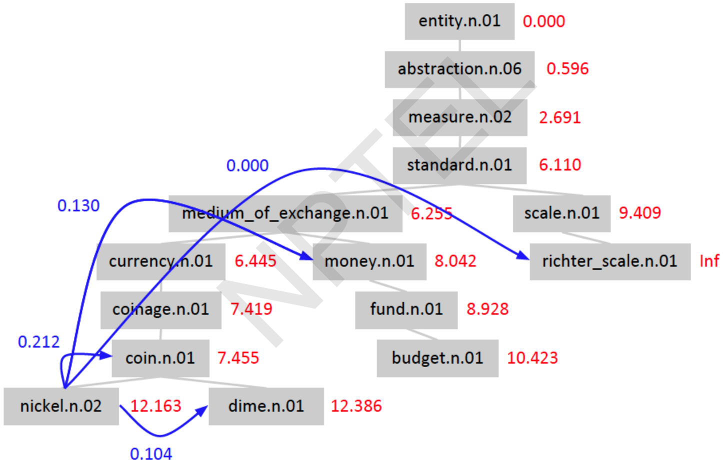
We can use IC to assign lengths to graph edges:

$$dist_{JC}(c, hypernym(c)) = IC(c) - IC(hypernym(c))$$

$$\begin{aligned} dist_{JC}(c_1, c_2) &= dist_{JC}(c_1, LCS(c_1, c_2)) + dist_{JC}(c_2, LCS(c_1, c_2)) \\ &= IC(c_1) - IC(LCS(c_1, c_2)) + IC(c_2) - IC(LCS(c_1, c_2)) \\ &= IC(c_1) + IC(c_2) - 2 \times IC(LCS(c_1, c_2)) \end{aligned}$$

$$sim_{JC}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \times IC(LCS(c_1, c_2))}$$

Example: Jiang-Conrath distance



The (extended) Lesk Algorithm

- 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*

The (extended) Lesk Algorithm

- 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

The (extended) Lesk Algorithm

- 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** $\rightarrow 1+4=5$

Problem in mapping words to wordnet senses

I saw a man who is 98 years old and can still walk and tell jokes

Ambiguity is rampant!

Word Sense Disambiguation - I

Pawan Goyal

CSE, IIT Kharagpur

Week 8, Lecture 3

Word Sense Disambiguation (WSD)

Sense ambiguity

- Many words have several meanings or senses
- The meaning of **bass** depends on the context
- Are we talking about music, or fish?
  - 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.
  - And it all started when fishermen decided the striped **bass** in Lake Mead were too skinny.

Word Sense Disambiguation (WSD)

Sense ambiguity

- Many words have several meanings or senses
- The meaning of **bass** depends on the context
- Are we talking about music, or fish?
  - 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.
  - And it all started when fishermen decided the striped **bass** in Lake Mead were too skinny.

Disambiguation

- The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word.

Word Sense Disambiguation (WSD)

Sense ambiguity

- Many words have several meanings or senses
- The meaning of **bass** depends on the context
- Are we talking about music, or fish?
  - 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.
  - And it all started when fishermen decided the striped **bass** in Lake Mead were too skinny.

Disambiguation

- The task of disambiguation is to determine which of the senses of an ambiguous word is invoked in a particular use of the word.
- This is done by looking at the context of the word's use.

Algorithms

- Knowledge Based Approaches
  - Overlap Based Approaches
- Machine Learning Based Approaches
  - Supervised Approaches
  - Semi-supervised Algorithms
  - Unsupervised Algorithms
- Hybrid Approaches

Knowledge Based Approaches

Overlap Based Approaches

- Require a **Machine Readable Dictionary** (MRD).
- Find the overlap between the features of different senses of an ambiguous word (**sense bag**) and the features of the words in its context (**context bag**).
- The features could be sense definitions, example sentences, hypernyms etc.
- The features could also be given weights.
- The sense which has the maximum overlap is selected as the contextually appropriate sense.

