Natural Language Processing

Chapter 6 Semantic Analysis

Dr Raymond Lee
Associate Professor, DST
BNU-HKBU United International College

Semantic Analysis

1. Introduction
   - What is Semantic Analysis?
   - How important Semantic Analysis is?
   - How Good Human in Semantic Analysis?
2. Lexical vs Compositional Semantic Analysis
3. Word Senses & Relations
   - Six types of commonly used lexical semantics
4. Word Sense Disambiguation
   - What is Word Sense Disambiguation?
   - Difficulties in WSD
5. WordNet and Online Thesauri
6. Word Similarity & Thesaurus Methods
   - Path-based Similarity
7. Distributed Similarity
   - Problems with thesaurus-based meaning
   - Term-Document Matrix
   - Pointwise Mutual Information (PMI) and PPMI model
   - Similarity measurement
8. Summary

Semantic Analysis
Part 1 – Introduction

What is Semantic Analysis

What is Semantic Analysis?

- Semantic analysis is the process of taking in some linguistic input and producing a meaning representation for it.
- In terms of NLP, semantic analysis is the process of drawing meaning from text and utterance. It allows computers to understand and interpret sentences, paragraphs, or whole documents, by analyzing their grammatical structure, and identifying relationships between individual words in a particular context via written texts and verbal communication.
- The truth is: It's an essential sub-task of Natural Language Processing (NLP) and the driving force behind machine learning tools like text analysis, search engines, and chatbots – the golden grail.
- Semantic analysis-driven tools can help companies automatically extract meaningful information from unstructured data, such as emails, support tickets, and customer feedback.
- There are many ways of doing this, ranging from completely ad hoc domain specific methods to more theoretically founded by not quite useful methods.
- Different methods make more or less (or no) use of syntax.
- However, due to the vast complexity and subjectivity involved in human language, interpreting it is quite a complicated task for machines.
- Semantic Analysis of Natural Language captures the meaning of the given text while taking into account context, logical structuring of sentences and grammar roles.
- From a computer science perspective, semantics are "tokens" that provide context to language—clues to the meaning of words and those words' relationships with other words. From these "tokens" the expectation is for the machine to look beyond the individual words used to identify the true meaning of what's being said as a whole.
- Successful semantic analysis requires a machine to look at MASSIVE data sets, and in analyzing those sets form accurate assumptions that account for context. Put another way, it's about asking a machine to make meaningful cognitive leaps using data-based measures (frequency, location, etc.).
- For example: Having a computer make the connection that a "Great Panda" as a "special breed of animal with a funny look and appearance that can only be found in China" from semantic information provided through large data sets instead of just name of representation.

How important Semantic Analysis is?

How important Semantic Analysis is?

- Why do we care if a computer really knows that a “Great Panda” is instead of just a name in various languages? …
- Because if it knows what a “Great Panda” is instead of just a name, it will know all the relation knowledge and information about “Great Panda” such as its look and appearance, how different it is as compared with other panda species, their history of in terms of evolution and all the related news and information about it.
- In terms of NLP, in order to make sure content is relevant to the user, two components are required:
  - An understanding of the user
  - An understanding of the content and context
- The problem with establishing relationships between pieces of content (and context) is that most “scraping” or “data-capture” technology doesn’t understand the contextual language within a document very well.
- There may be simplistic levels of machine learning involved, but those levels rely heavily on provided tags and a cursory understanding of the individual words on the page…leaving the door wide open for improvement.
- “If we can understand the content and the user behavior at a deep, semantic level, we can deliver more relevant content and thereby create a more resonant user experience.”
- In fact, many automatic classification systems out there today use a pure bag of words approach for finding relevant features that determine the meaning of a document. Few are using correlation and collocation – to account for the fact that words have a different meaning based on their context. None of them is using full semantic analysis of the meaning of words. But this is very much needed to be able to accurately classify a document. The main reason is that (especially English) language is so ambiguous. English nouns have on average 5-8 close synonyms.
- For example: there are words – example “run” – that have more than 100 common meanings (running towards the finish line, run to a meeting, run a company, the machine is running, tears ran down her face, ran for president, run you a couple thousand dollars, etc.). Now if you use a simple bag of words as features the software will never be able to make a clear distinction between an important fact and irrelevant information. Hence the classification result is also ambiguous and not very precise.

How Good Human in Semantic Analysis?

How good human in Semantic Analysis?

- In fact, we're so good at it's generally an unconscious exercise, like breathing…we just do it without thinking about it.
- For example the meaning of "apple: Half a century before, when we talk about the concept of "apple", 99% we are talking about the apple as a kind of fruit we are eating in daily basis. But now, in most of the text and conversation in human life, when we talk about "apple", over 90% we are talking about the brandname "Apple", which almost dominate the market of cellphone and computer nowadays.
- In other words, we human is excellent in "extracting"
  - Context surrounding words
  - Phrases
  - Objects
  - Scenarios

  To built-up the overall context and meaning of words/phrase in a text or conversation and pull out the relevant information.
- Compare that information against prior experience and even the World Knowledge and Common Sense.
- And then use the output of that analysis to predict an outcome with incredible accuracy.
- With the evolution of Artificial Intelligence, machine learning, and natural language processing has changed all that. Advancing algorithms, increasingly powerful computers, and data-based practice have made machine-driven semantic analysis a real thing with a number of real world applications.
- Machine-driven semantic analysis can…
  - Discover the meaning of colloquial speech in online posts
  - Find an answer to a question without having to ask a human
  - Extract relevant and useful information from large bodies of unstructured data
- The truth is: Semantic Analysis is related to "make sense of everything", started with our words and language.

Semantic Analysis
Part 2 – Lexical vs Compositional Semantic Analysis

Lexical Semantic Analysis

What is Lexical Semantic Analysis?

- In linguistics, as a subfield of linguistic semantics, Lexical Semantic Analysis is the study of word meanings.
- It includes the study of how words structure their meaning, how they act in grammar and compositionality, and the relationships between the distinct senses and uses of a word.
- Lexical units include the catalogue of words in a language, the lexicon.
- The units of analysis in lexical semantics are lexical units which includes not only words, but also sub-words, affixes (sub-units), compound words and phrases also, which are collectively called lexical items.
- In other words, we can say that lexical semantics is the relationship between lexical items, meaning of sentences and syntax of sentence.
- Lexical semantics looks at how the meaning of the lexical units correlates with the structure of the language or syntax.
- The study of lexical semantic analysis include:
    - the classification and decomposition of lexical terms and tokens
    - the investigation of the differences and similarities in lexical semantic structure cross-linguistically
    - the relationship of lexical meaning to sentence meaning and syntactic structure.
- Lexical relation in terms of lexical semantic involves the study of how meaning or words are related to each other in the lexical level.
- Lexical relation in lexical analysis include:
    - Homonymy
    - Polysemy
    - Metonymy
    - Synonyms
    - Antonyms
    - Hyponymy and Hypernymy
- Which will be discussed in detail in the next section – Word Sense and Relation.

Compositional Semantic Analysis

What is Compositional Semantic Analysis?

- Compositionality is a concept in the philosophy of language.
- In linguistic, a sentence is compositional if the meaning of every complex expression E in that system depends not only on the meaning of every single word, but also the syntactic structure and arrangement of different word (part of words) within the sentence (utterance).
- In compositional languages, the meaning of a sentence S directly depends only on the meanings of the words composing S, and the way those words are syntactically related to one another.
- In other words, compositional semantics is the study of the meaning of linguistic sentences with syntactic structure instead of individual words.
- The concept behind is: Words contribute to the meaning of sentences but don't have a meaning by themselves
- For example:

  [6.1] Michael likes Mary => likes(Michael, Mary) vs

  [6.2] Mary likes Michael => likes(Mary, Michael)

  Although the individual meaning of every single word in the sentence are the same, due to the different in the arrangement of words inside the sentence, their meanings and predicate logic are totally different.
- In reality, Compositional Semantics is the study of the meaning of complex linguistic units such as sentences, paragraphs, or documents.
- We need to be able to convert the information expressed in linguistic units into some exploitable (formal) representation.
- For a formal representation, to be exploitable means, among others, that:
  - It can be modified through various transformations, also expressed in linguistic terms;
  - It can the subject of various analysis (e.g. counting some of its constituents), also expressed in linguistic terms.
- Symbolic representations:
  - various formal logics: the meaning is expressed as a logical formula that can then be manipulated through various inferential mechanisms;
  - various graph based representations: the meaning is expressed as a graph that can then be manipulated through various graph transformations;
- Vectorial representations:
  - typically approaches based on "distributional semantics" (e.g. Word embeddings): the meaning is represented as a vector in a (usually high dimension) vector space and can then be manipulated through vector based operations such as weighted sums, projections, etc.
- Currently, only vectorial representations can be deployed at a large scale because:
  - It is extremely difficult to guarantee the consistency of large sets of logical propositions derived from textual input, which often makes the inferential mechanisms very hard to use;
  - there isn't yet a consensus neither on which are the most suitable graph based representations such as semantic nets for expressing the meaning of linguistic entities, nor on which are the proper operations to be applied to these representations
- In the next section, let's explore the basis of Semantic Analysis – Word Sense and Relation.

Semantic Analysis
Part 3 – Word Sense and Relation

What is Word Sense?

What is Word Sense?

- In linguistics, a word sense is one of the meanings of a word.
- For example, a dictionary may have over 20 different senses of the word “bank", each of these having a different meaning based on the context of the word's usage in a sentence, as follows:

1. A financial institution that accepts deposits and channels the money into lending activities (Noun)

[6.3] He go to bank and draw some money. (“bank as” financial institution)

2. A supply or stock held in reserve for future use especially in emergencies (Noun)

[6.4] He go to the food bank to get some food.

3. A container (usually with a slot in the top) for keeping money at home (Noun)

[6.5] His coin bank was empty now.

4. A sloping land (especially the slope beside a body of water) (Noun)

[6.6] He pulled the canoe up on the bank. (Noun)

5. A long ridge or pile (Noun)

[6.7] A huge bank of earth.

6. Enclose with a bank (Verb)

[6.8] bank roads.

7. Cover with ashes so to control the rate of burning (Verb)

[6.9] Bank a fire.

8. Tip laterally (Verb)

[6.10] The pilot had to bank the aircraft.

