Unit 5

NLP Applications (Continued) :

(Disclaimer: All content is from various internet sources and for academic purpose only)

Machine translation - Basic issues in MT.

There are several challenges that language translation poses for machine translation systems:

1. Ambiguity: Language is often ambiguous, and the same word or phrase can have multiple meanings depending on the context in which it is used. This can make it difficult for machine translation systems to accurately translate words and phrases.

2. Idioms and colloquialisms: Idioms and colloquialisms are expressions that are unique to a particular language or culture, and they do not have a direct equivalent in other languages. This can make it difficult for machine translation systems to accurately translate these expressions.

3. Linguistic diversity: Machine translation also has to deal with the diversity and complexity of natural languages, which can vary in terms of syntax, morphology, semantics, pragmatics, and style. For example, translating between languages that have different word orders, such as English and Japanese, may require reordering the words or phrases to preserve the meaning and coherence.

4. Cultural differences: Language is closely tied to culture, and different cultures have different ways of expressing ideas and concepts. This can make it difficult for machine translation systems to accurately translate cultural references and concepts.

5. Data quality: Machine translation relies on large amounts of parallel data, which are pairs of sentences in different languages that have the same meaning. However, parallel data can be scarce, noisy, or biased, depending on the source and domain. For example, translating between low-resource languages, such as Kannada, Telugu or Urdu may require collecting and cleaning data from various sources, such as web pages, subtitles, or dictionaries. Moreover, parallel data may reflect cultural or political biases, such as gender stereotypes, propaganda, or censorship, which can affect the accuracy and fairness of machine translation.

6. Language morphology: We may find more spelling variations in morphological rich language. - In morphological rich language, multiple words are combined together, which effects the word alignment in parallel corpus

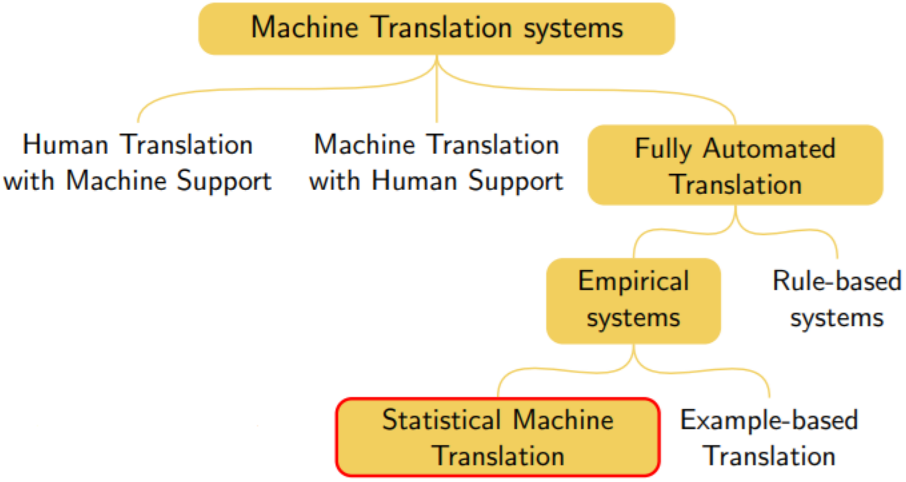
7. Word sense disambiguation(WSD)- Sometimes one word can have multiple meanings. So, it creates difficulty in aligning the source word with the target word.

8. Evaluation metrics : Machine translation also faces the challenge of evaluating the quality and usefulness of the output, which can be subjective and dependent on various factors, such as the task, the domain, the user, and the reference. For example, measuring the similarity between the output and a human reference translation using metrics such as BLEU or METEOR may not capture the adequacy or fluency of the output, or the diversity or creativity of the translation.

9. The language can be divided into 6 categories based on the subject, object and verb order in the language. Among them, 3 categories are most popular