Lesk's Algorithm

Sense Bag: contains the words in the definition of a candidate sense of the ambiguous word.

Lesk's Algorithm

Sense Bag: contains the words in the definition of a candidate sense of the ambiguous word.

Context Bag: contains the words in the definition of each sense of each context word.

Lesk's Algorithm

Sense Bag: contains the words in the definition of a candidate sense of the ambiguous word.
Context Bag: contains the words in the definition of each sense of each context word.

*On burning **coal** we get ash.*

Lesk's Algorithm

Sense Bag: contains the words in the definition of a candidate sense of the ambiguous word.
Context Bag: contains the words in the definition of each sense of each context word.

*On burning **coal** we get ash.*

Ash

- Sense 1
  Trees of the olive family with pinnate leaves, thin furrowed bark and gray branches.
- Sense 2
  The ***solid*** residue left when ***combustible*** material is thoroughly ***burn***ed or oxidized.
- Sense 3
  To convert into ash

Coal

- Sense 1
  A piece of glowing carbon or ***burn***t wood.
- Sense 2
  charcoal.
- Sense 3
  A black ***solid combustible*** substance formed by the partial decomposition of vegetable matter without free access to air and under the influence of moisture and often increased pressure and temperature that is widely used as a fuel for ***burn***ing

In this case Sense 2 of ash would be the winner sense.

Walker's Algorithm

- A Thesaurus Based approach

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words.

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*
  - E.g. The money in this bank fetches an interest of 8% per annum

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*
    - E.g. The money in this bank fetches an interest of 8% per annum
    - Target word: *bank*
    - Clue words from the context: *money, interest, annum, fetch*

Walker's Algorithm

- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word matches that of the sense.*
  - E.g. The money in this bank fetches an interest of 8% per annum
  - Target word: *bank*
  - Clue words from the context: *money, interest, annum, fetch*

	Sense1: Finance	Sense2: Location
Money	+1	0
Interest	+1	0
Fetch	0	0
Annum	+1	0
Total	3	0

Context words add 1 to the sense when the topic of the word matches that of the sense

WSD Using Random Walk Algorithm

The **church bells** no longer **rung** on **Sundays**.

church

1: one of the groups of Christians who have their own beliefs and forms of worship
2: a place for public (especially Christian) worship
3: a service conducted in a church

bell

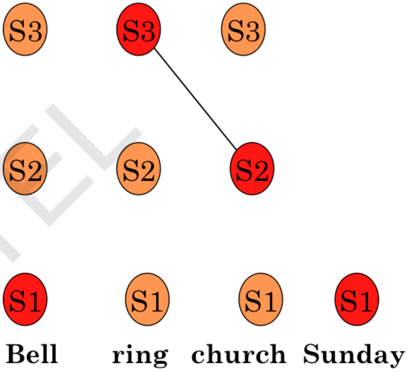
1: a hollow device made of metal that makes a ringing sound when struck
2: a push button at an outer door that gives a ringing or buzzing signal when pushed
3: the sound of a bell

ring

1: make a ringing sound
2: ring or echo with sound
3: make (bells) ring, often for the purposes of musical edification

Sunday

1: first day of the week; observed as a day of rest and worship by most Christians



Step 1: Add a vertex for each possible sense of each word in the text.

WSD Using Random Walk Algorithm

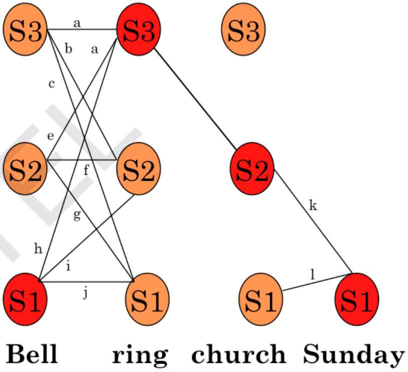
The **church bells** no longer **rung** on **Sundays**.

church
1: one of the groups of Christians who have their own beliefs and forms of worship
2: a place for public (especially Christian) worship
3: a service conducted in a church

bell
1: a hollow device made of metal that makes a ringing sound when struck
2: a push button at an outer door that gives a ringing or buzzing signal when pushed
3: the sound of a bell

ring
1: make a ringing sound
2: ring or echo with sound
3: make (bells) ring, often for the purposes of musical edification

Sunday
1: first day of the week; observed as a day of rest and worship by most Christians



Step 2: Add weighted edges using definition based semantic similarity (Lesk's method).

