# Module 4

## Semantic Analysis

# Content

- Lexical Semantics
- Ambiguous words
- Word senses
- Relations between senses: synonym,antonym,, hyponym, hypernym, meronym, structured polysemy
- Introduction to WordNet, gloss, synset, sense relations in WordNet.
- Cosine distance between documents.
- Word sense disambiguation.

# Introduction

- Semantic analysis is the process of extracting meaning from text.
- It permits computers to know and interpret sentences, paragraphs, or whole documents.
- It is done by analysing their grammatical structure, and distinguishing relationships between individual words in a specific context.
- It's an important sub-task of Natural Language Processing (NLP).
- It drives machine learning tools like chatbots, search engines, and text analysis.
- Semantic analysis-driven tools can facilitate industries to automatically extract meaningful information from unstructured data, like emails, comments, and feedback.

# Parts of semantic Analysis

- Semantic Analysis of Natural Language can be classified into two broad parts:

**1. Lexical Semantic Analysis:** Lexical Semantic Analysis involves understanding the meaning of each word of the text individually. It basically refers to fetching the dictionary meaning that a word in the text is deputed to carry.

**2. Compositional Semantics Analysis:** Although knowing the meaning of each word of the text is essential, it is not sufficient to completely understand the meaning of the text.

- For example, consider the following two sentences:
- Sentence 1: Students love CRCE.
- Sentence 2: CRCE loves Students.
- Although both these sentences 1 and 2 use the same set of root words {student, love, CRCE}, they convey entirely different meanings.
- Hence, under Compositional Semantics Analysis, we try to understand how combinations of individual words form the meaning of the text.

- **Example :** "Your customer service is a joke! I've been on hold for 30 minutes and counting!"
- a. Any human can understand that a customer is frustrated because a customer service agent is taking too long to respond.
- b. However, machines first need to be trained to make sense of human language and understand the context in which words are used
- c. Otherwise, they might misinterpret the word “joke” as positive.

# Lexical Semantics

- Lexical semantics plays a crucial role in semantic analysis, allowing computers to know relationships between words, phrasal verbs, etc.
- In Lexical Semantics words, sub-words, etc. are called lexical items.
- In simple terms, lexical semantics is the relationship between lexical items, meaning of sentences and syntax of sentence.
- Lexical semantics includes the following two main points :

a. Classification and Decomposition of lexical items.

b. Analyze the differences and similarities between numerous lexical-semantic structures.

# Ambiguity and word senses

- Semantic ambiguity happens when the word themselves have a variable meaning.
- Take the below example, The car hit the pole while it was moving.
- The interpretation can be
- The car (while moving) hit the pole
- The car hit the pole that was moving.
- The first interpretation is preferred over the second one because we have a model of the world that help us to distinguish what is logical (or possible) from what is not.

# Word Sense

- A sense (or word sense) is a discrete representation of one aspect of the meaning of a word.
- Generally, we represent each sense with a sub script.
- Bank$^1$ and bank$^2$, mouse$^1$ and mouse$^2$.
- In context , its easy to see the different meanings:
- Mouse$^1$ : A mouse controlling a computer system.
- Mouse$^2$: a quite animal like mouse

# Relations between senses: (IMP)

- Synonym
- Antonym
- Hyponym
- Hypernym
- Meronym
- Metonymy

# Synonyms :

- They are words that have the same sense or mostly the same meaning as another, e.g., sad, unhappy, depressed, etc. When two lexical items may be swapped out for one another in a sentence without changing its meaning or acceptability, the words are said to be synonyms.
- Example:
- How big is that plane?
- Would I be flying on a ‘large’ or a ‘small’ plane.
- Exchanging big and large in these examples has no noticiable effect on the meaning of the sentence.

# Antonyms :

- They are words that have close to opposite meanings e.g., good, bad.
- An antonym occurs when two or more lexemes have opposite meaning from one another.
- Two senses can be antonyms if they define a binary opposition or are at opposite ends of some scale
- Example: Good – bad, Big – small, Fast – Slow
- Complementary pairs: Refers to the existence of pairs that the deniel of one, implies assertion of other.
- Example : Male-Female, Alive-Dead, Present – Absent
- Another group of antonyms, **<u>reversives</u>**, describe change or movement in opposite directions, such as rise/fall or up/down

# Hoponym (Class-object)

- Hyponymys refers to a term that is an instance of a generic term.
- They can be understood by taking class-object as an analogy. For example: ‘Color‘ is a hypernymy while ‘grey‘, ‘blue‘, ‘red‘, etc, are its hyponyms.
- Another example is ‘car’ is a hyponym of ‘vehicle’
- dog is a hyponym of animal
- mango is a hyponym of fruit

# Hypernyms (Class-Object)

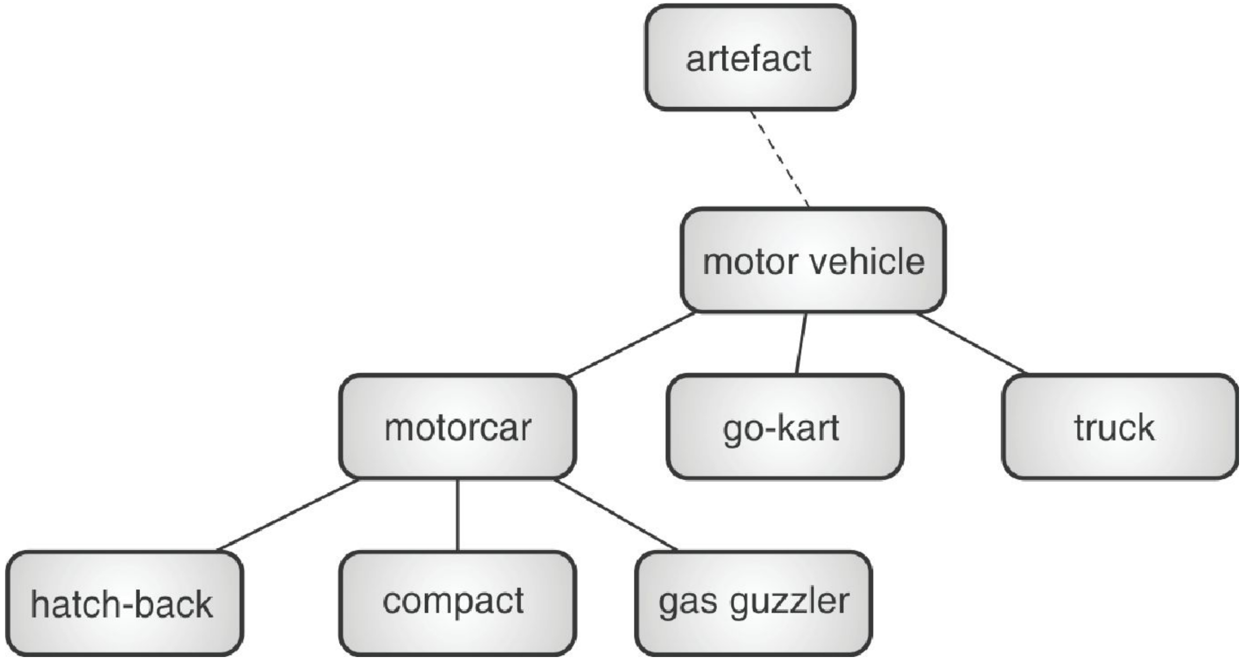
- Hypernym is a term whose meaning includes the meaning of other words, its a broad superordinate label that applies to many other members of set. .
- For example, Fruit is a hypernym for mango, Banana etc. or
- Hypernym is the sense which is a superclass :
- i. vehicle is a hypernym of car
- ii. animal is a hypernym of dog

**Hypernymy and Hyponymy**

- In simpler terms, a hyponym is in a type-of relationship with its hypernym.
- • Hypernyms and hyponyms are asymmetric.
- • Hyponymy can be tested by substituting X and Y in the sentence "X is a kind of Y" and
- determining if it makes sense.
- • For example, "A screwdriver is a kind of tool" makes sense, but not "A tool is a kind of
- screwdriver".
- • Strictly speaking, the meaning relation between hyponyms and hypernyms applies to
- lexical items of the same word class (or parts of speech), and holds between senses rather
- than words.
- • Hyponymy is a transitive relation, if X is a hyponym of Y, and Y is a hyponym of Z, then X
- is a hyponym of Z.
- • For example, violet is a hyponym of purple and purple is a hyponym of color ; therefore,
- violet is a hyponym of color.

## Hypernymy and Hyponymy

- They could be observed from top to bottom, where the higher level is more general and
- the lower level is more specific.
- • Hyponymy is the most frequently encoded relation among synsets used in lexical
- databases such as WordNet.