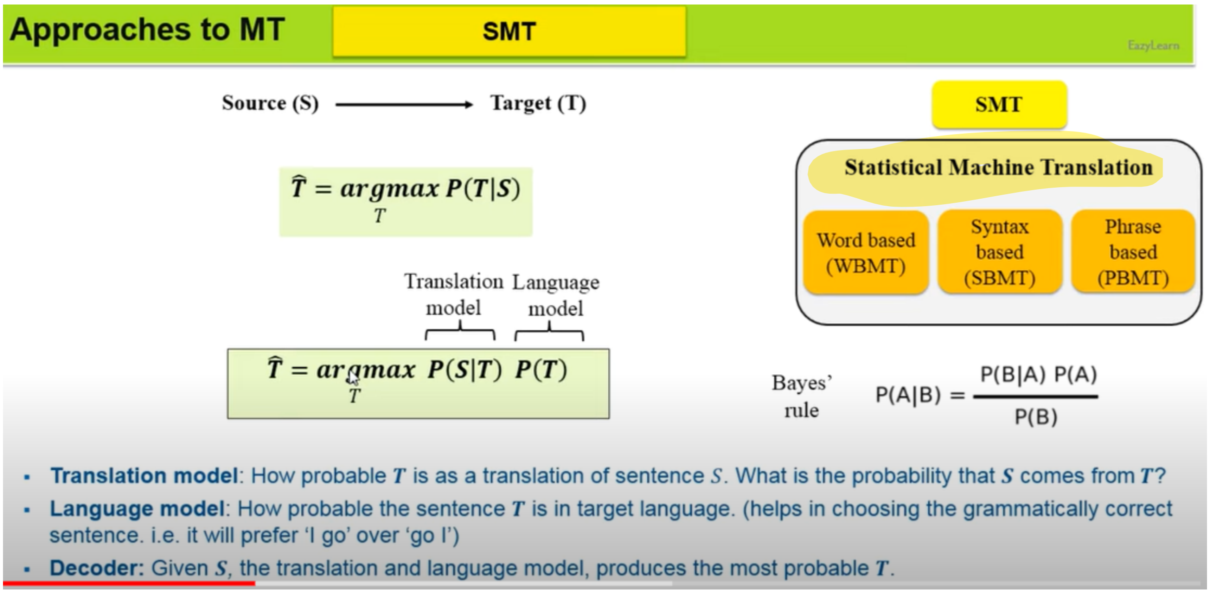
Subject-Object-Verb(SOV) language -Indian languages
Subject-Verb-Object(SVO) language- English & European lang
VSO language- Hebrew, Arabic
Statistical translation faces difficulty in translating the language from one language family to another language family. We need to perform long distance reordering for translation between different families of language. Syntax based translation is used for capturing word order.

Statistical translation


Machine Translation systems
Human Translation with Machine Support
Machine Translation with Human Support
Fully Automated Translation
Empirical systems
Rule-based systems
Statistical Machine Translation
Example-based Translation

Introduction

- Parallel corpora are available in several language pairs
- Basic idea is to use parallel corpus as a training set of translation examples



Approaches to MT **SMT**

EazyLearn

Source (S) → Target (T)

$$\hat{T} = \underset{T}{argmax}\, P(T|S)$$

$$\hat{T} = \underset{T}{argmax}\ \overbrace{P(S|T)}^{\text{Translation model}}\ \overbrace{P(T)}^{\text{Language model}}$$

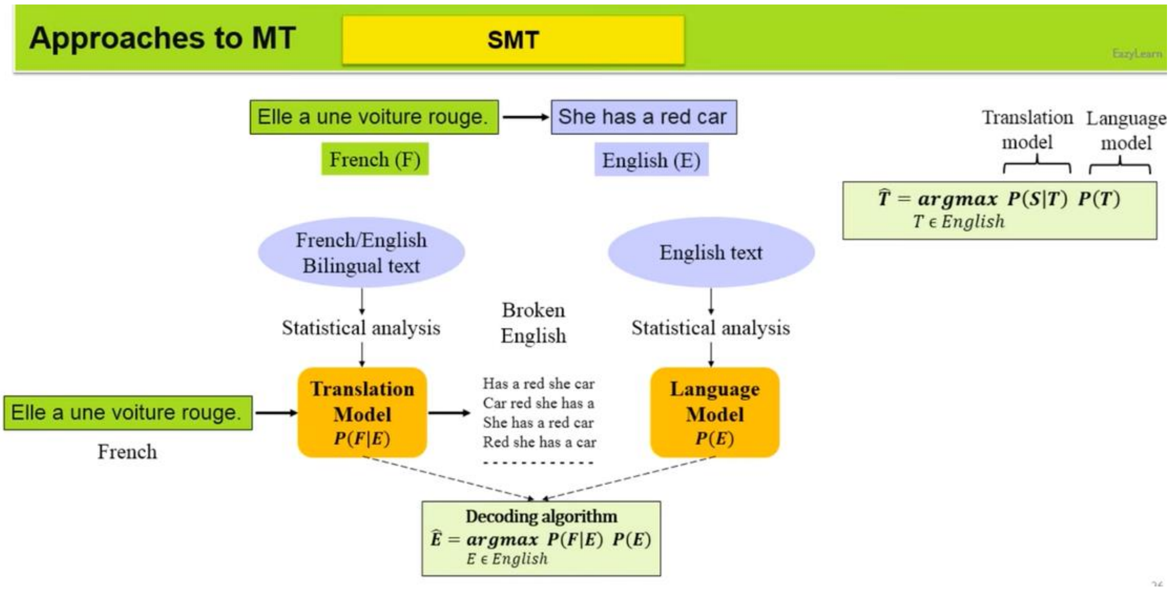
SMT

Statistical Machine Translation

Word based (WBMT) | Syntax based (SBMT) | Phrase based (PBMT)

Bayes' rule $P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$

- **Translation model**: How probable T is as a translation of sentence S. What is the probability that S comes from T?
- **Language model**: How probable the sentence T is in target language. (helps in choosing the grammatically correct sentence. i.e. it will prefer 'I go' over 'go I')
- **Decoder:** Given S, the translation and language model, produces the most probable T.