WSD Using Random Walk Algorithm

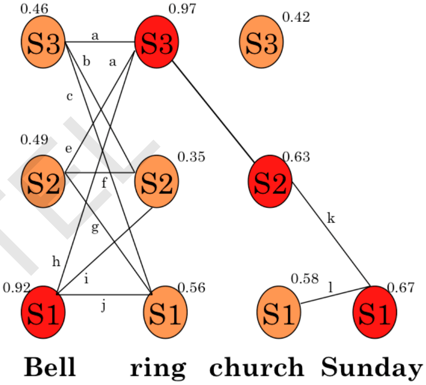
The **church bells** no longer **rung** on **Sundays**.

church
1: one of the groups of Christians who have their own beliefs and forms of worship
2: a place for public (especially Christian) worship
3: a service conducted in a church

bell
1: a hollow device made of metal that makes a ringing sound when struck
2: a push button at an outer door that gives a ringing or buzzing signal when pushed
3: the sound of a bell

ring
1: make a ringing sound
2: ring or echo with sound
3: make (bells) ring, often for the purposes of musical edification

Sunday
1: first day of the week; observed as a day of rest and worship by most Christians



Step 3: Apply graph based ranking algorithm to find score of each vertex (i.e. for each word sense).

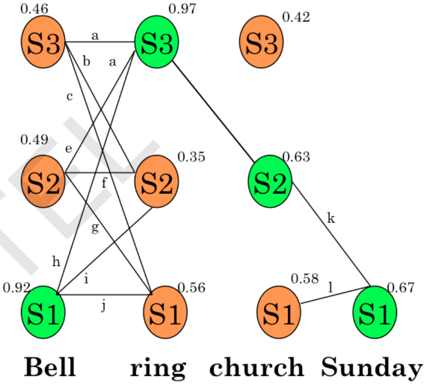
The **church bells** no longer **rung** on **Sundays**.

church
- 1: one of the groups of Christians who have their own beliefs and forms of worship
- 2: a place for public (especially Christian) worship
- 3: a service conducted in a church

bell
- 1: a hollow device made of metal that makes a ringing sound when struck
- 2: a push button at an outer door that gives a ringing or buzzing signal when pushed
- 3: the sound of a bell

ring
- 1: make a ringing sound
- 2: ring or echo with sound
- 3: make (bells) ring, often for the purposes of musical edification

Sunday
- 1: first day of the week; observed as a day of rest and worship by most Christians



Step 4: Select the vertex (sense) which has the highest score.

Naïve Bayes for WSD

- A Naïve Bayes classifier chooses the most likely sense for a word given the features of the context:

$$\hat{s} = \arg\max_{s \in S} P(s|f)$$

Naïve Bayes for WSD

- A Naïve Bayes classifier chooses the most likely sense for a word given the features of the context:

$$\hat{s} = \arg\max_{s \in S} P(s|f)$$

- Using Bayes' law, this can be expressed as:

$$\hat{s} = \arg\max_{s \in S} \frac{P(s)P(f|s)}{P(f)}$$

$$= \arg\max_{s \in S} P(s)P(f|s)$$

Naïve Bayes for WSD

- A Naïve Bayes classifier chooses the most likely sense for a word given the features of the context:

$$\hat{s} = \arg\max_{s \in S} P(s|f)$$

- Using Bayes' law, this can be expressed as:

$$\hat{s} = \arg\max_{s \in S} \frac{P(s)P(f|s)}{P(f)}$$

$$= \arg\max_{s \in S} P(s)P(f|s)$$

- The 'Naïve' assumption: all the features are conditionally independent, given the sense':

$$\hat{s} = \arg\max_{s \in S} P(s) \prod_{j=1}^{n} P(f_j|s)$$

Training for Naïve Bayes

- 'f' is a feature vector consisting of:
  - POS of w
  - Semantic and Syntactic features of w
  - Collocation vector (set of words around it) $\rightarrow$ next word (+1), +2, -1, -2 and their POS's
  - Co-occurrence vector