9. A flight maneuver with the aircraft tips laterally about its longitudinal axis. (Noun)

[6.11] The F19 fighter went into a steep bank.

10. An arrangement of similar objects in a row or in tiers (Noun)

[6.12] He operated a bank of switches.

Types of Lexical Semantics

Totally there are six types of commonly used lexical semantics

- Homonymy
- Polysemy
- Metonymy
- Synonyms
- Antonyms
- Hyponymy and Hypernymy

Homonymy 同音词

Homonyms:

- Homonyms are words that are spelled the same and sound the same but have different meanings.
- The word homonym comes from the prefix homo- which means "the same," and the suffix -nym, which means "name."
- Therefore, a homonym is a word that has at least two different meanings, even though all uses look and sound exactly alike.
- For example:
  - $bank_1$: financial institution vs
    $bank_2$: sloping land
    [6.13] He went to the bank to draw some cash.
    [6.14] He was standing on the bank of the lake in the forest.
  - bat_1: club for hitting a ball vs
    bat_2: nocturnal flying mammal
    [6.15] He fields his position well and can handle the bat nicely.
    [6.16] And among mammals, bats live the longest relative to body size.
  - $play_1$: light-hearted recreational activity for diversion or amusement vs
    $play_2$: the activity of doing something in an agreed succession
    [6.17] This Shakespeare play is excellent.
    [6.18] It is still my play.
- Two related concepts with Homonymy:

1. Homographs - are usually defined as words that share the same spelling, regardless of how they are pronounced.[note 1] If they are pronounced the same then they are also homophones (and homonyms)
   E.g. bank1/bank2, bat1/bat2
2. Homophones – are usually defined as words that share the same pronunciation, regardless of how they are spelled. If they are spelled the same then they are also homographs (and homonyms); if they are spelled differently then they are also heterographs (literally "different writing").
   E.g. Write and right, Piece and peace, Two and too.

Homonymy causes problems for NLP applications

- Information retrieval
  - “cat scan”
- Machine Translation
  - $bank_1$ - Financial institution
    bank (English) -> la banque (French)
    [6.19] He goes to the bank to draw some cash. (English)
    [6.20] Il se rend à la banque pour retirer de l'argent. (French)
    $bank_2$ – Sloping land
    bank (English) -> la rive (French)
    [6.21] He lived by bank of the lake. (English)
    [6.22] Il habitait au bord du lac. (French)
- Text-to-Speech
  - bass低音 (stringed instrument) **vs.** bass鲈鱼 (fish)

Polysemy 多义词

What is Polysemy?

- Polysemy are words with the same spelling and distinct but related meanings.
- The distinction between polysemy and homonymy is often subtle and subjective, and not all sources consider polysemous words to be homonyms.

Example: Bank

[6.23] The **bank** was constructed in 1875 out of local red brick.

[6.24] He withdrew some money from the **bank** early this morning.

Are those the same sense?

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

A **polysemous** word has **related** meanings

- Most non-rare words have multiple meanings
- E.g. Get as word have at least 3 very distinct meanings:

[6.25] I get an apple from the basket. (to have something)
[6.26] I get it. (understand)
[6.27] She gets thinner. (Reach or cause to reach a specified state or condition)

Metonymy 換喻

What is Metonymy?

- Metonymy is a figure of speech (or trope) in which one word or phrase is substituted for another with which it's closely associated (such as "crown" for "royalty").
- Metonymy is also the rhetorical strategy of describing something indirectly by referring to things around it, as in describing someone's clothing to characterize the individual.
- In fact, Metonymy can be considered as a systematic relationship between senses, or known as systematic Polysemy.
- Lots of types of polysemy are systematic
  - `School, university, hospital`
  - All can mean the institution or the building.
- A systematic relationship:
  - Building ⇔ Organization
- Difference Between Metaphor and Metonymy
  - "Metonymy and metaphor also have fundamentally different functions.
  - Metonymy is about referring: a method of naming or identifying something by mentioning something else which is a component part or symbolically linked. Typical example is “crown” for “royalty” or “monarch”.
  - In contrast, a metaphor is about understanding and interpretation: it is a means to understand or explain one phenomenon by describing it in terms of another.“ For example:
    [6.28] Her business rises like phoenix.

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

The "Zeugma Test"

- Zeugma is a figure of speech in which a single word joins two (or more) parts of a sentence.
- Easy Examples of Zeugma which caused conflicts in semantics:
  [6.29] Wage neither war nor peace. (In literal meaning, there's a term to wage war but not to wage peace.)
  [6.30] He watched the brightness of the lightning and the pounding of the thunderstorm. (Obviously, only lightning can be watched but not the thunder).
- Zeugma Test in Word Sense analysis is to use a supposedly ambiguous expression is placed in a sentence in which several of its supposed meanings are forced together.
- If the resulting sentence sounds zeugmatic, that is taken as evidence for ambiguity.
- If it does not sound zeugmatic, that is taken as evidence against ambiguity.
- The "zeugma" test: Two senses of word serve?

  [6.31] Which United Airlines flights **serve** dinner?

  [6.32] Does Jack **serve** the Army?

  [6.33] ?Does United Airlines flights serve dinner and the Army?
- Since this conjunction sounds weird,
  - we say that these are **two different senses of "serve"**

Synonyms 同义词

- Word that have the same meaning in some or all contexts.
- Synonyms occur in a language in different contexts, such as formal and informal language, like you'd use in conversation vs. a business or academic paper.
- Also, some synonyms have slightly different connotations when they're used, even though they might mean the same thing.
- Example:
  - shut / close
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - pretty / beauty
- Two lexemes are synonyms
  - if they can be substituted for each other in all situations
  - If so they have the same **propositional meaning**

Can Two Words Be Truly Synonymous?

- But there are few (or no) examples of perfect synonymy.
- There is some debate as to whether two words can truly be synonymous.
- If they're different words, they must mean something slightly different or have contexts where you'd use one or the other, the reasoning goes, which makes them only nearly synonymous but not truly the same thing.
- In many cases, two words just can't be completely interchangeable in all occurrences.
- Even if many aspects of meaning are identical
- Still may not preserve the acceptability based on notions of politeness, slang, register, genre, etc.
- Example:
  - Big/large (Question: Are they exactly the same)?

  [6.34] This building is very big vs
  [6.35] This building is very large.

  [6.36] Janet is her big sister vs

  [6.37] ? Janet is her large sister.
- Why?
  - big has a sense that means being older, or grown up
  - large lacks this sense

Antonyms 反义词

- Antonyms is the semantic qualities or sense relations that exist between words with opposite meanings in certain contexts, contrast with synonymy.
- "Antonymy holds a place in society which other sense relations simply do not occupy.
- Whether or not there exists a 'general human tendency to categorize experience in terms of dichotomous contrast' is not easily judged.
- However, our exposure to antonymy is immeasurable.
- We memorise 'opposites' in childhood, encounter them throughout our daily lives, and possibly even use antonymy as a cognitive device to organise human experience.
- For example: `dark/light, short/long, fast/slow, rise/fall, hot/cold, up/down, in/out`
- More formally: antonyms can
  - Define a binary opposition or be at opposite ends of a scale
    - `long/short, fast/slow`
  - Be **reversive**:
    - `rise/fall, up/down`

Hyponymy and Hypernymy
下位词和上位词

- In linguistics, one sense is a **hyponym** of another if the first sense is more specific, denoting a subclass of the other sense.
- For example:
  - *car* is a hyponym of *vehicle*
  - *mango* is a hyponym of *fruit*
  - *chair* is a hyponym of *furniture*
- Conversely **hypernym/superordinate** (“hyper is super”)
  - *vehicle* is a **hypernym** of *car*
  - *fruit* is a **hypernym** of *mango*
  - *furniture* is a **hypernym** of *chair*
- Hyponymy is not restricted to nouns.
- The verb to see, for example, has several hyponyms— gaze, glimpse or stare, which can be considered as the specific “moment” of “seeing”.
- In terms of Computer Science and in particular in Object-oriented Programming, Hyponymy and Hypernymy relationship between word sense and relation can be considered as the relationship between “Class” and “Sub-class” concepts.
- For example: The Class “Vehicle” have THREE subclasses: Car, Lorry and Bus, while the Class “Fruit” can have numerous “Sub-classes” such as “Apple”, “Orange” and “Mango”.
- Or in the reverse manner: The concept “Vehicle” is the Superclass of “Car” and the concept “Fruit” is the Superclass of “ Mango”.
- Besides, words that are hyponyms of the same broader term (that is, a hypernym) are called co-hyponyms.
- The semantic relationship between each of the more specific words (such as daisy and rose) and the broader term (flower) is called hyponymy or inclusion.
- Same situation for the word sense relation of co-hypernymy.

Characteristics of Hyponymy

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

Hyponyms and Instances

- **WordNet** has both **classes** and **instances**.
- An **instance** is an individual, a proper noun that is a unique entity
  - `New York` is an **instance** of `city`
  - `USA is an` **instance** `of country`
- In Object Programming perspective, it can be considered as the relationship between Class vs Object
  - Class: The concept of things, objects
  - Object: The instance of the class
  - Example: “Person” is a class concept to describe an individual person while “John” is an object, which is an instance of that concept.
  - How about: the relationship between “Car” and “Tesla” ?
  - Are they Class-Object relationship or Superclass-subclass relationship?

Semantic Analysis
Part 4 – Word Sense Disambiguation

Word Sense Disambiguation

What is Word Sense Disambiguation?

- Word-sense disambiguation (WSD) is an open problem in computational linguistics concerned with identifying which sense of a word is used in a sentence.
- In terms of NLP, WSD may be defined as the ability to determine which meaning of word is activated by the use of word in a particular context.
- Lexical ambiguity, syntactic or semantic, is one of the very first problem that any NLP system faces.
- Part-of-speech (POS) taggers with high level of accuracy can solve Word's syntactic ambiguity.
- On the other hand, the problem of resolving semantic ambiguity is called WSD (word sense disambiguation).
- Resolving semantic ambiguity is harder than resolving syntactic ambiguity.
- For example, consider the two examples of the distinct sense that exist for the word "bass" –

  [6.38] Mary hate to hear the bass sound.

  [6.39] John likes to eat fried bass.

  - The occurrence of the word bass clearly denotes the distinct meaning.
  - In first sentence, it means frequency and in second, it means fish.
  - Hence, if it would be disambiguated by WSD then the correct meaning to the above sentences can be assigned as follows –

  [6.40] I can hear bass/frequency sound.