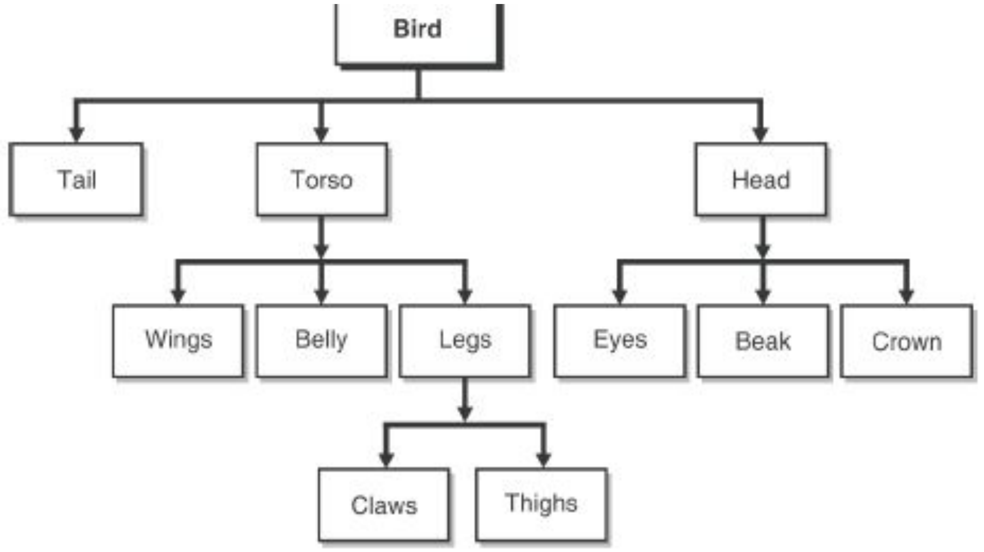


# Homonyms

- Homonymy refers to two or more lexical terms with the same spellings but completely distinct in meaning.
- For example: ‘Rose‘ might mean ‘the past form of rise‘ or ‘a flower‘, – same spelling but different meanings; hence, ‘rose‘ is a homonymy. e.g., right (correct), right (turn).
- bat (wooden stick thing) vs bat (flying scary mammal)
- bank (financial institution) vs bank (riverside)

# Meronomy (part-to-whole relation)

- A Meronym in semantic is a word that designates a component or a member of something.
- For example, apple is a meronym of apple tree (sometimes written as apple -> apple tree)
- The part-to-whole relationship is called meronym.
- Wheel‘ is a meronym of ‘Automobile.



# Polysemy

- **Polysemy** is a relationship between the meanings of words or phrases, although slightly different, they share a common basic meaning
- E.g. Polysemy refers to lexical terms that have the same spelling but multiple closely related meanings. It differs from homonymy because the meanings of the terms need not be closely related in the case of homonymy.
- For example: ‘man‘ may mean ‘the human species‘ or ‘a male human‘ or ‘an adult male human‘ – since all these different meanings bear a close association, the lexical term ‘man‘ is a polysemy.
- Bank- bank could be a river bank or financial organization

# Polysemy

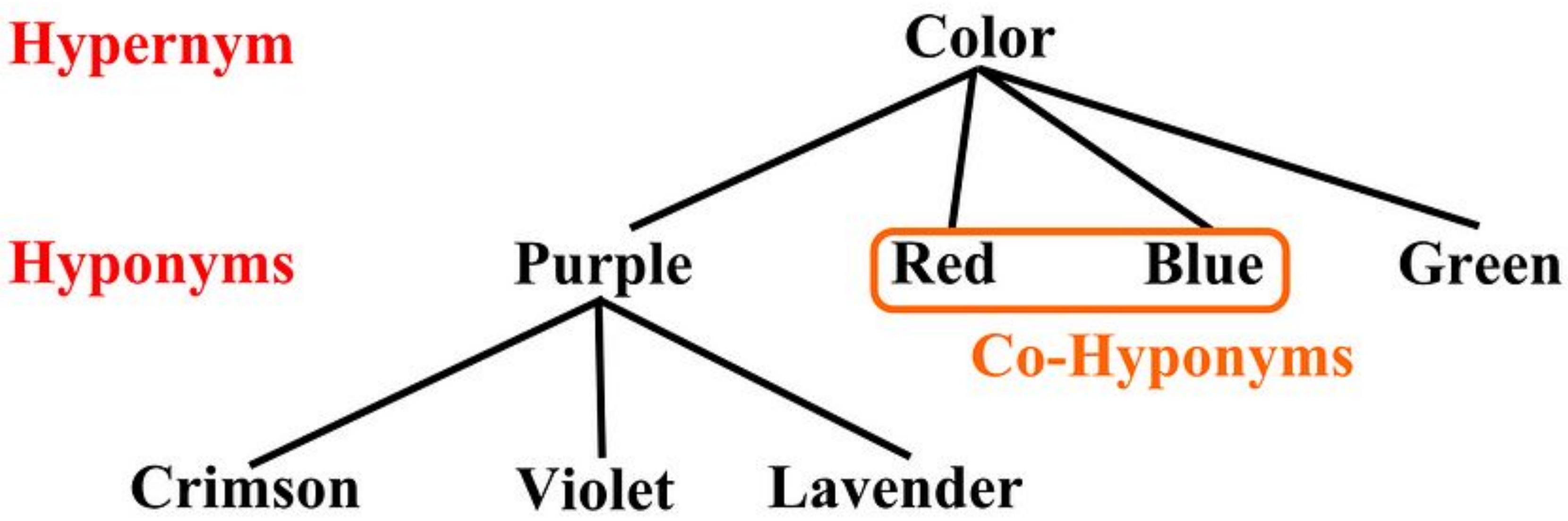
- i. He drank a glass of milk.
- ii. He forgot to milk the cow.
- iii. The enraged actor sued the newspaper.
- iv. He read the newspaper.

# Difference between Polysemy and Homonymy

| Sr. No. | Polysemy | Homonymy |
|---|---|---|
| 1. | In Polysemy two words or phrase may have many possible meanings. | In Homonymy two unrelated words that look or sound the same and have different meanings. |
| 2. | They have different, but related meanings. | They have completely different meanings. |
| 3. | They have related word origins. | They have different origins. |
| 4. | Polysemous words are entered under one entry in dictionaries. | Homonyms are entered separately in dictionaries. |
| 5. | Polysemous words can be understood if one knows the meaning of one word. | Homonyms words meaning cannot be guessed as the words have unrelated different meanings. |

# Metonymy

- Metonymy is a figure of speech in which the name of one item is substituted for the name of another to which it is related or to which it is an attribute.
- Example:
- Using crown for King/Queen : The land belonging to the 'crown'
- Using the white House for the US president – The White House today will announce the secret first hand.


Hypernym
Color
Hyponyms
Purple
Red
Blue
Green
Co-Hyponyms
Crimson
Violet
Lavender

# Introduction to WordNet

- Automatically interpreting and analysing the meaning of words and pre-processing textual input can be complex in Natural Language Processing (NLP).
- We frequently employ **lexicons** to help with this.
- We frequently link the text in our data to the lexicon, which aids us in comprehending the relationships between those terms.
- **WordNet is one such lexical resource.**
- Its unique **semantic network aids in the discovery of word relationships, synonyms, grammar, and other topics. This aids NLP tasks like sentiment analysis, automatic language translation, and text similarity.**

# WordNet

- WordNet is a **large lexical database of words, senses, and their semantic relations.**
- In WordNet, the sense is defined by a set of synonyms, called **sysnets**, that have a similar meaning or sense. This means WordNet represents words (or senses) as lists of the word senses that can be used to express the concept.
- Here is an example of synset. Sense for the word **'fool'** can be defined by the list of synonyms as {**chump, jester, gull, fritter, dupe, fool around**}
- The English WordNet consists of three separate databases, **one each for nouns and verbs and a third for adjectives and adverbs.**
- There **are several lemmas in each database, and each one is labeled with several senses.**
- **There are 117798 nouns, 11529 verbs, 22479 adjectives and 4481 adverbs in wordNet 30 release.**

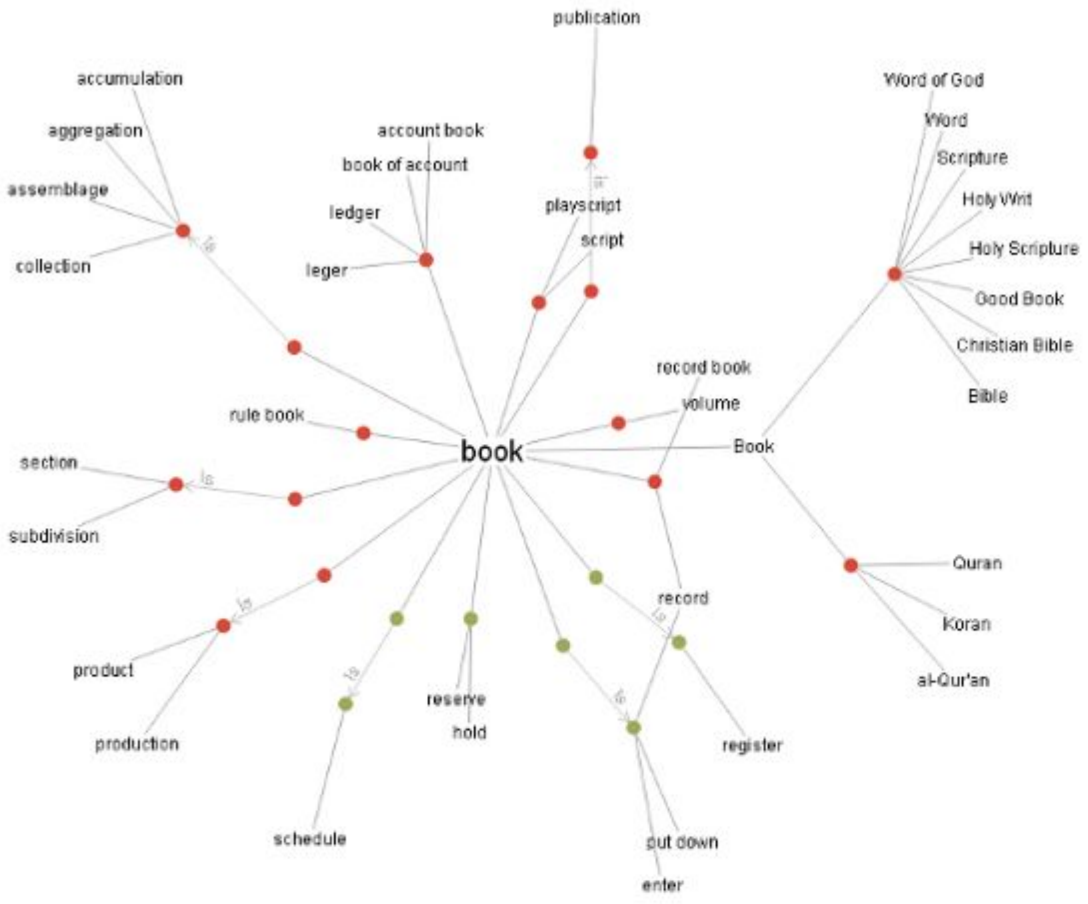
- Figure shows the lemma entry for the noun and adjective bass.