Training for Naïve Bayes

- 'f' is a feature vector consisting of:
  - POS of w
  - Semantic and Syntactic features of w
  - Collocation vector (set of words around it) $\rightarrow$ next word (+1), +2, -1, -2 and their POS's
  - Co-occurrence vector
- Set parameters of Naïve Bayes using maximum likelihood estimation (MLE) from training data

$$P(s_i) = \frac{count(s_i, w_j)}{count(w_j)}$$

$$P(f_j|s_i) = \frac{count(f_j, s_i)}{count(s_i)}$$

Decision List Algorithm

- Based on ‘One sense per collocation’ property
  - Nearby words provide strong and consistent clues as to the sense of a target word
- Collect a large set of collocations for the ambiguous word
- Calculate word-sense probability distributions for all such collocations

Decision List Algorithm

- Based on ‘One sense per collocation’ property
  - Nearby words provide strong and consistent clues as to the sense of a target word
- Collect a large set of collocations for the ambiguous word
- Calculate word-sense probability distributions for all such collocations
- Calculate the log-likelihood ratio

$$log(\frac{P(Sense-A|Collocation_i)}{P(Sense-B|Collocation_i)})$$

- Higher log-likelihood $\Rightarrow$ more predictive evidence

Decision List Algorithm

- Based on 'One sense per collocation' property
  - Nearby words provide strong and consistent clues as to the sense of a target word
- Collect a large set of collocations for the ambiguous word
- Calculate word-sense probability distributions for all such collocations
- Calculate the log-likelihood ratio

$$log\left(\frac{P(Sense-A|Collocation_i)}{P(Sense-B|Collocation_i)}\right)$$

- Higher log-likelihood $\Rightarrow$ more predictive evidence
- Collocations are ordered in a decision list, with most predictive collocations ranked highest

Decision List Algorithm

Training Data

Sense	Training Examples (Keyword in Context)
A	used to strain microscopic *plant* life from the ...
A	... zonal distribution of *plant* life . ...
A	close-up studies of *plant* life and natural ...
A	too rapid growth of aquatic *plant* life in water ...
A	... the proliferation of *plant* and animal life ...
A	establishment phase of the *plant* virus life cycle ...
A	... ...
B	... ...
B	computer manufacturing *plant* and adjacent ...
B	discovered at a St. Louis *plant* manufacturing
B	... copper manufacturing *plant* found that they
B	copper wire manufacturing *plant* , for example ...
B	's cement manufacturing *plant* in Alpena ...
B	polystyrene manufacturing *plant* at its Dow ...
B	company manufacturing *plant* is in Orlando ...

Resultant Decision List

Final decision list for *plant* (abbreviated)

LogL	Collocation	Sense
10.12	*plant* growth	⇒ A
9.68	car (within $\pm k$ words)	⇒ B
9.64	*plant* height	⇒ A
9.61	union (within $\pm k$ words)	⇒ B
9.54	equipment (within $\pm k$ words)	⇒ B
9.51	assembly *plant*	⇒ B
9.50	nuclear *plant*	⇒ B
9.31	flower (within $\pm k$ words)	⇒ A
9.24	job (within $\pm k$ words)	⇒ B
9.03	fruit (within $\pm k$ words)	⇒ A
9.02	*plant* species	⇒ A
...	...	