  [6.41] He likes to eat grilled bass/fish.

Word Sense Disambiguation

Difficulties in WSD

- Difference meaning across dictionaries
  - One problem with word sense disambiguation is deciding what the senses are, as different dictionaries and thesauruses will provide different divisions of words into senses.
  - Most research in the field of WSD is performed by using WordNet as a reference sense inventory for English.
  - WordNet is a computational lexicon that encodes concepts as synonym sets (e.g. the concept of car is encoded as { car, auto, automobile, machine, motorcar }).
  - More recently, BabelNet, a multilingual encyclopedic dictionary, has been used for multilingual WSD.
- POS Tagging
  - Part-of-speech (POS) tagging and sense tagging having been proven to be very closely related with each potentially making constraints to the other. Both WSD and part-of-speech tagging involve disambiguating or tagging with words.
  - However, algorithms used for one do not tend to work well for the other, mainly because the part of speech of a word is primarily determined by the immediately adjacent one to three words, whereas the sense of a word may be determined by words further away.
  - The success rate for part-of-speech tagging algorithms is at present much higher than that for WSD, state-of-the art being around 96% accuracy or better, as compared to less than 75% accuracy in word sense disambiguation with supervised learning.

Word Sense Disambiguation

Difficulties in WSD

- Inter-judge variance
  - WSD systems are normally tested by having their results on a task compared against those of a human.
  - However, while it is relatively easy to assign parts of speech to text, training people to tag senses has been proven to be far more difficult.
  - As human performance serves as the standard, it is an upper bound for computer performance.
  - Human performance, however, is much better on coarse-grained than fine-grained distinctions, so this again is why research on coarse-grained distinctions has been put to test in recent WSD evaluation exercises.
- Pragmatic (Discourse)
  - As will be discussed in Chapter 7, pragmatic and discourse is one of the most difficult problems in NLP.
  - Many AI researchers including me believe that one cannot parse meanings from words without some form of common sense ontology, which should all be analysis in pragmatic level.
  - As agreed by researchers, to properly identify senses of words one must know common sense facts and so-called World Knowledge as well.
  - Moreover, sometimes the common sense is needed to disambiguate such words like pronouns in case of having anaphoras or cataphoras in the text.
- Discreteness of senses
  - The notion of "word sense" is slippery and controversial.
  - Most people can agree in distinctions at the coarse-grained homograph level, but go down one level to fine-grained polysemy, and disagreements arise.
  - For example, in Senseval-2, which used fine-grained sense distinctions, human annotators agreed in only 85% of word occurrences.
  - Word meaning is in principle infinitely variable and context-sensitive. It does not divide up easily into distinct or discrete sub-meanings.

Word Sense Disambiguation

Method for Word Sense Disambiguation

- KnowledgeBase (Corpora & Dictionaries)
  - As the name suggests, for disambiguation, these methods primarily rely on dictionaries, treasures and lexical knowledge base.
  - They do not use corpora evidences for disambiguation. The Lesk method is the seminal dictionary-based method introduced by Michael Lesk in 1986. The Lesk definition, on which the Lesk algorithm is based is “measure overlap between sense definitions for all words in context”. However, in 2000, Kilgarriff and Rosensweig gave the simplified Lesk definition as “measure overlap between sense definitions of word and current context”, which further means identify the correct sense for one word at a time. Here the current context is the set of words in surrounding sentence or paragraph.
- Supervised Learning
  - For disambiguation, machine learning methods make use of sense-annotated corpora to train.
  - These methods assume that the context can provide enough evidence on its own to disambiguate the sense.
  - In these methods, the words knowledge and reasoning are deemed unnecessary.
  - The context is represented as a set of “features” of the words.
  - It includes the information about the surrounding words also.
  - Support vector machine and memory-based learning are the most successful supervised learning approaches to WSD.
  - These methods rely on substantial amount of manually sense-tagged corpora, which is very expensive to create.

Word Sense Disambiguation

Method for Word Sense Disambiguation

- Semi-Supervised Method
  - Due to the lack of training corpus, most of the word sense disambiguation algorithms use semi-supervised learning methods.
  - It is because semi-supervised methods use both labelled as well as unlabeled data.
  - These methods require very small amount of annotated text and large amount of plain unannotated text.
  - The technique that is used by semi-supervised methods is bootstrapping from seed data.
  - The bootstrapping approach starts from a small amount of seed data for each word: either manually tagged training examples or a small number of fire decision rules. The seeds are used to train an initial classifier, using any supervised method.
  - This classifier is then used on the untagged portion of the corpus to extract a larger training set, in which only the most confident classifications are included. The process repeats, each new classifier being trained on a successively larger training corpus, until the whole corpus is consumed, or until a given maximum number of iterations is reached.
  - Other semi-supervised techniques use large quantities of untagged corpora to provide co-occurrence information that supplements the tagged corpora. These techniques have the potential to help in the adaptation of supervised models to different domains.
- Un-Supervised Learning
  - These methods assume that similar senses occur in similar context.
  - That is why the senses can be induced from text by clustering word occurrences by using some measure of similarity of the context.
  - This task is called word sense induction or discrimination.
  - Unsupervised methods have great potential to overcome the knowledge acquisition bottleneck due to non-dependency on manual efforts.
  - Although the performance has been lower than for the other methods described above, but comparisons are difficult since senses induced must be mapped to a known dictionary of word senses.
  - If a mapping to a set of dictionary senses is not desired, cluster-based evaluations can be performed.
  - Alternatively, word sense induction methods can be tested and compared within an application.
  - For instance, it has been shown that word sense induction improves Web search result clustering by increasing the quality of result clusters and the degree diversification of result lists.
  - It is hoped that unsupervised learning will overcome the knowledge acquisition bottleneck because they are not dependent on manual effort.

Semantic Analysis
Part 5 – WordNet and Online Thesauri

WordNet

What is WordNet?

- WordNET is a lexical database of words in more than 200 languages in which we have adjectives, adverbs, nouns, and verbs grouped differently into a set of cognitive synonyms, where each word in the database is expressing its distinct concept.
- Unlike a dictionary that's organized alphabetically, WordNet is organized by concept and meaning.
- In fact, traditional dictionaries were created for humans but what's needed is a lexical resource more suited for computers. This is where WordNet becomes useful and powerful in NLP.
- WordNet is also freely and publicly available for download.
- WordNet's structure makes it a useful tool for computational linguistics and natural language processing.
- WordNet superficially resembles a thesaurus, in that it groups words together based on their meanings.
- However, there are some important distinctions.
- First, WordNet interlinks not just word forms—strings of letters—but specific senses of words.
- As a result, words that are found in close proximity to one another in the network are semantically disambiguated.
- Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity.
- Official site: https://wordnet.princeton.edu/
- Some other languages available or under development (Arabic, Finnish, German, Portuguese…)
- Fig. 6.1 shows the some basic statistical information about WordNet.

Category	Unique Strings
Noun	117,798
Verb	11,529
Adjective	22,479
Adverb	4,481

Fig. 6.1 Basic statistical information about WordNet

WordNet

What is Synsets?

- As said, WordNet is a network of words linked by lexical and semantic relations.
- Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, called synsets, each expressing a distinct concept.
- Synsets are interlinked using conceptual-semantic and lexical relations such as hyponymy and antonymy.
- In summary, WordNet contain over 117,000 sysets.
- Each of these synsets is linked to other synsets by means of a small number of "conceptual relations."
- Additionally, a synset contains a brief definition ("gloss") and, in most cases, one or more short sentences illustrating the use of the synset members.
- Word forms with several distinct meanings are represented in as many distinct synsets.
- Thus, each form-meaning pair in WordNet is unique.
- Fig. 6.2 shows an example concept of "book" defined by WordNet.
- In WordNet terminology, each group of synonyms is a synset, and a synonym that forms part of a synset is a lexical variant of the same concept.
- For example, in the network above, Word of God, Word, Scripture, Holy Writ, Holy Scripture, Good Book, Christian Bible and Bible make up the synset that corresponds to the concept Bible, and each of these forms is a lexical variant.
- The resulting network of meaningfully related words and concepts can be navigated with the WordNet browser (Fig. 6.3) which can be visited by http://wordnetweb.princeton.edu/perl/webwn.
- Fig. Show the word senses of word "Book" in WordNet.