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

The adjective "bass" has 1 sense in WordNet.
1. bass$^{1}$, deep$^{6}$ - (having or denoting a low vocal or instrumental range) *"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"*

Note that there are eight senses for the noun and one for the adjective, each of gloss which has a gloss (a dictionary-style definition), a list of synonyms for the sense, and sometimes also usage examples (shown for the adjective sense). **Unlike dictionaries, WordNet doesn't represent pronunciation, so doesn't distinguish the pronunciation [b ae s] in bass4 , bass5 , and bass8 from the other senses pronounced [b ey s]**


WordNet Search - 3.1
- WordNet home page - Glossary - Help
Word to search for: climb
Search WordNet
Display Options: (Select option to change)
Change
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"
Noun
S: (n) ascent, acclivity, rise, raise, climb, upgrade (an upward slope or grade (as in a road)) "the car couldn't make it up the rise"
S: (n) climb, climbing, mounting (an event that involves rising to a higher point (as in altitude or temperature or intensity etc.))
S: (n) climb, mount (the act of climbing something) "it was a difficult climb to the top"
Verb
S: (v) climb, climb up, mount, go up (go upward with gradual or continuous progress) "Did you ever climb up the hill behind your house?"
S: (v) climb (move with difficulty, by grasping)
S: (v) wax, mount, climb, rise (go up or advance) "Sales were climbing after prices were lowered"
S: (v) climb (slope upward) "The path climbed all the way to the top of the hill"
S: (v) climb (improve one's social status) "This young man knows how to climb the social ladder"
S: (v) rise, go up, climb (increase in value or to a higher point) "prices climbed



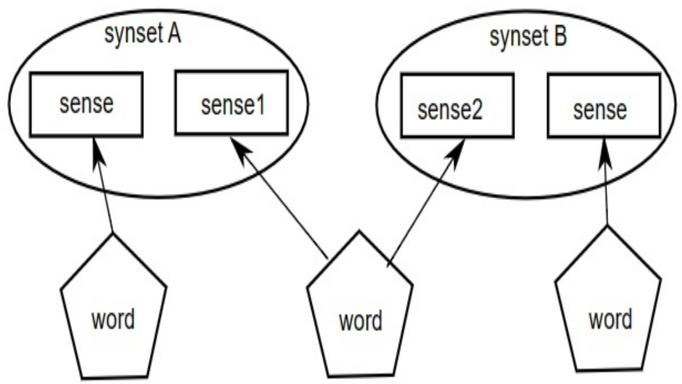
publication
accumulation
aggregation
assemblage
collection
account book
book of account
ledger
leger
playscript
script
Word of God
Word
Scripture
Holy Writ
Holy Scripture
Good Book
Christian Bible
Bible
record book
volume
Book
rule book
book
section
subdivision
product
production
schedule
reserve
hold
record
register
put down
enter
Quran
Koran
al-Qur'an
is

# Gloss

- **The Glosses are textual definitions for each sense provided by dictionaries and thesauruses.**
- Glosses are properties of a synset, so that each sense included in the synset has the same gloss and can express this concept.
- **Here are the Glosses for two senses of ‘bank’**
- **1. The financial institution that accepts deposits and channels the money into leading activities**
- **2. Sloping land (especially the slope beside a body of water)**
- **Glosses are not a formal meaning representation; they are just written for people.**

# Synset

- WordNet categorizes English words into synonyms, referred to as **Synsets** (short for a set of synonyms).
- The set of near-synonyms for a WordNet sense is called a synset.
- synsets are an important primitive in WordNet.\
- **Every Synset contains a name, a part-of-speech (nouns, verbs, adverbs, and adjectives), and a number.**
- Synsets are used to store synonyms, where each word in the Synset shares the same meaning. Essentially, each Synset is a collection of synonyms.
- **Some words have just one Synset, while others have multiple Synsets. Every Synset has a definition associated with it.**
- **Synset makes it easier for users to look up words in the WordNet database.**



# The Structure of a Wordnet

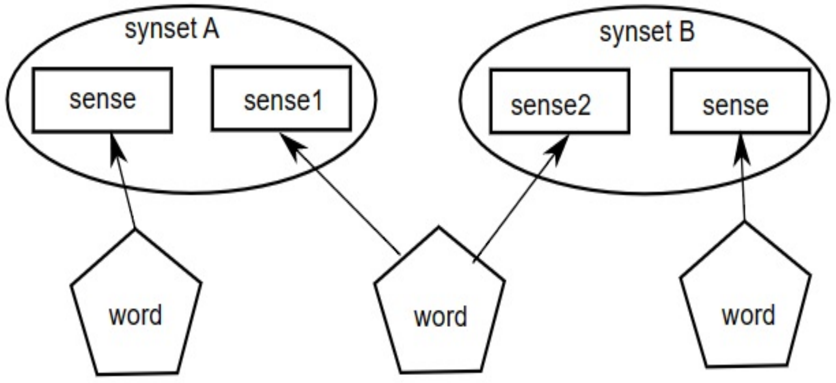
- A **wordnet** is an online lexicon which is organized by concepts.
- The basic unit of a wordnet is the synonym set (**synset**), a group of words that all refer to the same concept.
- Words and synsets are linked by means of conceptual-semantic relations to form the structure of wordnet.

# Words, Senses, and Synsets

- We all know that **words** are the basic building blocks of languages.
- A word is built up with two parts, its form and its meaning.
- But in natural languages, the word form and word meaning are not in an elegant one-to-one match, one word form may connect to many different meanings, so hereforth, we need **senses**, to work as the unit of word meanings, for example, the word '*bank*' has at least two senses:

1. bank$^1$: financial institution, like *City Bank*;
2. bank$^2$: sloping land, like *river bank*;

- Since **synsets** are group of words sharing the same concept, bank$^1$ and bank$^2$ are members of two different synsets, although they have the same word form.
- On the other hand, different word forms may also convey the same concept, such as '*cab*' and '*taxi*', these word forms with the same concept are grouped together into one synset.



# Sense Relations in WordNet

- WordNet also represents relations between senses, like the IS-A relation between dog and mammal or the part-whole relationship between car and engine. WordNet represents all the kinds of sense relations discussed in the previous section (Synonym, antonym, Meronym….etc)