Decision List Algorithm

Training Data

Sense	Training Examples (Keyword in Context)
A	used to strain microscopic *plant* life from the ...
A	... zonal distribution of *plant* life . ...
A	close-up studies of *plant* life and natural ...
A	too rapid growth of aquatic *plant* life in water ...
A	... the proliferation of *plant* and animal life ...
A	establishment phase of the *plant* virus life cycle ...
A	... ...
B	... ...
B	computer manufacturing *plant* and adjacent ...
B	discovered at a St. Louis *plant* manufacturing
B	... copper manufacturing *plant* found that they
B	copper wire manufacturing *plant* , for example ...
B	's cement manufacturing *plant* in Alpena ...
B	polystyrene manufacturing *plant* at its Dow ...
B	company manufacturing *plant* is in Orlando ...

Resultant Decision List

Final decision list for *plant* (abbreviated)

LogL	Collocation	Sense
10.12	*plant* growth	⇒ A
9.68	car (within $\pm k$ words)	⇒ B
9.64	*plant* height	⇒ A
9.61	union (within $\pm k$ words)	⇒ B
9.54	equipment (within $\pm k$ words)	⇒ B
9.51	assembly *plant*	⇒ B
9.50	nuclear *plant*	⇒ B
9.31	flower (within $\pm k$ words)	⇒ A
9.24	job (within $\pm k$ words)	⇒ B
9.03	fruit (within $\pm k$ words)	⇒ A
9.02	*plant* species	⇒ A
...	...	

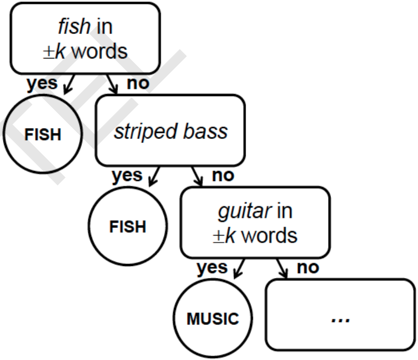
Classification of a test sentence is based on the highest ranking collocation, found in the test sentences.

plucking flowers affects plant growth.

Decision List: Example

Example: discriminating between bass (fish) and bass (music):

Context	Sense
fish in $\pm k$ words	FISH
striped bass	FISH
guitar in $\pm k$ words	MUSIC
bass player	MUSIC
piano in $\pm k$ words	MUSIC
sea bass	FISH
play bass	MUSIC
river in $\pm k$ words	FISH
on bass	MUSIC
bass are	FISH



Word Sense Disambiguation - II

Pawan Goyal

CSE, IIT Kharagpur

Week 8, Lecture 4

Minimally Supervised WSD - Yarowsky

- Annotations are expensive!
- "Bootstrapping" or co-training
  - Start with (small) seed, learn decision list
  - Use decision list to label rest of corpus
  - Retain 'confident' labels, treat as annotated data to learn new decision list
  - Repeat …

Minimally Supervised WSD - Yarowsky

- Annotations are expensive!
- "Bootstrapping" or co-training
  - Start with (small) seed, learn decision list
  - Use decision list to label rest of corpus
  - Retain 'confident' labels, treat as annotated data to learn new decision list
  - Repeat ...
- Heuristics (derived from observation):
  - One sense per discourse
  - One sense per collocation

More about heuristics

One Sense per Discourse

- A word tends to preserve its meaning across all its occurrences in a given discourse

More about heuristics

One Sense per Discourse

- A word tends to preserve its meaning across all its occurrences in a given discourse

One Sense per Collocation

- A word tends to preserve its meaning when used in the same collocation
  - Strong for adjacent collocations
  - Weaker as the distance between the words increases

Yarowsky's Method

Example

- Disambiguating plant (industrial sense) vs. plant (living thing sense)
- Think of seed features for each sense
  - Industrial sense: co-occurring with 'manufacturing'
  - Living thing sense: co-occurring with 'life'
- Use 'one sense per collocation' to build initial decision list classifier
- Treat results (having high probability) as annotated data, train new decision list classifier, iterate