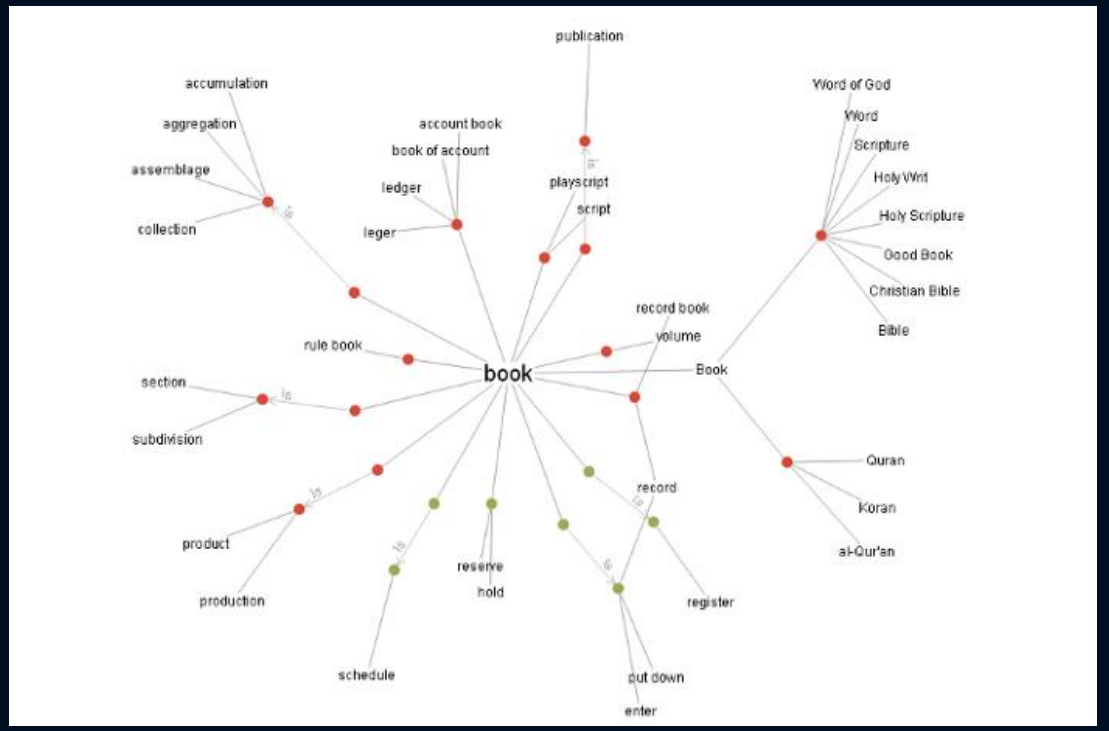


Fig. 6.2 The Concept of "Book" in WordNet

WordNet

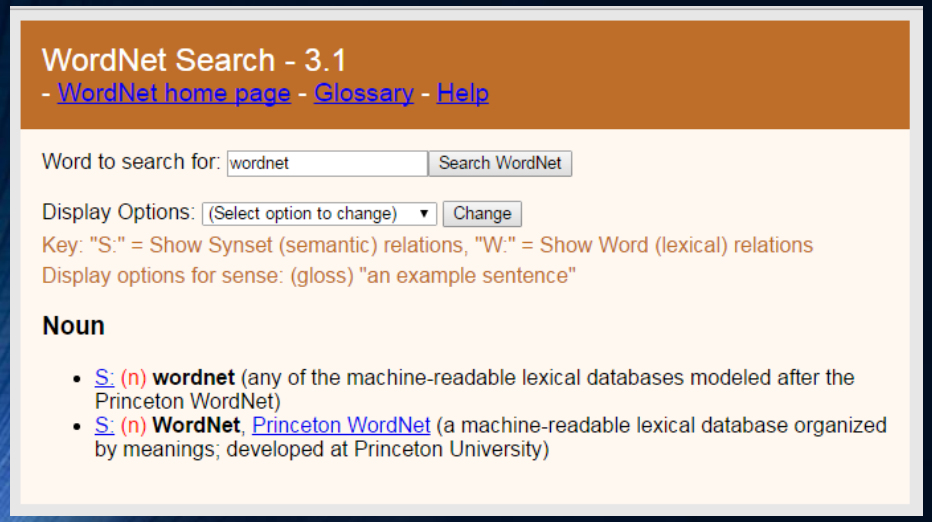


Fig. 6.3 A Snapshot of WordNet Browser

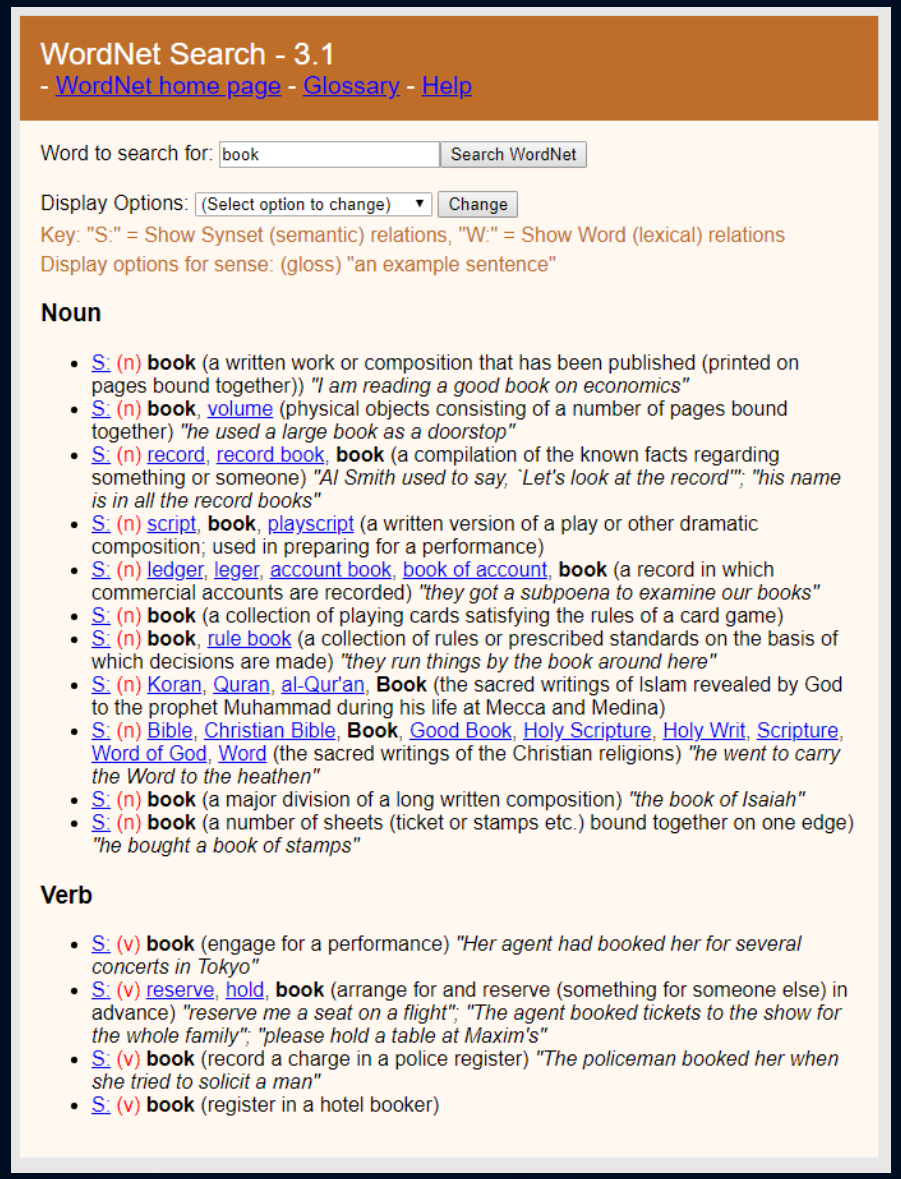


Fig. 6.4 The word sense of word “Book” in WordNet

WordNet

Knowledge Structure of WordNet

- Fig. 6.5 shows the basic structure of the WordNET.
- The main concept of the relationship between the words in the WordNETs network is that the words are synonyms like unhappy and sad and benefit and profit.
- These words show the same concept of using them in similar contexts by interchanging them.
- These types of words are grouped into synsets which are unordered sets.
- Where synsets are linked together if they are having even small conceptual relations.
- Fig. 6.6 shows the example of Synet “Benefit”.
- In this example, we can see the structure of any synset where we are having synonyms of benefit in the array of synsets with the definition and the example of usage of benefit word.
    - This synset is related to another synset word, where the words benefit and profit are defined as synomyms with the same meaning.
    - Benefit (profit) is defined as: An advantage or profit gained from some thing.
    - With live example: He receives benefits of computer trade.

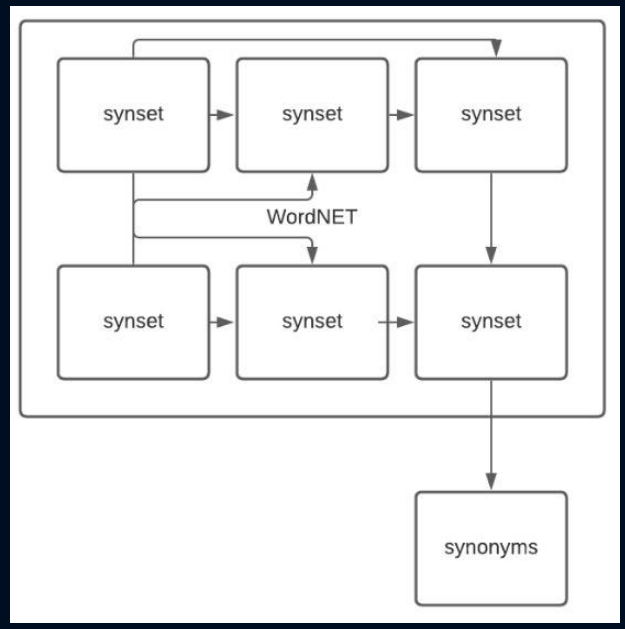


Fig. 6.5 Basic structure of WordNet

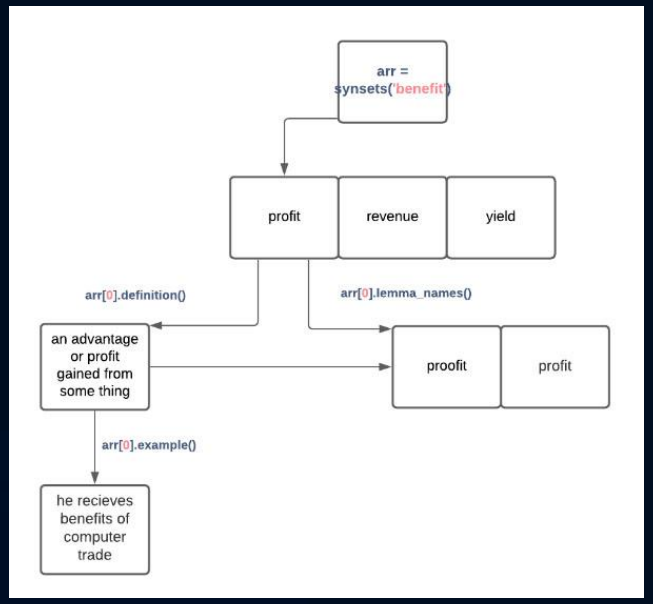


Fig. 6.6 An example of synet “benefit”

WordNet

What are major lexical relations captured in WordNet?

- The most frequently encoded relation among synsets is the super-subordinate relation (also called hyperonymy, hyponymy or ISA relation).
- It links more general synsets like {furniture, piece_of_furniture} to increasingly specific ones like {bed} and {bunkbed}.
- Thus, WordNet states that the category furniture includes bed, which in turn includes bunkbed; conversely, concepts like bed and bunkbed make up the category furniture.
- All noun hierarchies ultimately go up the root node {entity}. Hyponymy relation is transitive: if an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture.
- WordNet distinguishes among Types (common nouns) and Instances (specific persons, countries and geographic entities).
- Thus, book is a type of publication, George Bush is an instance of a president.
- Instances are always leaf (terminal) nodes in their hierarchies.
- Major lexical relations include the following:
  - Synonymy: Words with similar meanings.
  - Polysemy: Words have more than one sense.
  - Hyponymy/Hypernymy: Is-a relation between words.
  - Meronymy/Holonymy: Part-whole relation between words.
  - Antonymy: Lexical opposites such as (large, small).
  - Troponymy: Applicable for verbs. For example, whisper is a troponym of talk since whisper elaborates on the manner of talking.
- Fig. 6.7 shows the major lexical relations captured in WordNet with examples.