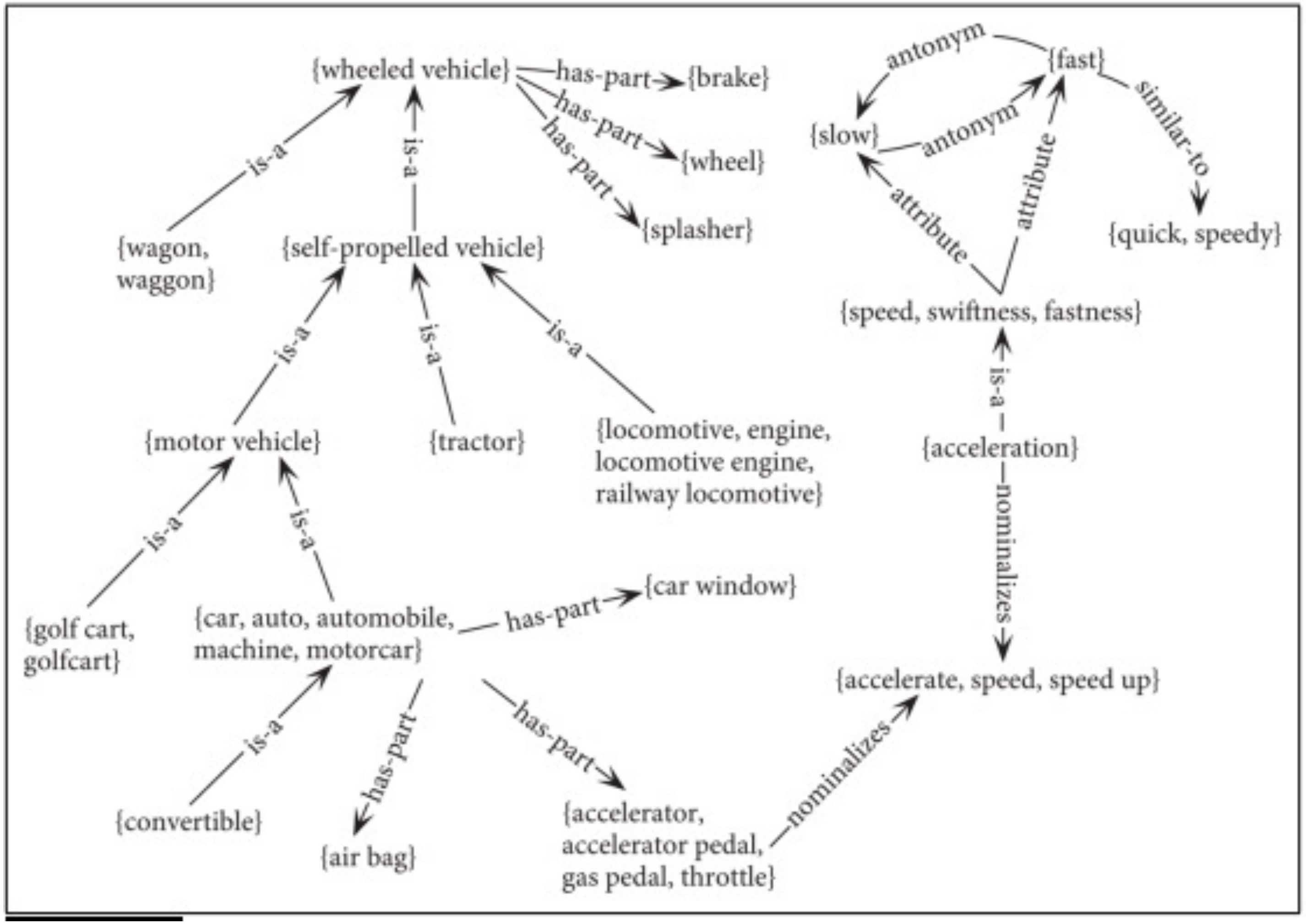
| Relation | Also Called | Definition | Example |
|---|---|---|---|
| Hypernym | Superordinate | From concepts to superordinates | $\textit{breakfast}^1 \rightarrow \textit{meal}^1$ |
| Hyponym | Subordinate | From concepts to subtypes | $\textit{meal}^1 \rightarrow \textit{lunch}^1$ |
| Instance Hypernym | Instance | From instances to their concepts | $\textit{Austen}^1 \rightarrow \textit{author}^1$ |
| Instance Hyponym | Has-Instance | From concepts to their instances | $\textit{composer}^1 \rightarrow \textit{Bach}^1$ |
| Part Meronym | Has-Part | From wholes to parts | $\textit{table}^2 \rightarrow \textit{leg}^3$ |
| Part Holonym | Part-Of | From parts to wholes | $\textit{course}^7 \rightarrow \textit{meal}^1$ |
| Antonym | | Semantic opposition between lemmas | $\textit{leader}^1 \Longleftrightarrow \textit{follower}^1$ |
| Derivation | | Lemmas w/same morphological root | $\textit{destruction}^1 \Longleftrightarrow \textit{destroy}^1$ |

Some of the noun relations in WordNet.

| Relation | Definition | Example |
|---|---|---|
| Hypernym | From events to superordinate events | $fly^9 \rightarrow travel^5$ |
| Troponym | From events to subordinate event | $walk^1 \rightarrow stroll^1$ |
| Entails | From verbs (events) to the verbs (events) they entail | $snore^1 \rightarrow sleep^1$ |
| Antonym | Semantic opposition between lemmas | $increase^1 \Longleftrightarrow decrease^1$ |

Some verb relations in WordNet.

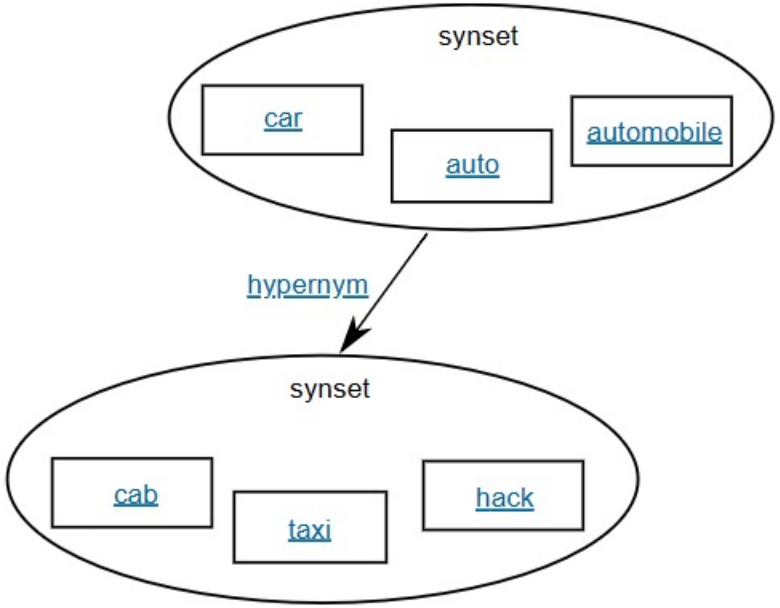
For example WordNet represents hyponymy by relating each synset to its immediately more general and more specific synsets through direct hypernym and hyponym relations.
These relations can be followed to produce longer chains of more general or more specific synsets.




WordNet viewed as a graph

# Synset Relations

- In wordnet, synsets are linked with each other to form various kinds of relations.
- For example, if the concept expressed by a synset is more general than a given synset, then it is in a *hypernym* relation with the given synset.
- As shown in the figure, the synset with *car*, *auto* and *automobile* as its member is the *hypernym* of the other synset with *cab*, *taxi* and *hack*. Such relation which is built on the synset level is categorized as synset relations.



# WordNet Hypernym Hierarchy for “bass”

- S: (n) **bass**, basso (an adult male singer with the lowest voice)
  - *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) singer, vocalist, vocalizer, vocaliser (a person who sings)
      - S: (n) musician, instrumentalist, player (someone who plays a musical instrument (as a profession))
        - S: (n) performer, performing artist (an entertainer who performs a dramatic or musical work for an audience)
          - S: (n) entertainer (a person who tries to please or amuse)
            - S: (n) person, individual, someone, somebody, mortal, soul (a human being) *"there was too much for one person to do"*
              - S: (n) organism, being (a living thing that has (or can develop) the ability to act or function independently)
                - S: (n) living thing, animate thing (a living (or once living) entity)
                  - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                    - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                      - S: (n) physical entity (an entity that has physical existence)
                        - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

# The Distinction Between WordNET and Thesaurus

- **Where thesaurus is helping us in finding the synonyms and antonyms of the words the WordNET is helping us to do more than that.**
- WordNET interlinks the specific sense of the words wherein thesaurus links words by their meaning only.
- In the WordNET the words are semantically disambiguated if they are in close proximity to each other.
- Thesaurus provides a level to the words in the network if the words have similar meaning but in the case of WordNET, we get levels of words according to their semantic relations which is a better way of grouping the words.

# Cosine distance between documents

In NLP, calculating the similarity between two text documents is a popular task with a variety of real-world use. For example, text similarity can be used for ranking the results of a search engine or suggesting related content to readers.

## What is Text Similarity?

Text Similarity is the process of comparing a piece of text with another and finding the similarity between them. It’s basically about determining the degree of closeness of the text.

Document/Text similarity can be used in…

1. Search engines need to model the relevance of a document to a query, beyond the overlap in words between the two. For instance, question-and-answer sites such as Quora or Stack Overflow need to determine whether a question has already been asked before.
2. Selecting the most similar product for a customer shopping in any online platform if that exact product is unavailable.
3. Checking similarity of multiple documents or letters.
4. Choosing the most appropriate or closest job role or profile a person’s resume.

- Why text similarity is challenging. Let’s have a look at two sentences.
- 1. “The teacher gave his speech to an empty room”
- 2. “There was almost no body when the professor was talking”
- Despite having an extremely similar meanings, they are spelled entirely differently. The only word shared by these two sentences is ‘the’.
- We would like to have a sophisticated algorithm that returns a high similarity score for this pair of sentences.
- Semantic text similarity is what we refer to when we want to calculate similarity based on the meaning.
- Traditional text similarity algorithm only functions on a lexical level. Depending on the use case, they can offer better trade-off and are quicker and easy to implement.

- To start with text similarity task, we need to convert the input text in a given document into a more machine-readable form by transforming the text into embeddings which then gets converted into vectors that are understood by the machine to calculate the similarity.
- The objective is to create a vector space where, using a selected similarity metric, documents that are similar to one another are near.

# Algorithms to transform the text into embeddings (or vectors)

**1) Word count or frequency count method:**

- Using <u>word count</u> is the easiest method for <u>creating a vector</u> from a document.
- Consider the following three sentences.

1. We went to the pizza place and you ate no pizza at all
2. I ate pizza with you yesterday at home
3. There’s no place like home.