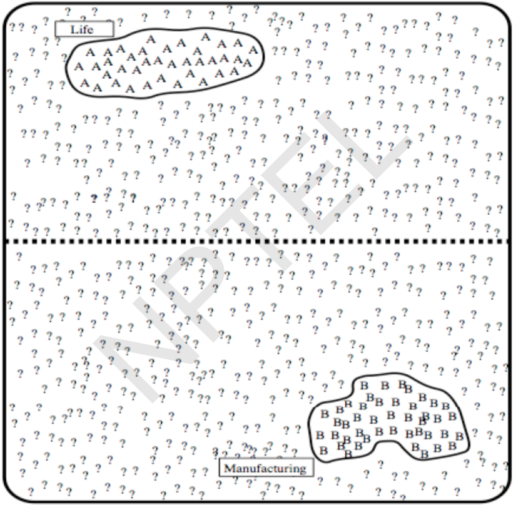
Yarowsky's Method: Example

used to strain microscopic plant life from the
zonal distribution of plant life .
close-up studies of plant life and natural
too rapid growth of aquatic plant life in water
the proliferation of plant and animal life
establishment phase of the plant virus life cycle
that divide life into plant and animal kingdom
many dangers to plant and animal life
mammals . Animal and plant life are delicately

automated manufacturing plant in Fremont
vast manufacturing plant and distribution
chemical manufacturing plant , producing viscose
keep a manufacturing plant profitable without
computer manufacturing plant and adjacent
discovered at a St. Louis plant manufacturing
copper manufacturing plant found that they
copper wire manufacturing plant , for example
s cement manufacturing plant in Alpena

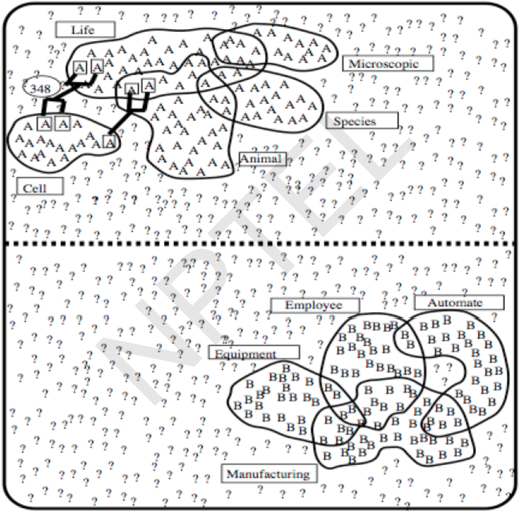
vinyl chloride monomer plant , which is
molecules found in plant and animal tissue
Nissan car and truck plant in Japan is
and Golgi apparatus of plant and animal cells
union responses to plant closures .
cell types found in the plant kingdom are
company said the plant is still operating
Although thousands of plant and animal species
animal rather than plant tissues can be

Yarowsky's Method: Example



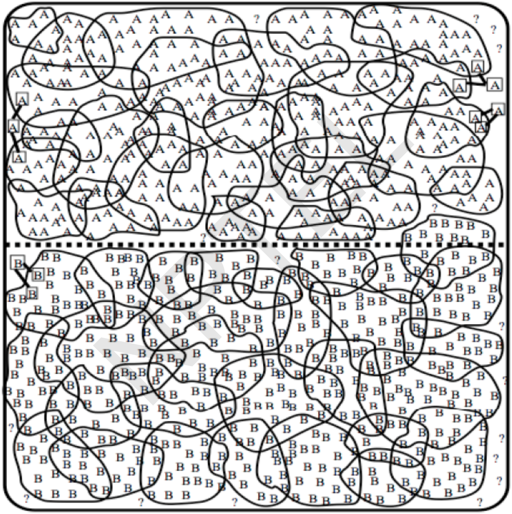
Initial state after use of seed rules

Yarowsky's Method: Example



Intermediate state

Yarowsky's Method: Example



Final state

Yarowsky's Method

Termination

- Stop when
  - Error on training data is less than a threshold
  - No more training data is covered
- Use final decision list for WSD