Semantic Relation	Syntactic Category	Examples
Synonymy (similar)	N, V, Aj, Av	pipe, tube rise, ascend sad, unhappy rapidly, speedily
Antonymy (opposite)	Aj, Av, (N, V)	wet, dry powerful, powerless friendly, unfriendly rapidly, slowly
Hyponymy (subordinate)	N	sugar maple, maple maple, tree tree, plant
Meronymy (part)	N	brim, hat gin, martini ship, fleet
Troponomy (manner)	V	march, walk whisper, speak
Entailment	V	drive, ride divorce, marry

Note: *N = Nouns* *Aj = Adjectives* *V = Verbs* *Av = Adverbs*

Fig. 6.7 Major lexical relations captured in WordNet with examples

WordNet

Applications of WordNet and Thesauri?

- Information Extraction
- Information Retrieval
- Question Answering
- Bioinformatics and Medical Informatics
- Machine Translation
- Another common use of WordNet is to determine the similarity between words.
- Various algorithms have been proposed, including measuring the distance among words and synsets in WordNet's graph structure, such as by counting the number of edges among synsets.
- The intuition is that the closer two words or synsets are, the closer their meaning.
- A number of WordNet-based word similarity algorithms are implemented in a Perl package called WordNet::Similarity, and in a Python package including NLTK and SpaCy, which will be explored in the Part II of this book – the NLP Workshop sessions.

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

What is MeSH?

- The Medical Subject Headings (MeSH) thesaurus is a controlled and hierarchically-organized vocabulary produced by the National Library of Medicine of USA.
- It is used for indexing, cataloging, and searching of biomedical and health-related information.
- MeSH includes the subject headings appearing in MEDLINE/PubMed, the NLM Catalog, and other NLM databases.
- 177,000 entry terms that correspond to 26,142 biomedical “headings”.
- As the conceptual territory of the biomedical literature has expanded, so too has the number of MeSH terms used for indexing.
- The 2020 edition contained more than 25,000 subject headings, considerably more than the approximately 4400 in the original set introduced in 1960.
- These subject headings are organized into an eleven-level hierarchy, as well as 83 subheadings. The vocabulary can be explored using the online MeSH browser provided by the US National Library of Medicine (NLM), at https://www.nlm.nih.gov/mesh/meshhome.html.
- The MeSH headings are organized in a “Knowledge Tree" with 16 main branches:
    - A. Anatomy, B. Organisms, C. Diseases, D. Chemicals and Drugs, E. Analytical, Diagnostic and Therapeutic Techniques and Equipment, F. Psychiatry and Psychology, G. Phenomena and Processes, H. Disciplines and Occupations, I. Anthropology, Education, Sociology and Social Phenomena, J. Technology, Industry, Agriculture, K. Humanities, L. Information Science, M. Named Groups, N. Health Care, V. Publication Characteristics, Z. Geographicals.
- In addition to this hierarchically structured set of canonical terms, the MeSH vocabulary also contains a vast number of entry terms, which are intended to be synonyms of the canonical heading terms.
- The following using Hemoglobins 血红蛋白 as an example.

The MeSH Hierarchy

Example:

- Hemoglobins 血红蛋白
- Entry Terms: Eryhem, Ferrous Hemoglobin, Hemoglobin
- Definition: The oxygen-carrying proteins of ERYTHROCYTES. They are found in all vertebrates and some invertebrates. The number of globin subunits in the hemoglobin quaternary structure differs between species. Structures range from monomeric to a variety of multimeric arrangements.

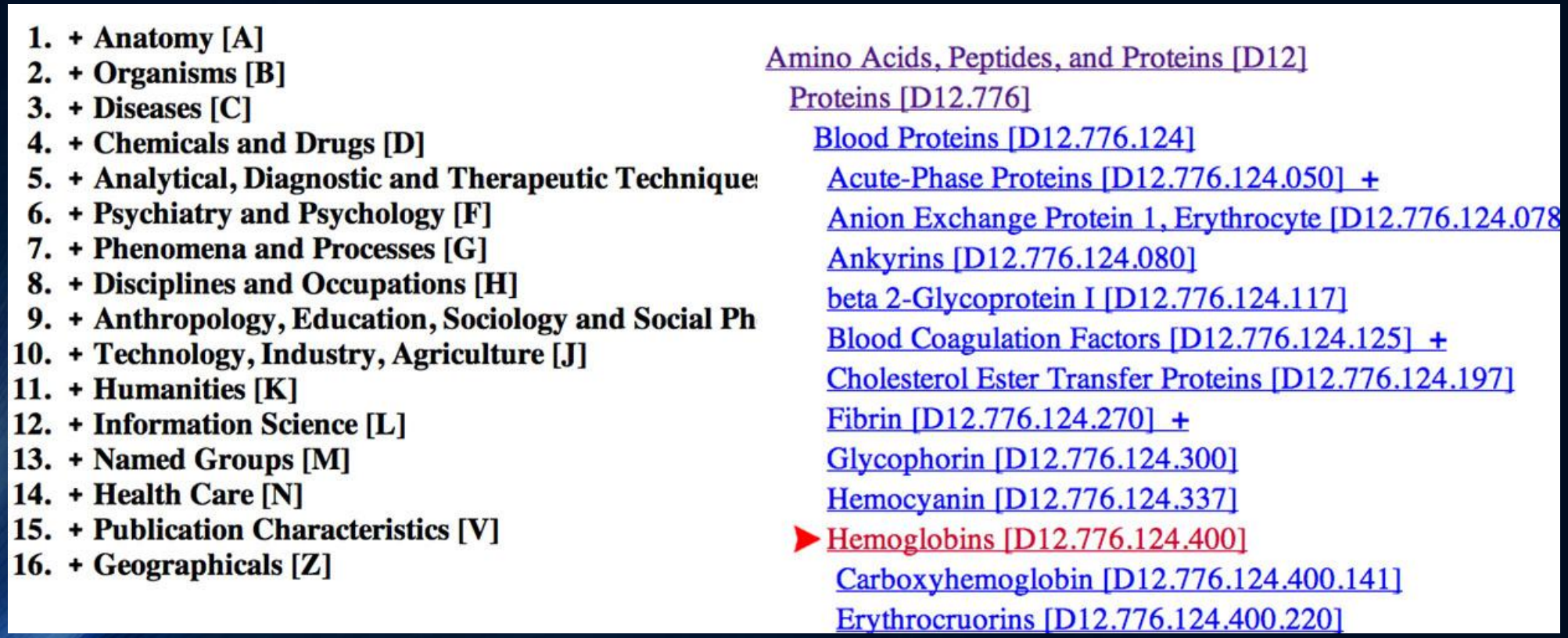


Fig. 6.8 Example of MeSH Hierarchy using the term “Hermoglobins”

Uses of the MeSH Ontology

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

Semantic Analysis
Part 6 – Word Similarity & Thesaurus Methods

Word Similarity

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

Why word similarity so important?

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

Word similarity and word relatedness

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

Two classes of similarity algorithms

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

Path based similarity

- Two concepts (senses/synsets) are similar if they are near each other in the thesaurus hierarchy
  - =have a short path between them
  - concepts have path 1 to themselves
  - Fig. 6.9 shows the synsets related to “car” from WordNet and their path-based similarity to different related synsets.
  - Example:
    - pathlen(car, car) = 1
    - pathlen(car, automotive) = 2
    - pathlen(car, truck) = 3
    - pathlen(car, minibike) = 5
    - pathlen(car, transport) = 5
    - pathlen(car, artifact) = 7
    - pathlen(car, tableware) = 10
    - pathlen(car, fork) = 12

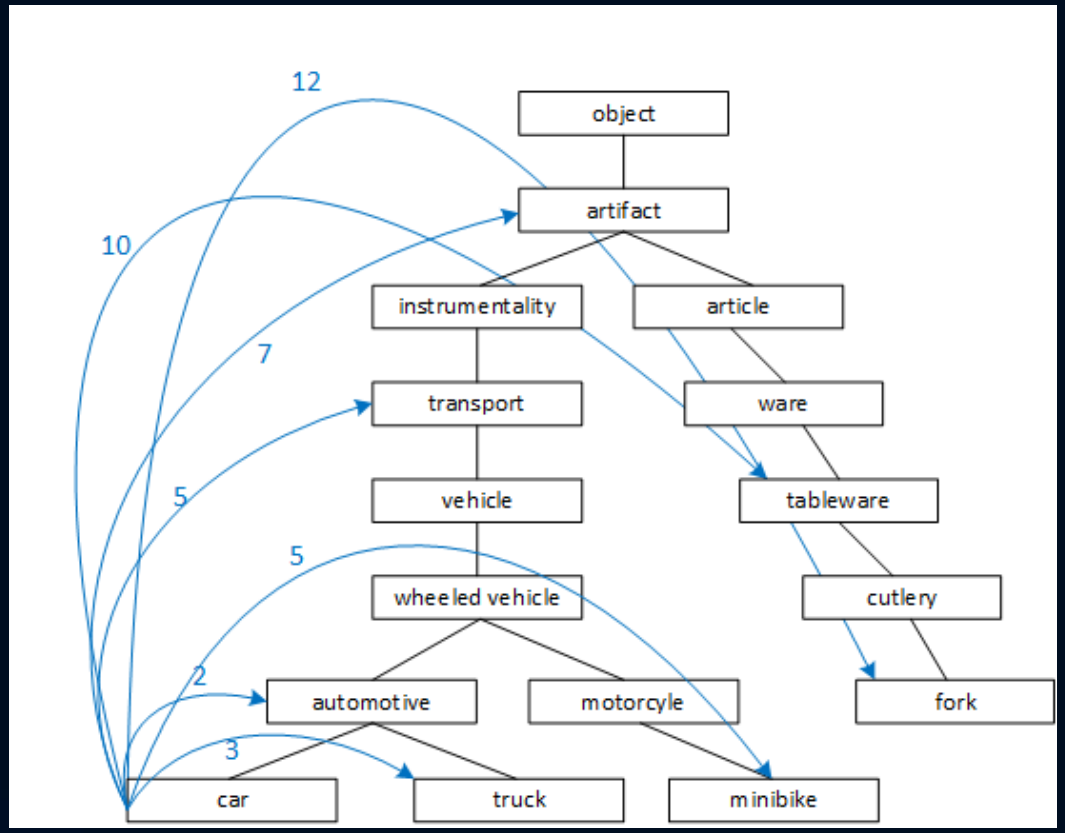


Fig. 6.9 Path-based Similarity for concept related to “car”