- To build the vector, we first count the occurrences of each word in a sentence.

| Document 1 | Document2 | Document3 |
| --- | --- | --- |
| We – 1 | I – 1 | There’s – 1 |
| Went -1 | Ate – 1 | No -1 |
| To -1 | Pizza – 1 | Place -1 |
| The – 1 | With -1 | Like -1 |
| Pizza-2 | You – 1 | Home -1 |
| Place - 1 | …. | …. |
| …. | …. | |
| … | | |

# Numerical:

Given two documents

D1: “digital information is new technology”

D2: “digital technology changes the world”

Represent the above two documents in the form vector using word count method. Compute cosine similarity between the two documents.

| | Digital | Information | Is | New | technology | Change | The | World |
|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 2 | 1 | 1 | 1 | 0 | 0 | 0 |
| D2 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Let D1 is represented by vector A= [1,2,1,1,1,0,0,0]

and D2 is represented by vector B = [1,0,0,0,1,1,1,1]

A= [1,2,1,1,1,0,0,0]
B = [1,0,0,0,1,1,1,1]

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

A.B = (1x1 + 2x0 + 1x0 + 1x0 + 1x1 + 0x1 + 0x1 +0x1) = 2

|A| = $\sqrt{8} =$
|B| = $\sqrt{5} =$

Similarity = 2 / [$\sqrt{8}$ x $\sqrt{5}$ ] = ___________ Ans

# 2) TF-IDF method

- The problem with the previous method(word count) is that all words in the phrase are treated with the same importance, there is no option to prioritize the words.
- The most often used words like ‘the’, ‘an’, ‘is’, are the ones which are less important. These are the stop words. We can eliminate stop words in pre-processing step.
- To overcome this problem, TF-IDF was introduced as a numeric approach that gives a certain importance measure to the word embeddings.

- Term Frequency - Inverse Document Frequency (TF-IDF) is a widely used statistical method in natural language processing and information retrieval.
- It measures how important a term is within a document relative to a collection of documents.
- Words within a text document are transformed into importance numbers by a text vectorization process.
- TF-IDF gets this importance score by multiplying  the term’s frequency (TF) and the term inverse document frequency (IDF).
- The higher the TF-IDF score the rarer the term in a document and the higher its importance.

- **Term Frequency (TF) :** TF of a term or word is the number of times the term appears in a document compared to the total number of words in the document.

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

- **Inverse Document Frequency (IDF):**
- IDF of a term reflects the proportion of documents in the corpus that contain the term. Words unique to a small percentage of documents (e.g., technical jargon terms) receive higher importance values than words common across all documents (e.g., a, the, and).

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

The TF-IDF of a term is calculated by multiplying TF and IDF scores.

$$TF\text{-}IDF = TF * IDF$$

- In simple language, Importance of a term is high when it occurs a lot in a given document and rarely in others. In short, commonality within a document measured by TF is balanced by rarity between documents measured by IDF. The resulting TF-IDF score reflects the importance of a term for a document in the corpus.

## Numerical: TF-IDF

## Idf (word) = log( N / [d €C :word € d] )

N – total number of documents in the corpus.

C – the corpus.

d – the number of documents in which the word appears.

For the previous example, if we consider three documents.

Doc 1 - We went to the pizza place and you ate no pizza at all

Doc 2 - I ate pizza with you yesterday at home

Doc 3 - There’s no place like home.

The word ‘Pizza’ will have IDF as

Log(3/2) = 0.4

And the word ‘yesterday’ will have an IDF as

Lod(3/1) = 1.1

For each word in our corpus, the IDF is calculated once as a preprocessing step and it will tell you how significant that word is inside the corpus.

At this point , we compute the document vectors by balancing it with the IDF rather than using the raw word counts.

The final score of each word can be calculated as,

Score(word) = frequency(word) x idf(word)

# Numerical Example

Imagine the term $t$ appears 20 times in a document that contains a total of 100 words. Term Frequency (TF) of $t$ can be calculated as follow:

$$TF = \frac{20}{100} = 0.2$$

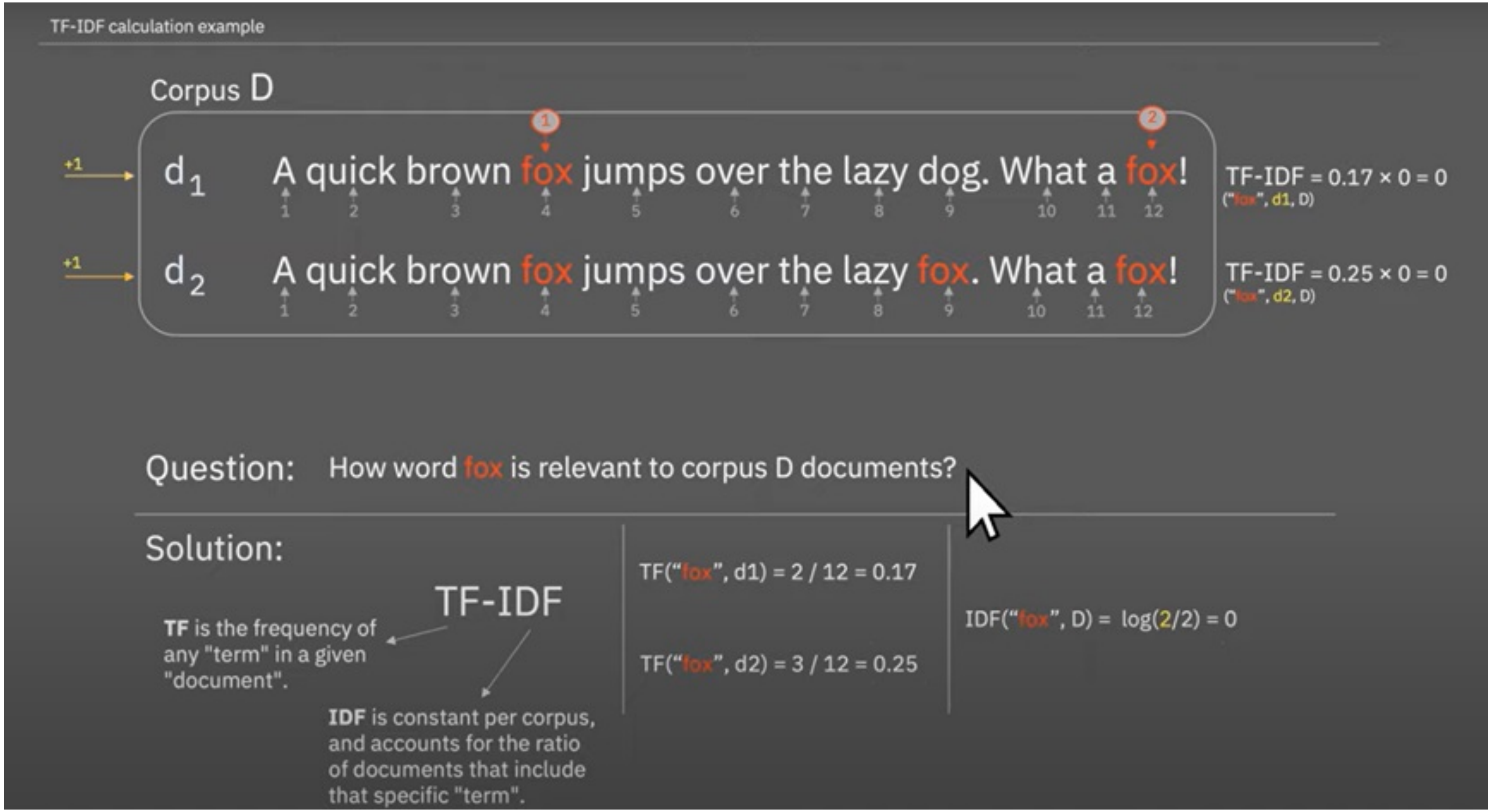
Assume a collection of related documents contains 10,000 documents. If 100 documents out of 10,000 documents contain the term $t$, Inverse Document Frequency (IDF) of $t$ can be calculated as follows

$$IDF = log\frac{10000}{100} = 2$$

Using these two quantities, we can calculate TF-IDF score of the term $t$ for the document.

$$TF\text{-}IDF = 0.2 * 2 = 0.4$$

# TF-IDF Numerical



# Example of TF-IDF calculation

| Document/Term | T1 | T2 | T3 | T4 | T5 | T6 |
|---|---|---|---|---|---|---|
| **D1** | 5 | 9 | 4 | 0 | 5 | 6 |
| **D2** | 0 | 8 | 5 | 3 | 10 | 8 |
| **D3** | 3 | 5 | 6 | 6 | 5 | 0 |
| **D4** | 4 | 6 | 7 | 8 | 4 | 4 |

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of $i$ in $j$
$df_i$ = number of documents containing $i$
$N$ = total number of documents

$$TF-IDF(T1\ in\ D1) = 5 * \log\left(\frac{4}{3}\right) = 0.625$$

$$TF-IDF(T2\ in\ D1) = 9 * \log\left(\frac{4}{4}\right) = 0.0$$

$$TF-IDF(T3\ in\ D1) = 4 * \log\left(\frac{4}{4}\right) = 0.0$$

$$TF-IDF(T4\ in\ D1) = 0 * \log\left(\frac{4}{3}\right) = 0.0$$

$$TF-IDF(T5\ in\ D1) = 5 * \log\left(\frac{4}{4}\right) = 0.0$$

$$TF-IDF(T6\ in\ D1) = 6 * \log\left(\frac{4}{3}\right) = 0.7496$$

$$TF-IDF(T1\ in\ D2) = 0 * \log\left(\frac{4}{3}\right) = 0.0$$

Given two documents

A : “Jupiter is the largest planet”