Advantages

- Accuracy is about as good as a supervised algorithm
- Bootstrapping: far less manual effort

HyperLex

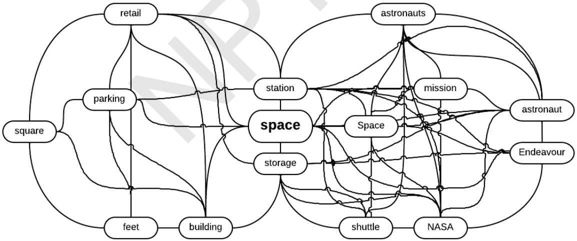
Key Idea: Word Sense Induction

- Instead of using “dictionary defined senses”, extract the “senses from the corpus” itself
- These “corpus senses” or “uses” correspond to clusters of similar contexts for a word.

HyperLex

Key Idea: Word Sense Induction

- Instead of using "dictionary defined senses", extract the "senses from the corpus" itself
- These "corpus senses" or "uses" correspond to clusters of similar contexts for a word.



HyperLex

Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.

HyperLex

Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.
- **Step 2:** Arrange nodes in G in decreasing order of degree.

HyperLex

Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.
- **Step 2:** Arrange nodes in G in decreasing order of degree.
- **Step 3:** Select the node from G which has the highest degree. This node will be the hub of the first high density component.

HyperLex

Detecting Root Hubs

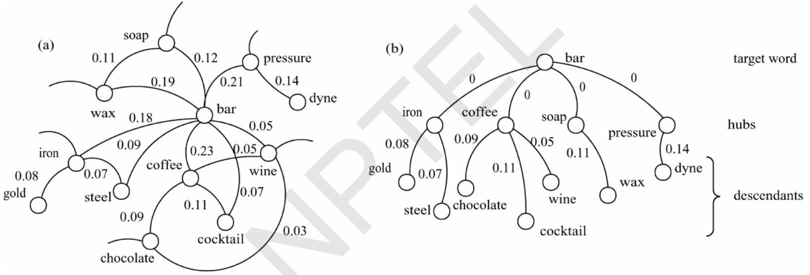
- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.
- **Step 2:** Arrange nodes in G in decreasing order of degree.
- **Step 3:** Select the node from G which has the highest degree. This node will be the hub of the first high density component.
- **Step 4:** Delete this hub and all its neighbors from G.

HyperLex

Detecting Root Hubs

- Different uses of a target word form highly interconnected bundles (or high density components)
- In each high density component one of the nodes (hub) has a higher degree than the others.
- **Step 1:** Construct co-occurrence graph, G.
- **Step 2:** Arrange nodes in G in decreasing order of degree.
- **Step 3:** Select the node from G which has the highest degree. This node will be the hub of the first high density component.
- **Step 4:** Delete this hub and all its neighbors from G.
- **Step 5:** Repeat Step 3 and 4 to detect the hubs of other high density components

HyperLex: Detecting Root Hubs



Delineating Components

- Attach each node to the root hub closest to it.
- The distance between two nodes is measured as the smallest sum of weights of the edges on the paths linking them.

Delineating Components

- Attach each node to the root hub closest to it.
- The distance between two nodes is measured as the smallest sum of weights of the edges on the paths linking them.