Simpath and Wordsim

- pathlen(c_1,c_2) = 1 + number of edges in the shortest path in the hypernym graph between sense nodes c_1 and c_2 (6.1)
- ranges from 0 to 1 (identity)

- simpath(c_1,c_2) = $$\frac{1}{\text{pathlen}(c_1, c_2)}$$ (6.2)

- wordsim(w_1,w_2) = $\max\ (\ \text{simpath}(c_1,c_2)\ )$ (6.3)

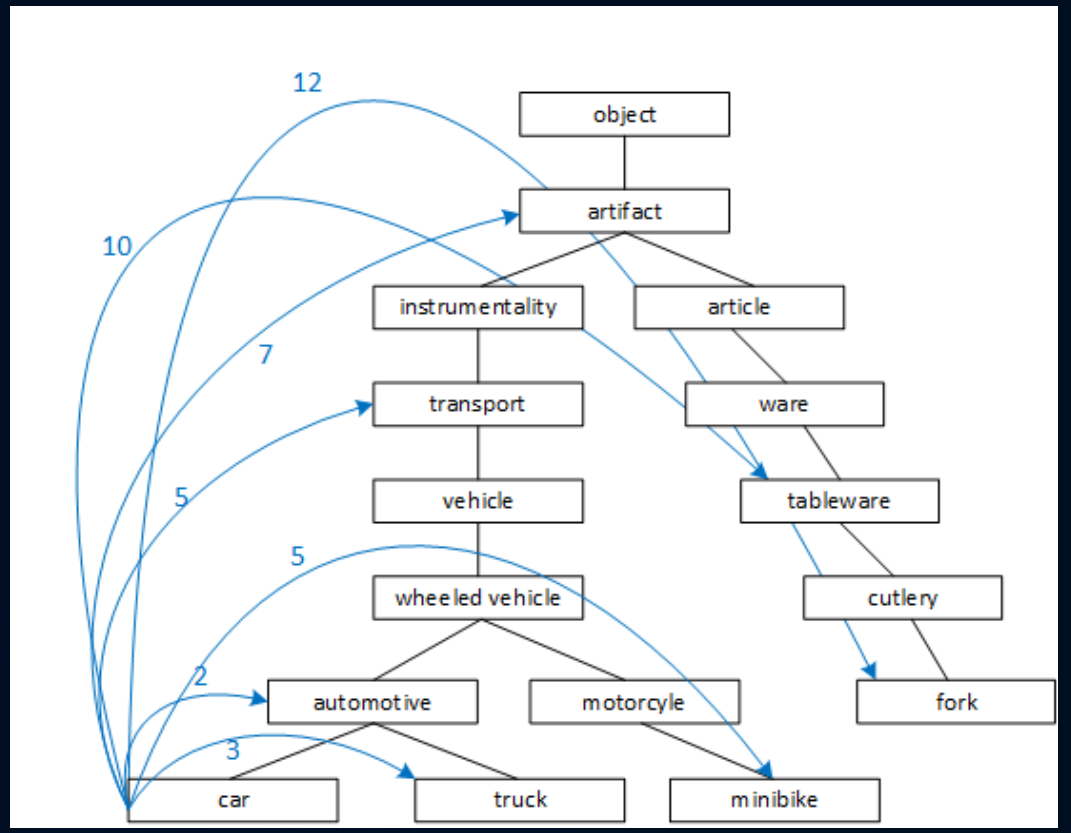
$$\forall\ c_1 \in \text{senses}(w_1), c_2 \in \text{senses}(w_2)$$

Example: path-based similarity

- Using

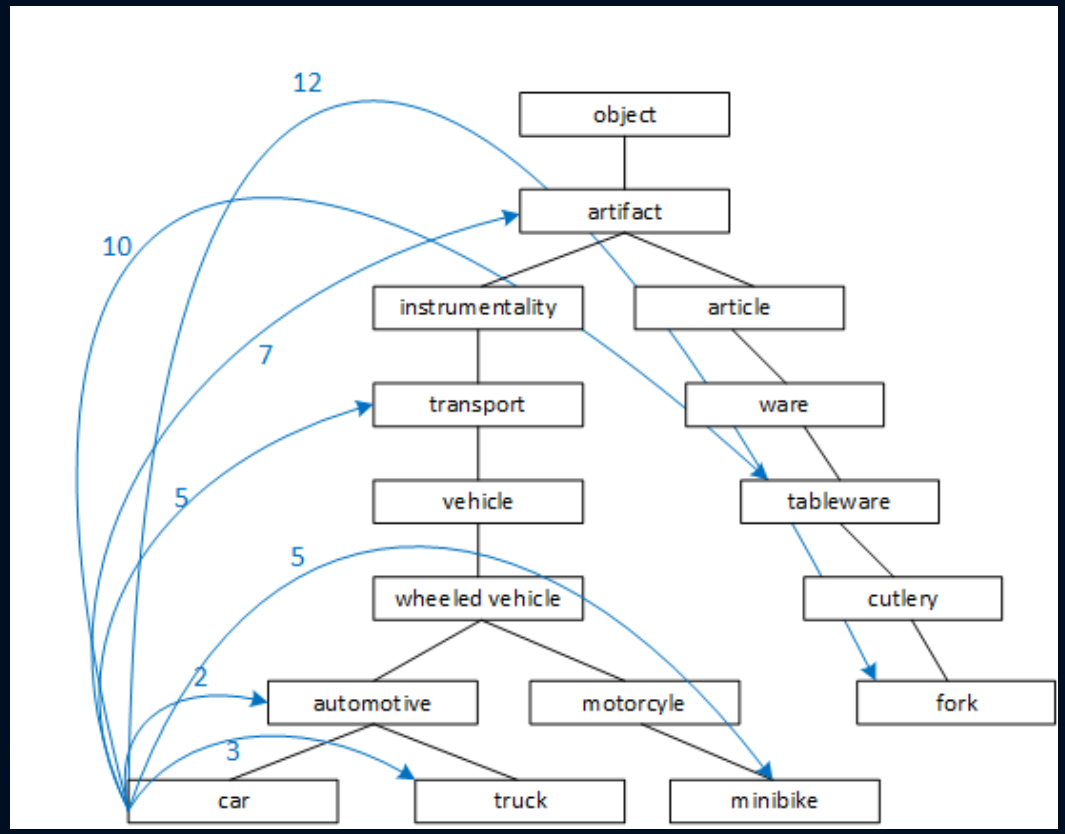
$$\text{simpath}(c_1,c_2) = 1/\text{pathlen}(c_1,c_2) \qquad (6.2)$$

- For our previous "car" concept example.
  - simpath(car, car) = 1/1 = 1.0
  - simpath(car, automotive) = ½ = 0.50
  - simpath(car, truck) = 1/3 = 0.33
  - simpath(car, minibike) = 1/5 = 0.20
  - simpath(car, transport) = 1/5 = 0.20
  - simpath(car, artifact) = 1/7 = 0.14
  - simpath(car, tableware) = 1/10 = 0.10
  - pathlen(car, fork) = 1/12 = 0.08



Problem with basic path-based similarity

- Assumes each link represents a uniform distance
  - But *car* to *minibike* seems to us to be closer than *car* to *transport (Why?)*
  - In general, the "higher" the synsets in the synset tree, the most "abstract" they are
  - Example: Object is more abstract than Artifact, Transport is more abstract than Vehicle, etc.
  - That why, even simpath(car, minibike) and simpath(car, transport) are the same, their "semantic relation" with each other are rather difference.
  - Of course, synsets in the other "branch" of the synset tree are even more "related"
    - E.g. car vs. tableware or even fork.
- We instead want a metric that
  - Represents the cost of each edge independently
  - Words connected only through abstract nodes
    - are less similar



Information content similarity metrics

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

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

Information content similarity

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

    of *automotive*, *wheeled vehicle*, *vehicle*, etc
  - Let words(c) be the set of all words that are children of node c,
  - The probability of information content similarity P(c) in a corpus is given by:

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N} \qquad (6.4)$$

  - Using our "car" concept as example:
    - words("transport") = {vehicle, wheeled vehicle, automotive, car, truck, motorcycle, minibike}
    - words("automotive") = {car, truck}
  - Fig. 6.10 shows the synset tree of "car" with the associated P(c) (up to the "transport" level in a given corpus.