B: “Mars is the fourth planet form the earth”. Represent both the document in a vector form using TF-IDF

**Step 1: Find TF (Term freq) for each term for each document in a corpus**

$$TF(w,d) = \frac{occurences\ of\ w\ in\ document\ d}{total\ number\ of\ words\ in\ document\ d}$$

| Words | TF (for A) | TF (for B) |
|---|---|---|
| Jupiter | 1/5 | 0 |
| Is | 1/5 | 1/8 |
| The | 1/5 | 2/8 |
| largest | 1/5 | 0 |
| Planet | 1/5 | 1/8 |
| Mars | 0 | 1/8 |
| Fourth | 0 | 1/8 |
| From | 0 | 1/8 |
| Sun | 0 | 1/8 |

# Step 2 : Calculate IDF for each term

IDF (w,C) = log ( Total number of documents in a given corpus C) / No. of documents containing w)

| Words | TF (for A) | TF (for B) | IDF |
|---|---|---|---|
| Jupiter | 1/5 | 0 | ln(2/1) = 0.69 |
| Is | 1/5 | 1/8 | ln(2/2) = 0 |
| The | 1/5 | 2/8 | ln(2/2) = 0 |
| largest | 1/5 | 0 | ln(2/1) = 0.69 |
| Planet | 1/5 | 1/8 | ln(2/2) = 0 |
| Mars | 0 | 1/8 | ln(2/1) = 0.69 |
| Fourth | 0 | 1/8 | ln(2/1) = 0.69 |
| From | 0 | 1/8 | ln(2/1) = 0.69 |
| Sun | 0 | 1/8 | ln(2/1) = 0.69 |

# Step 3 – Calculate TF-IDF score for each term

TF –IDF (w,d,C) = TF(w,D) x IDF (w,C)

| Words | TF (for A) | TF (for B) | IDF | TFIDF (A) | TFIDF (B) |
|---|---|---|---|---|---|
| Jupiter | 1/5 | 0 | ln(2/1) = 0.69 | 0.138 | 0 |
| Is | 1/5 | 1/8 | ln(2/2) = 0 | 0 | 0 |
| The | 1/5 | 2/8 | ln(2/2) = 0 | 0 | 0 |
| largest | 1/5 | 0 | ln(2/1) = 0.69 | 0.138 | 0 |
| Planet | 1/5 | 1/8 | ln(2/2) = 0 | 0.138 | 0 |
| Mars | 0 | 1/8 | ln(2/1) = 0.69 | 0 | 0.086 |
| Fourth | 0 | 1/8 | ln(2/1) = 0.69 | 0 | 0.086 |
| From | 0 | 1/8 | ln(2/1) = 0.69 | 0 | 0.086 |
| Sun | 0 | 1/8 | ln(2/1) = 0.69 | 0 | 0.086 |

Vector representation of documents
A = [0.138,0,0,0.138,0.138,0,0,0,0]
B = [0,0,0,0,0,0.086, 0.086, 0.086, 0.086, 0.086]

# Metrics to calculate the similarity between embeddings

- Once we have our document vectors ready, we can use any similarity measure to find similarities between two documents.
- Cosine similarity is the most commonly used one such measure.
- Other measures are J***accard Similarity and Euclidean Distance***

# Cosine similarity

- Cosine similarity measures the similarity between two vectors of an inner product space in general.
- It measures the cosine of the angle between two embeddings and determines whether they are pointing in roughly the same direction or not.
- When the embeddings are pointing in the same direction the angle between them is zero so their cosine similarity is 1.
- When the embeddings are perpendicular to each other the angle between them is 90 degrees and the cosine similarity is 0 finally when the angle between them is 180 degrees the cosine similarity is -1

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Cosine Similarity ranges from 1 to -1, where 1 represents most similar and -1 represents least similar.

If we apply it to our example vectors we get

Similarity(doc1, doc2) = 0.45

Similarity(doc1, doc3) = 0.23

Similarity(doc2, doc3) = 0.15

We can say that, the first two documents resemble.

# Problem based on IDF [Home Assignment]

We have a document database of five documents with the following content.

D1: “Information Retrieval System”

D2: “Information Storage”

D3: “Digital Speech Synthesis System”

D4: ”Speech filtering”

D5: “Speech retrieval”

a) Calculate Term frequency (TF) matrix.

b) Calculate inverse document frequency (IDF) for each term.

- Calculate TF-IDF for the term “Speech” and explain how the term “**Speech**” is related to all the documents in a given corpus. Which document is most relevant to the term “Speech”?

# Word sense disambiguation

- The task of selecting the **correct sense for a word** is called word sense disambiguation, or WSD.
- WSD algorithms take as input a word in context and a fixed inventory of potential word senses and, outputs the correct word sense in context.
- The input and the senses, depend on the task.
- For machine translation from English to Spanish, the sense tag inventory for an English word might be the set of different Spanish translations.
- For automatic indexing of medical articles, the sense-tag inventory might be the set of MeSH (Medical Subject Headings) thesaurus entries.
- There are mainly two WSD tasks: Lexical sample and all – word.
- In the **lexical sample** task, a small pre-selected set of target words is chosen, along with an inventory of senses for each word from some lexicon. Since the set of words and the set of senses are small, simple supervised classification approaches are used.

- In the all-words task, systems are given entire texts and a lexicon with an inventory of senses for each entry and are required to disambiguate every content word in the text.
- The all-words task is similar to part-of-speech tagging, except with a much larger set of tags since each lemma has its own set.
- A consequence of this larger set of tags is data sparseness; it is unlikely that adequate training data for every word in the test set will be available. Moreover, given the number of polysemous words in reasonably sized lexicons, approaches based on training one classifier per term are unlikely to be practical.

# Approaches for Word Sense Disambiguation

- Supervised Machine Learning
- Unsupervised
- Knowledge based or Dictionary based approach

# Supervised

- The Supervised method is based on a annotated data/ corpus.
- The supervised approach is based on trained sense annotated corpus to build classifiers. Initially, annotated corpus is required to build a classifier.
- The classifier is used to recognized the sense of the word based on their context of use. This approach assumes that the context can provide enough evidence on its own to disambiguate words.
- A major disadvantage of this approach is he requirement of a large sense-tagged corpora.
- The accuracy of the classifier depends on the size and the variety of the data incorporated in the corpus. Larger the corpus, better the results. The creation of corpus itself is a challenge in terms of time and money

- Some common techniques used in supervised WSD include:

1. Decision list: A decision list is a set of rules that are used to assign a sense to a target word based on the context in which it appears.
2. Neural Network: Neural networks such as feedforward networks, recurrent neural networks, and transformer networks are used to model the context-sense relationship.
3. Support Vector Machines: SVM is a supervised machine learning algorithm used for classification and regression analysis.
4. Naive Bayes: Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to classify text into predefined categories.
5. Decision Trees: Decision Trees are a flowchart-like structure in which an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome.

- **Supervised WSD Exploiting Glosses:** Textual definitions are a prominent source of information in sense inventories (also known as glosses). Definitions, which follow the format of traditional dictionaries, are a quick and easy way to clarify sense distinctions
- **Purely Data-Driven WSD:** In this case, a token tagger is a popular baseline model that generates a probability distribution over all senses in the vocabulary for each word in a context.
- **Supervised WSD Exploiting Other Knowledge**: Additional sources of knowledge, both internal and external to the knowledge base, are also beneficial to WSD models. Some researchers use BabelNet translations to fine-tune the output of any WSD system by comparing the output senses' translations to the target's translations provided by an NMT system.

# Unsupervised

- The Unsupervised approach is based on unannotated corpora.
- It is based on clustering of words. It assumes that sense of the ambiguous word can be induced from the neighboring words.
- The underlying assumption is that similar senses occur in similar contexts, and thus senses can be induced from the text by clustering word occurrences using some measure of similarity of context.
- Since this approach is based on unannotated data, it is very essential to have most of the senses of the word in the training corpus.
- Some algorithms based on this approach are: Context Group Discrimination, Co-occurrence graphs, WSD using parallel corpora.

# Knowledge Based

- 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..
- Lesk Algorithm is the classical algorithm based on Knowledge-Based WSD.
- Lesk algorithm assumes that words in a given “neighborhood” (a portion of text) will have a similar theme. The dictionary definition of an uncertain word is compared to the terms in its neighborhood in a simplified version of the Lesk algorithm.

# Applications of Word Sense Disambiguation (WSD)

- Machine Translation
- Machine translation or MT is the most obvious application of WSD. In MT, Lexical choice for the words that have distinct translations for different senses, is done by WSD. The senses in MT are represented as words in the target language. Most of the machine translation systems do not use explicit WSD module.
- Information Retrieval (IR)
- Information retrieval (IR) may be defined as a software program that deals with the organization, storage, retrieval and evaluation of information from document repositories particularly textual information. The system basically assists users in finding the information they required but it does not explicitly return the answers of the questions. WSD is used to resolve the ambiguities of the queries provided to IR system. As like MT, current IR systems do not explicitly use WSD module and they rely on the concept that user would type enough context in the query to only retrieve relevant documents.