Computing distance between two nodes w_i and w_j

$$w_{ij} = 1 - max\{P(w_i|w_j), P(w_j|w_i)\}$$

where $P(w_i|w_j) = \frac{freq_{ij}}{freq_j}$

Disambiguation

- Let $W = (w_1, w_2, \ldots, w_i, \ldots, w_n)$ be a context in which w_i is an instance of our target word.
- Let w_i has k hubs in its minimum spanning tree

Disambiguation

- Let $W = (w_1, w_2, \ldots, w_i, \ldots, w_n)$ be a context in which w_i is an instance of our target word.
- Let w_i has k hubs in its minimum spanning tree
- A score vector s is associated with each $w_j \in W (j \neq i)$, such that s_k represents the contribution of the kth hub as:

$$\begin{aligned} s_k &= \frac{1}{1 + d(h_k, w_j)} \text{if } h_k \text{ is an ancestor of } w_j \\ s_i &= 0 \text{ otherwise.} \end{aligned}$$

- All score vectors associated with all $w_j \in W (j \neq i)$ are summed up
- The hub which receives the maximum score is chosen as the most appropriate sense

Novel Word Sense Detection

Pawan Goyal

CSE, IIT Kharagpur

Week 8, Lecture 5

Classical sense

sick *adjective* \'sik\

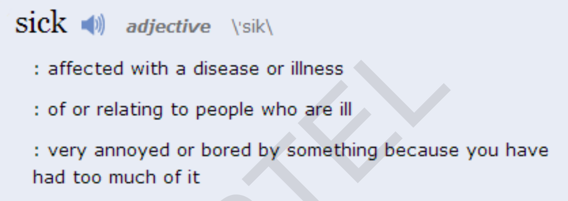
: affected with a disease or illness

: of or relating to people who are ill

: very annoyed or bored by something because you have had too much of it [1]

[1] http://www.merriam-webster.com/

Classical sense


sick adjective \'sik\

: affected with a disease or illness

: of or relating to people who are ill

: very annoyed or bored by something because you have had too much of it

[1]

Novel sense

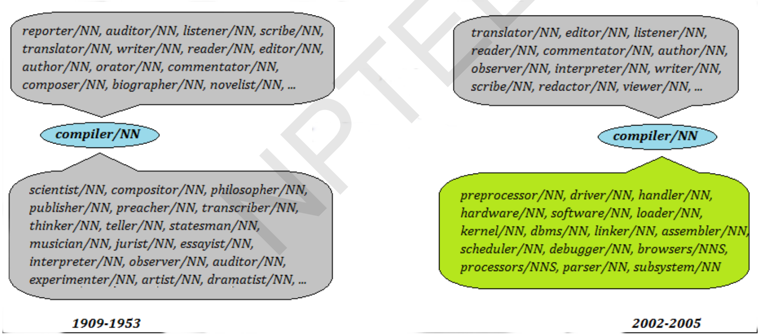
[1] http://www.merriam-webster.com/

Comparing sense clusters

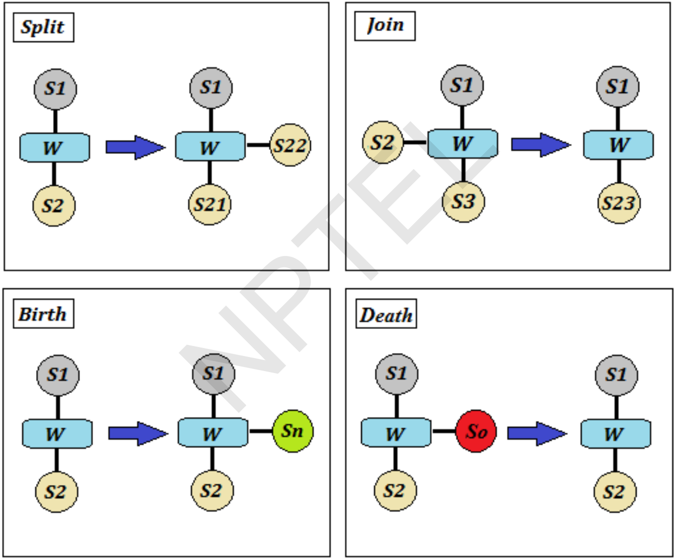
- If a word undergoes sense change, this can be detected by comparing the sense clusters obtained from two different time periods

Comparing sense clusters

- If a word undergoes sense change, this can be detected by comparing the sense clusters obtained from two different time periods



Split, join, birth and death



A real example of birth