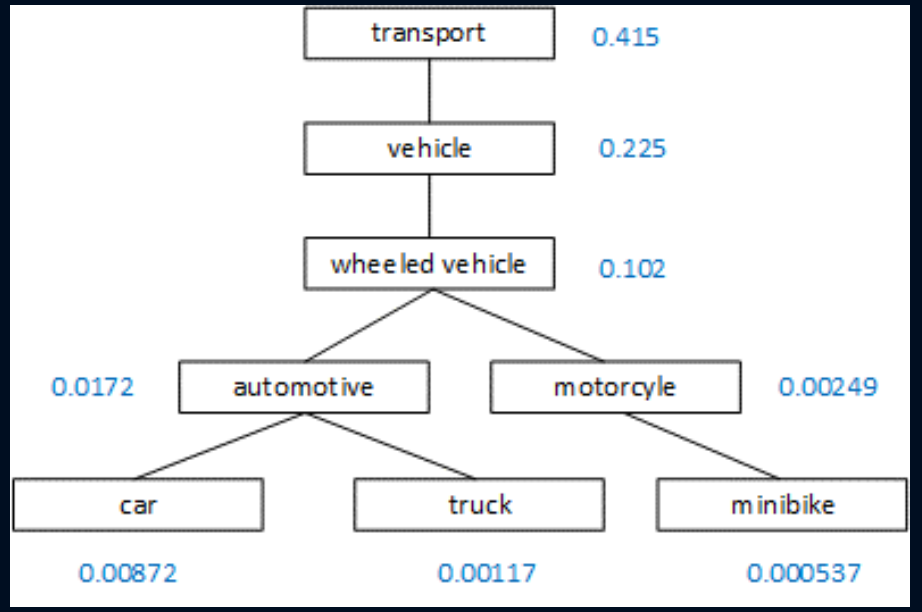
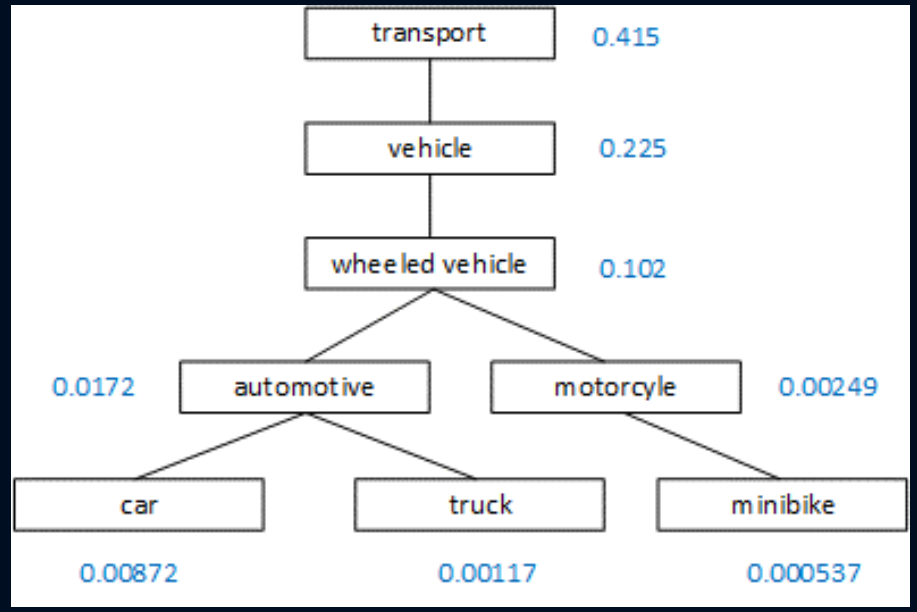


Fig. 6.10 Syntset tree of "car" with the associated P(c) (up to the "transport" level in a given corpus

Information content: definitions

- Information content:
  - IC(c)=-logP(c) (6.5)
- Lowest common subsumer 归类
  - LCS(c1,c2) = the lowest common subsumer (6.6)
    - 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



Using information content for similarity: the Resnik method

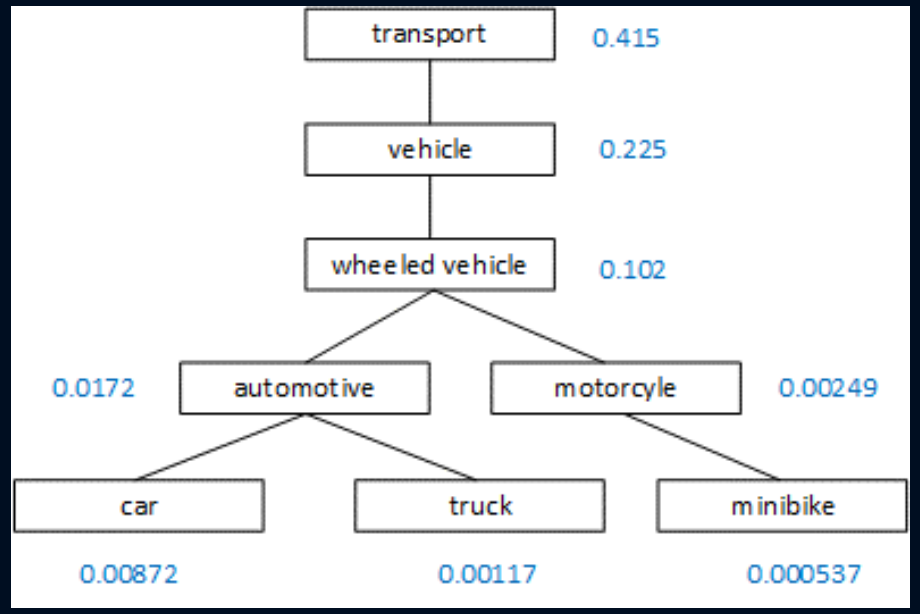
- The similarity between two words is related to their common information
- The more two words have in common, the more similar they are
- Resnik: measure common information as:
  - The information content of the most informative (lowest) subsumer (MIS/LCS) of the two nodes
  - $sim_{resnik}(c_1,c_2) = -\log P(LCS(c_1,c_2))$ (6.7)

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

Dekang Lin method

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

- Intuition: Similarity between A and B is not just what they have in common
- The more **differences** between A and B, the **less similar** they are:
  - Commonality: the more A and B have in common, the more similar they are
  - Difference: the more differences between A and B, the less similar
- Commonality: IC(common(A,B))
- Difference: IC(description(A,B)-IC(common(A,B))
- 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 give by:
  simLin(A,B) = log P(common(A,B)) / log P(description(A,B)) (6.8)
- Lin furthermore shows (modifying Resnik)that info in common is twice the info content of the LCS



- $\text{SimLin}(c1, c2) = \frac{2 \times \log P(LCS(c1,c2))}{\log P(c1) + \log P(c2)}$ (6.9)
- $\text{SimLin}(car, minibike) = \frac{2 \times \log P(wheeled\ vehicle)}{\log P(\text{car}) + \log P(\text{minibike})} = \frac{2 \times \log P(0.102)}{\log P(0.00872) + \log P(0.000537)}$ = 0.372
- $\text{SimLin}(car, \text{truck}) = \frac{2 \times \log P(\text{automotive})}{\log P(\text{car}) + \log P(\text{truck})} = \frac{2 \times \log P(0.0172)}{\log P(0.00872) + \log P(0.00117)}$ = 0.707
- Rather “make sense” indeed!

The (extended) Lesk Algorithm

- A thesaurus-based measure that looks at **glosses**
- Two concepts are similar if their glosses contain similar words
  - ***Drawing paper***: paper that is specially prepared for use in drafting
  - the art of transferring designs from specially prepared paper to a wood or glass or metal surface

- For each *n*-word phrase that's in both glosses
  - Add a score of n^2
  - Paper and specially prepared for $1 + 2^2 = 5$
  - Compute overlap also for other relations
    - glosses of hypernyms and hyponyms

Summary: thesaurus-based similarity

$$
\begin{aligned}
\text{sim}_{\text{path}}(c_1, c_2) &= -\log \text{pathlen}(c_1, c_2) \\
\text{sim}_{\text{Resnik}}(c_1, c_2) &= -\log P(\text{LCS}(c_1, c_2)) \\
\text{sim}_{\text{Lin}}(c_1, c_2) &= \frac{2 \times \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)} \\
\text{sim}_{\text{jc}}(c_1, c_2) &= \frac{1}{2 \times \log P(\text{LCS}(c_1, c_2)) - (\log P(c_1) + \log P(c_2))} \\
\text{sim}_{\text{eLesk}}(c_1, c_2) &= \sum_{r,q \in \text{RELS}} \text{overlap}(\text{gloss}(r(c_1)), \text{gloss}(q(c_2)))
\end{aligned}
\tag{6.10}
$$

Semantic Analysis
Part 7 – Word Similarity: Distributed Similarity

Problems with thesaurus-based meaning

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

Distributional models of meaning

- Also called vector-space models of meaning
- Offer much higher recall than hand-built thesauri
  - Although they tend to have lower precision
- Zellig Harris (1954): "**oculist** and **eye-doctor** … occur in almost the same environments….
  If A and B have almost identical environments, we say that they are synonyms.
- Firth (1957): "You shall know a word by the company it keeps!"
- Example:
  [6.41] A bottle of ***Baileys*** is on the table
  [6.42] Many coffee drinker likes ***Baileys***
  [6.43] Baileys will make you drunk
  [6.44] We make Baileys out of Irish whiskey and cream
- From context words humans can guess ***Baileys*** means
  - an alcoholic coffee beverage flavoured with cream and Irish whiskey.
- Intuition for algorithm:
  - Two words are similar if they have similar word contexts.

Word Vectors

- Basically, Word Vector it is a vector of weights.
- In a simple 1-of-N encoding every element in the vector is associated with a word in the vocabulary.
- The encoding of a given word is the vector in which the corresponding element is set to one, and all other elements are zero.
- 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 neighbourhood of w", given by:
  w = (f1, f2, f3, … , fN)
- In our Baileys example:
  If w = Baileys, v1 = coffee, v2 = Whiskey, v3 = beer, v4 = cream, …
  w = (1 , 1, 0, 1, …)

Term-document matrix

- Term document matrix is also a method for representing the text data.
- In this method, the text data is represented in the form of a matrix.
- The rows of the matrix represent the sentences from the data which needs to be analyzed and the columns of the matrix represent the word.
- Mathematically, each cell: count of term *t* in a document *d*: $tf_{t,d}$:
  - Each document is a count vector in $\mathbb{N}^v$: a column below

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	1	1	8	15	1	20
soldier	2	2	12	36	0	4
fool	37	58	1	5	3	7
trick	1	3	1	1	3	3

Fig. 6.11 Term-document matrix of 6 famous English literature

Term-document matrix

- Two documents are similar if their vectors are similar

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	1	1	8	15	1	20
soldier	2	2	12	36	0	4
fool	37	58	1	5	3	7
trick	1	3	1	1	3	3

Fig. 6.12 Term-document matrix comparison by document vectors

The words in a term-document matrix

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

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	1	1	8	15	1	20
soldier	2	2	12	36	0	4
fool	37	58	1	5	3	7
trick	1	3	1	1	3	3

Fig. 6.13 Illustration of count vector for a SIX document domain

The words in a term-document matrix

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

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	1	1	8	15	1	20
soldier	2	2	12	36	0	4
fool	37	58	1	5	3	7
trick	1	3	1	1	3	3

Fig. 6.14 Sample of two similar word by vector comparison across six documents

The Term-Context matrix

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

Should we use raw counts?