- ext Mining and Information Extraction (IE)
- In most of the applications, WSD is necessary to do accurate analysis of text. For example, WSD helps intelligent gathering system to do flagging of the correct words. For example, medical intelligent system might need flagging of “illegal drugs” rather than “medical drugs”
- Lexicography
- Lexicography
- WSD and lexicography can work together in loop because modern lexicography is corpusbased. With lexicography, WSD provides rough empirical sense groupings as well as statistically significant contextual indicators of sense.

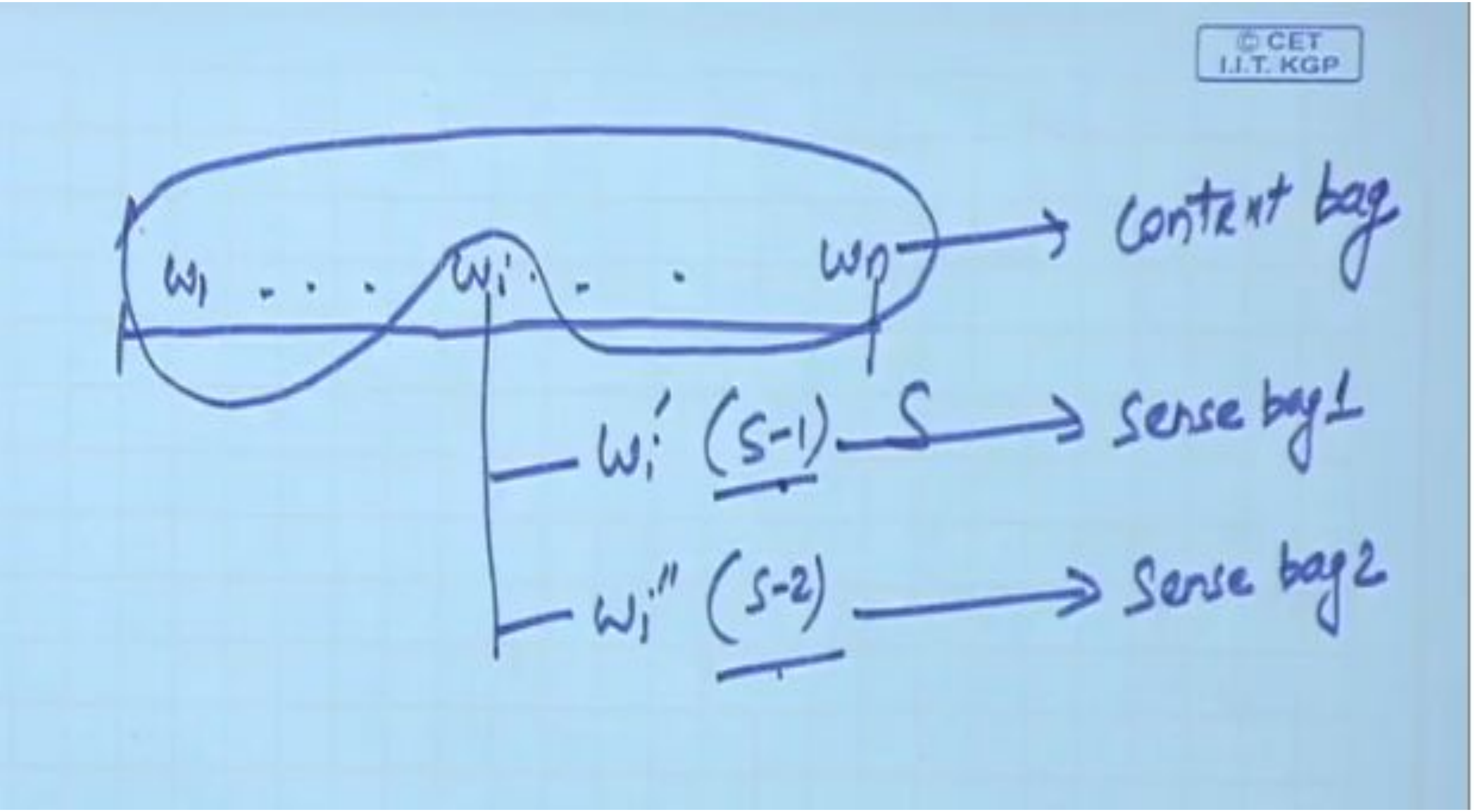
# Difficulties in Word Sense Disambiguation (WSD)

- Followings are some difficulties faced by word sense disambiguation (WSD) −
- Differences between dictionaries
- The major problem of WSD is to decide the sense of the word because different senses can be very closely related. Even different dictionaries and thesauruses can provide different divisions of words into senses.
- Different algorithms for different applications
- Another problem of WSD is that completely different algorithm might be needed for different applications. For example, in machine translation, it takes the form of target word selection; and in information retrieval, a sense inventory is not required.
- Inter-judge variance
- Another problem of WSD is that WSD systems are generally tested by having their results on a task compared against the task of human beings. This is called the problem of interjudge variance.
- Word-sense discreteness
- Another difficulty in WSD is that words cannot be easily divided into discrete submeanings.

# *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.



© CET
I.I.T. KGP
$w_1 \ldots w_i \ldots w_n$
content bag
$w_i'$ (S-1)
sense bag 1
$w_i''$ (S-2)
sense bag 2

# *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.*

*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 ***burnt*** 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

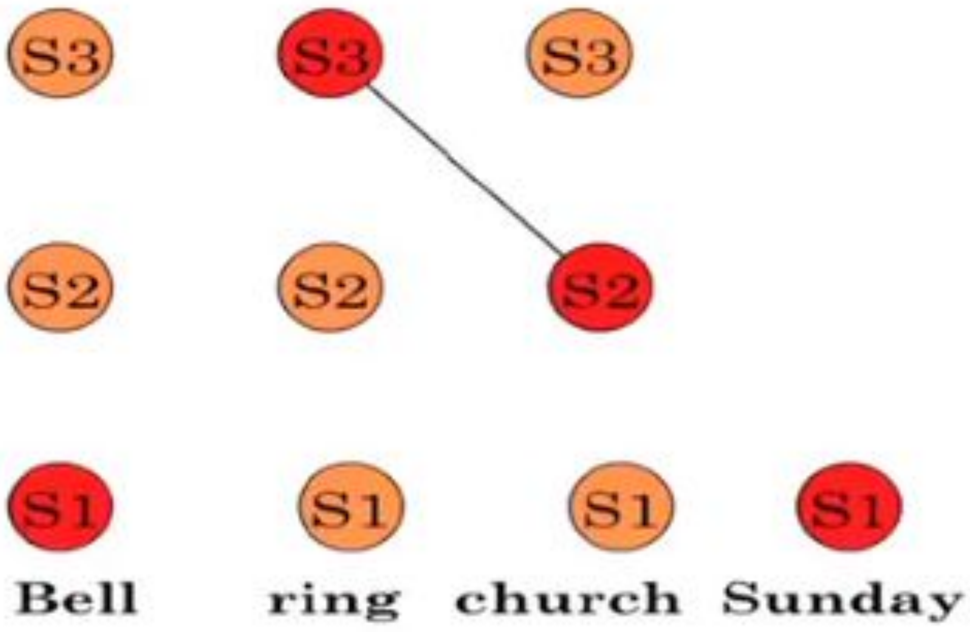
- A Thesaurus Based approach
- **Step 1:** For each sense of the target word find the thesaurus category to which that sense belongs
- **Step 2:** Calculate the score for each sense by using the context words. *A context word will add 1 to the score of the sense if the thesaurus category of the word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

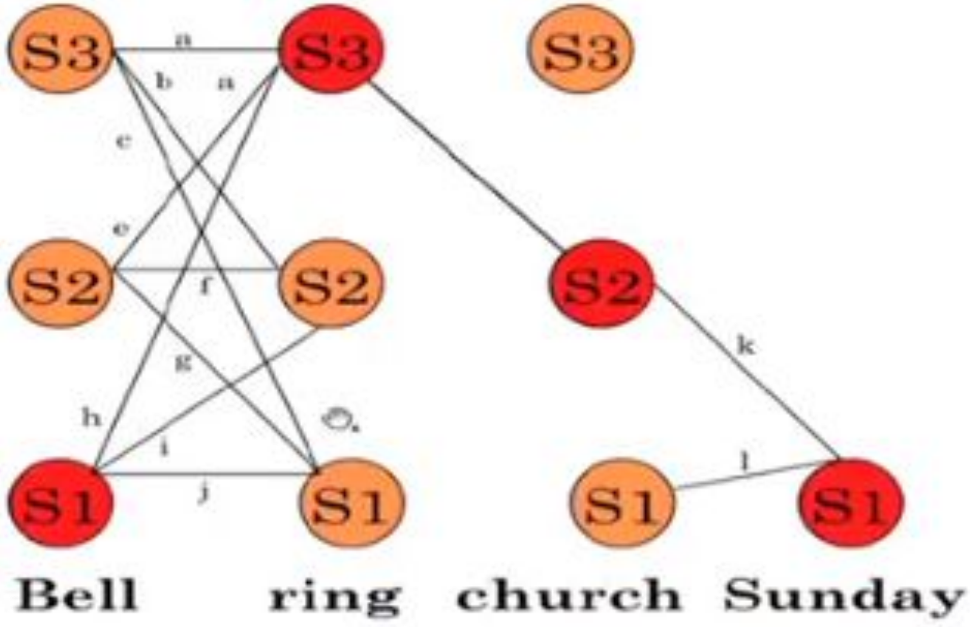
The **church bells** no longer **rung** on **Sundays**.