- For the term-document matrix
  - We used tf-idf instead of raw term counts
- For the term-context matrix
  - Positive Pointwise Mutual Information (PPMI) is common

Pointwise Mutual Information

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

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)} \quad (6.11)$$

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

$$\mathrm{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)} \quad (6.12)$$

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

Computing PPMI on a term-context matrix

Given Matrix F with W rows (words) and C columns (contexts) and $\mathrm{f_{ij}}$ is # of times $\mathrm{w_i}$ occurs in context $\mathrm{c_j}$
Positive PMI (PPMI) between word1 and word2 can be written as follows:-

$$PPMI(Word1, Word2) = max(\log_2 \frac{p(Word1, \quad Word2)}{p(Word1) \quad p(Word2)}, 0) \tag{6.13}$$

Where :

$$\begin{Bmatrix} \mathrm{PMI(W,C)} & ,if\ \mathrm{PMI(W,C)} > 0 \\ 0 & ,if\ \mathrm{PMI(W,C)} < 0 \end{Bmatrix} \tag{6.14}$$

$$p(W, C) = \frac{f_{ij}}{\sum_{i=1}^{W} \sum_{j=1}^{C} f_{ij}}$$

$$p(W_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \tag{6.15}$$

$$p(C_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

In which:
p(W,C) -is the probability of seeing Target word w and context word c together.
p(W) & p(C) – the probability of occurring Target word w & context word C , if they're independent
f_{ij} is number of times W_i occurs in context C_j

Example of Computing PPMI on a term-context matrix

Let's use the previous document term matrix of six English literature as example:

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick	Word
battle	1	1	8	15	1	20	46
soldier	2	2	12	36	0	4	56
fool	37	58	1	5	3	7	111
trick	1	3	1	1	3	3	12
Context	41	64	22	57	7	34	225

Fig. 6.15 Term-context matrix of the six contexts with word and context total counts

P(W = fool, C=As You like it) = 37/225 = 0.164
P(W=fool) = 111/225 = 0.493
P(C=As You like it) = 41/225 = 0.182

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick	Word
battle	1	1	8	15	1	20	0.204
soldier	2	2	12	36	0	4	0.249
fool	37	58	1	5	3	7	0.493
trick	1	3	1	1	3	3	0.053
Context	0.182	0.284	0.098	0.253	0.031	0.151	1

Fig. 6.16 Term-context matrix of the six contexts with word and context total probabilities

Example of Computing PPMI on a term-context matrix

So from above information let's calculate the PMI score for the word "fool" with co-occurred with context from "C1 = As You Like it" :-

$$PMI(W,C) = log\frac{p(W,C)}{p(W)p(C)}$$

$$PMI(fool,C1) = log\frac{0.164}{0.493*0.182} = 0.604$$

Similarly, we can calculate the rest of PMI values for this term-context matrix as follow:

Note that: $PPMI(W,C) = \begin{Bmatrix} PMI(W,C), if\ PMI(W,C) > 0 \\ 0, if\ PMI(W,C) < 0 \end{Bmatrix}$

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	0.000	0.000	0.576	0.252	0.000	1.057
soldier	0.000	0.000	0.785	0.931	-	0.000
fool	0.604	0.608	0.000	0.000	0.000	0.000
trick	0.000	0.000	0.000	0.000	2.084	0.503

Fig. 6.17 Term-context matrix of the six contexts with PPMI values

Weighing PMI

- If you notice from above matrix then you'll know that PMI is biased toward infrequent events.
- Very rare words have very high PMI values .
- So we can improve PMI further with two possible solutions-
  - Use add-k smoothing (e.g. Add-1)
  - Give rare words slightly higher probabilities (which has a similar effect)

K-Smoothing in PMI computation

- As we've seen PMI is biased toward infrequent events , in our case possibility of two words getting co-occurred together.
- So we add 2 (i.e. set k = 2) in every cell of co-occurrence matrix like below:-

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick	Word
battle	3	3	10	17	3	22	58
soldier	4	4	14	38	2	6	68
fool	39	60	3	7	5	9	123
trick	3	5	3	3	5	5	24
Context	49	72	30	65	15	42	273

Fig. 6.18 Term-context matrix of the six contexts with word and context total count with Add-2 Smoothing

- The corresponding probabilities matrix after Add-2 Smoothing as below.

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick	Word
battle	0.011	0.011	0.037	0.062	0.011	0.081	0.212
soldier	0.015	0.015	0.051	0.139	0.007	0.022	0.249
fool	0.143	0.220	0.011	0.026	0.018	0.033	0.451
trick	0.011	0.018	0.011	0.011	0.018	0.018	0.088
Context	0.179	0.264	0.110	0.238	0.055	0.154	1.000

Fig. 6.19 Term-context matrix of the six contexts with word and context total prob. with Add-2 Smoothing

K-Smoothing in PMI computation

- The Term-context matrix with PPMI values after applying Add-2 Smoothing is shown in Fig. 6.20
- Theoretically speaking, giving the rare context words, it might have certain improvement in the PPMI values.
- However, in our case, not much improvements are found.
- Another method is achieved by raising the context probabilities to a certain factor α, say 0.8.
- $PPMI_\alpha(w,c) = max\left(log\frac{P(w,c)}{P(w)P_\alpha(c)}, 0\right)$

  where: $P_\alpha(c) = \frac{count(c)^\alpha}{\sum_c count(c)^\alpha}$
- For example: say P(a) = 0.95 and P(b) = 0.05,
- $P_\alpha(a) = \frac{0.95^{0.8}}{0.95^{0.8}+0.05^{0.8}}$ = 0.913, $P_\alpha(b) = \frac{0.05^{0.8}}{0.95^{0.8}+0.05^{0.8}}$ = 0.083.
- Fig. 6.22 and 6.23 shows the results using α = 0.8 and 0.9 respectively of our example for comparing six context doc.

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	0.000	0.000	0.450	0.208	0.000	0.902
soldier	0.000	0.000	0.628	0.853	0.000	0.000
fool	0.569	0.615	0.000	0.000	0.000	0.000
trick	0.000	0.000	0.129	0.000	1.333	0.303

Fig. 6.20 Term-context matrix of the six contexts with PPMI values with Add-2 Smoothing

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	0.000	0.000	0.144	0.000	0.000	0.625
soldier	0.000	0.000	0.435	0.581	-	0.000
fool	0.369	0.373	0.000	0.000	0.000	0.000
trick	0.000	0.000	0.000	0.000	1.315	0.000

Fig. 6.21 Term-context matrix of the six contexts with PPMI values with α = 0.80

	As You like it	Twelfth Night	Julius Caesar	Henry V	Adv of Sherlock Holmes	Moby Dick
battle	0.000	0.000	0.460	0.137	0.000	0.941
soldier	0.000	0.000	0.711	0.858	-	0.000
fool	0.587	0.592	0.000	0.000	0.000	0.000
trick	0.000	0.000	0.000	0.000	1.798	0.217

Fig. 6.22 Term-context matrix of the six contexts with PPMI values with α = 0.9

Context and Word Similarity Measurement

- For Context and Word similarity measurement against the context and word vector.
- Reminder: cosine for computing similarity, which is given by:
  where:

  v_i is the PPMI value for word v in context i
  w_i is the PPMI value for word w in context i.
  Cos(v,w) is the cosine similarity of v and w.

$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\vec{v}}{|\vec{v}|}\cdot\frac{\vec{w}}{|\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \quad (6.16)$$

- Fig. 6.23 shows context and word similarity measurement from the six literature.
- For context comparison, we compare the cosine similarity between C1: As You like it with other five literature.
  - In which cosine(C1, C2) is the highest among them, 0.453 as compared with other ranging from 0.044 (C3:Julius Caesar) and 0.157 (C6:Moby Dick)
  - Which is rather make sense as the context of Shapeware's work on As You like it is more similar in theme with Twelfth Night than other works, let's alone with other literatures.
- For word comparison, we compare the W4: Trick with the other 3 words across the six literature.
  - In which cosine(W4:trick, W3:fool) is the highest similarity among the other two words "W1:battle" and "W2:Solider" which in fact they are very close in term of meaning and use of English.
- Other possible similarity measurement include: Jaccard, Dice and JS, which is given by the following equation:

$$\begin{aligned} \text{sim}_{\text{Jaccard}}(\vec{v},\vec{w}) &= \frac{\sum_{i=1}^{N}\min(v_i,w_i)}{\sum_{i=1}^{N}\max(v_i,w_i)} \\ \text{sim}_{\text{Dice}}(\vec{v},\vec{w}) &= \frac{2\times\sum_{i=1}^{N}\min(v_i,w_i)}{\sum_{i=1}^{N}(v_i+w_i)} \\ \text{sim}_{\text{JS}}(\vec{v}||\vec{w}) &= D(\vec{v}|\frac{\vec{v}+\vec{w}}{2}) + D(\vec{w}|\frac{\vec{v}+\vec{w}}{2}) \end{aligned} \quad (6.17)$$

	C1:As You like it	C2:Twelfth Night	C3:Julius Caesar	C:4Henry V	C5:Adv of Sherlock Holmes	C6:Moby Dick	Wx * Wx	W4 * Wx	Sim(W4, Wx)
W1:battle	1	1	8	15	1	20	692	90	0.077
W2:soldier	2	2	12	36	0	4	1462	68	0.035
W3:fool	37	58	1	5	3	7	1511	247	0.124
W4:trick	1	3	1	1	3	3	24		
Cx * Cx	1375	3378	210	1547	19	474			
C1 * Cx		2154	70	273	115	290			
Sim(C1, Cx)		0.453	0.044	0.093	0.082	0.157			

Fig. 6.23 Context and Word Similarity from the SIX sample literature

Evaluating similarity

- Intrinsic Evaluation:
  - Correlation between algorithm and human word similarity ratings
- Extrinsic (task-based, end-to-end) Evaluation:
  - Malapropism (spelling error) detection
  - WSD
  - Essay grading
  - Taking TOEFL multiple-choice vocabulary tests

Summary

1. Introduction
2. Lexical vs Compositional Semantic Analysis
3. Word Senses & Relations
4. Word Sense Disambiguation
5. WordNet and Online Thesauri
6. Word Similarity & Thesaurus Methods
7. Distributed Similarity
8. In the workshop #4, we will have a practical session on Semantic Analysis with Word Vectors and Semantic Similarity using spaCy, which include:
   - Implementation of word vectors in Python
   - Learning how to use spaCy's pretrained vectors
   - Advanced semantic similarity methods using spaCy technology.

Next

Pragmatics (Discourse) Analysis