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

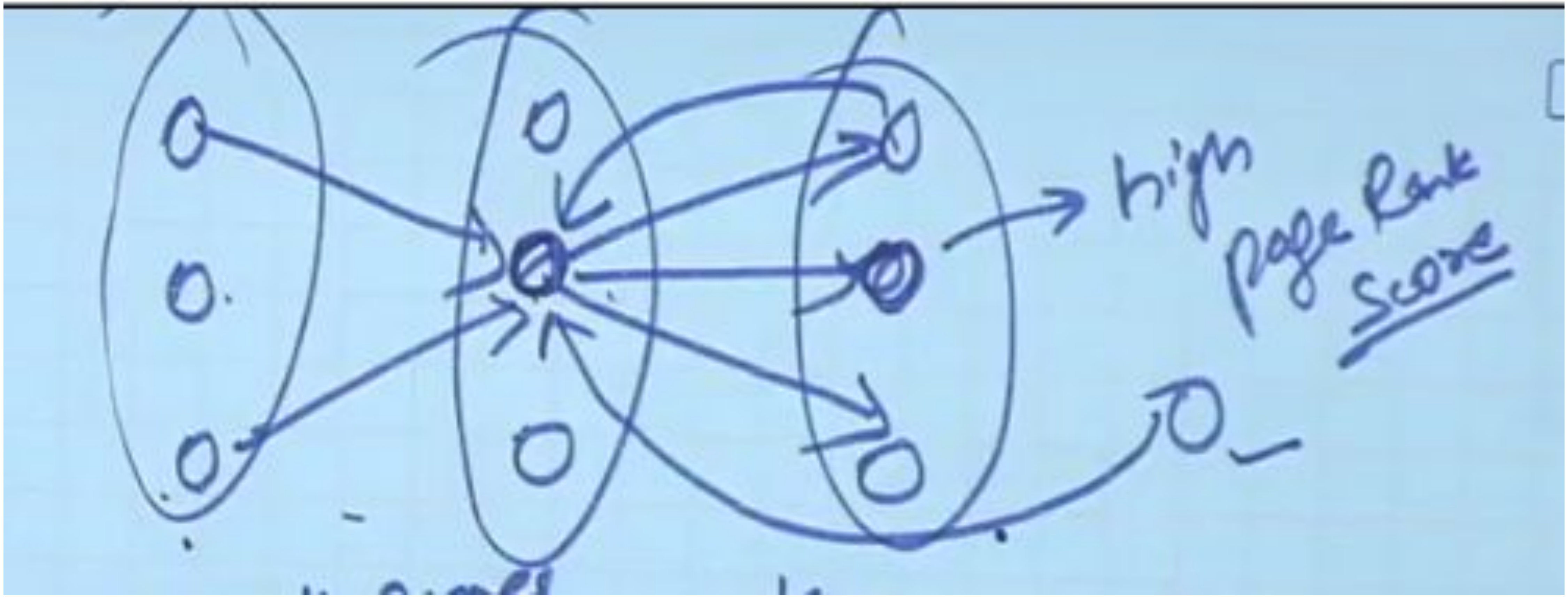
bell
1: a hollow device made of metal that makes a ringing sound when struck
2: a push button at an outer door that gives a ringing or buzzing signal when pushed
3: the sound of a bell

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).


high
page Rank
Score

# WSD Using Random Walk Algorithm

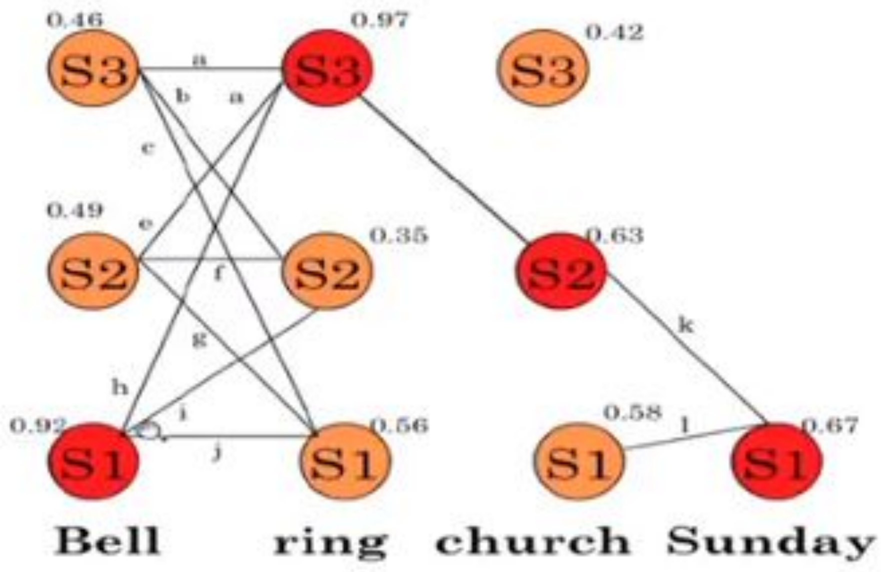
The **church bells** no longer **rung** on **Sundays**.

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).

# WSD Using Random Walk Algorithm

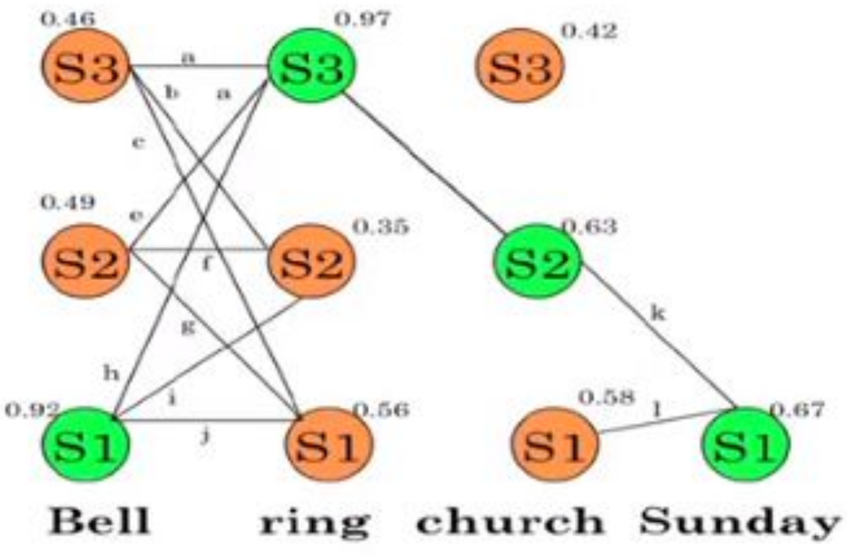
The **church bells** no longer **rung** on **Sundays**.

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

# *Naïve Bayes for WSD*

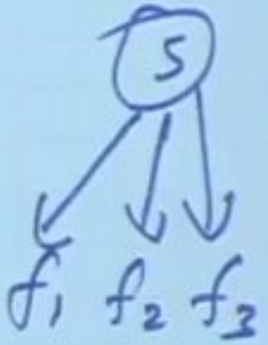
- A Naïve Bayes classifier chooses the most likely sense for a word given the features of the context:

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



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

$$= \arg\max_{s} \frac{P(f|s)\,P(s)}{P(f)}$$




$$= \arg\max_{s} \underline{P(s)}\,\underline{P(f|s)}$$

$f_1 \ldots f_n$

$$= \arg\max_{s} P(s) \prod_{i=1}^{n} P(f_i|s)$$

context

## *Training for Naïve Bayes*

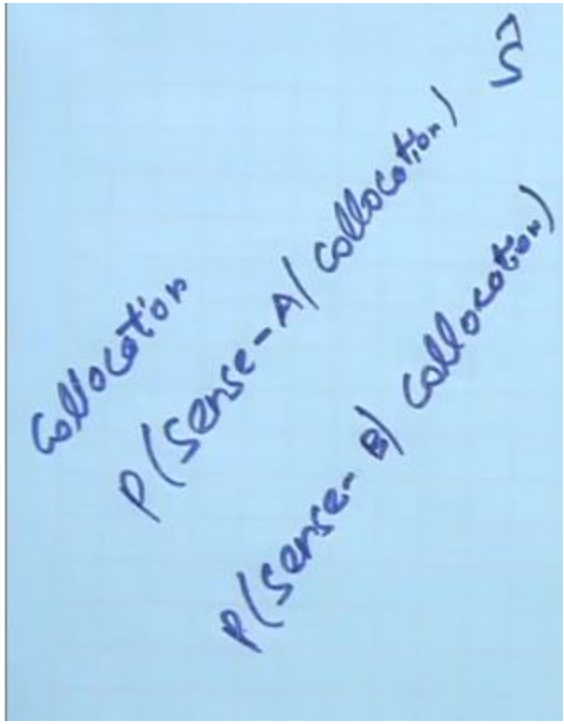
- '$f$' is a feature vector consisting of:
  - POS of $w$
  - Semantic and Syntactic features of $w$
  - Collocation vector (set of words around it) $\rightarrow$ next word (+1), +2, -1, -2 and their POS's
  - Co-occurrence vector
- Set parameters of Naïve Bayes using maximum likelihood estimation (MLE)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

# 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 |