Approaches to MT — SMT

Elle a une voiture rouge. → She has a red car

French (F) — English (E)

$$\hat{T} = \underset{T \in English}{argmax}\ \overbrace{P(S|T)}^{\text{Translation model}}\ \overbrace{P(T)}^{\text{Language model}}$$

French/English Bilingual text → Statistical analysis → Translation Model $P(F|E)$

English text → Statistical analysis → Language Model $P(E)$

Elle a une voiture rouge. (French) → Translation Model $P(F|E)$ → Broken English:

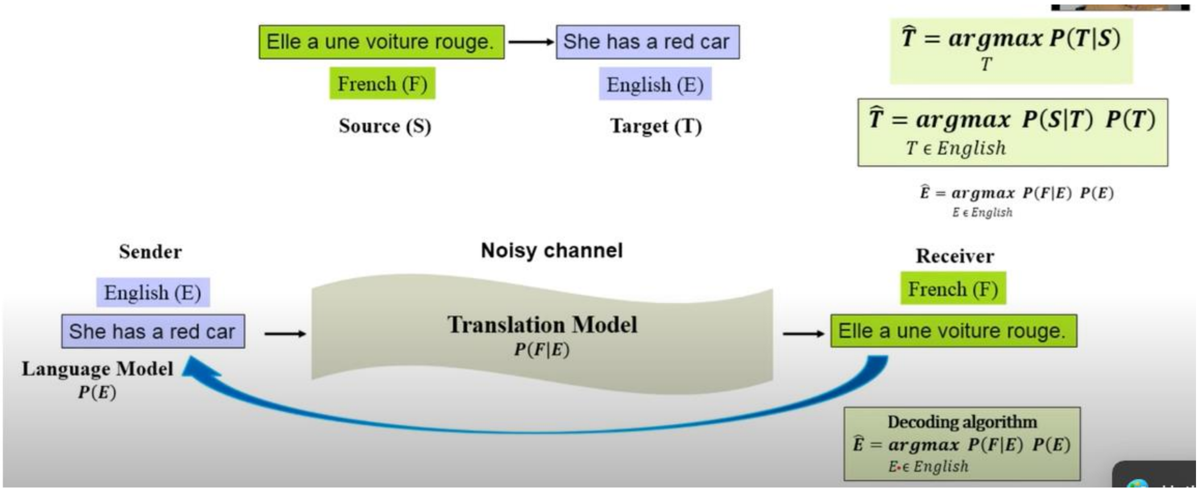
Has a red she car
Car red she has a
She has a red car
Red she has a car

Decoding algorithm

$$\hat{E} = \underset{E \in English}{argmax}\ P(F|E)\ P(E)$$

The Noisy Channel Model

- Goal: Translation system from French to English or Tamil to Hindi (Source to target language)



Elle a une voiture rouge. → She has a red car

French (F) — English (E)

Source (S) — Target (T)

$$\hat{T} = \underset{T}{argmax}\ P(T|S)$$

$$\hat{T} = \underset{T \in English}{argmax}\ P(S|T)\ P(T)$$

$$\hat{E} = \underset{E \in English}{argmax}\ P(F|E)\ P(E)$$

Sender — English (E) — She has a red car — Language Model $P(E)$

Noisy channel — Translation Model $P(F|E)$

Receiver — French (F) — Elle a une voiture rouge.

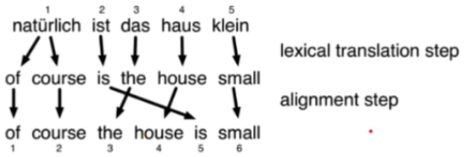
Decoding algorithm

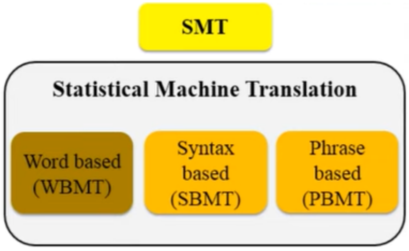
$$\hat{E} = \underset{E \in English}{argmax}\ P(F|E)\ P(E)$$

Types of SMT

WBMT:

- Word is the unit of translation


1 2 3 4 5
natürlich ist das haus klein
of course is the house small
of course the house is small
1 2 3 4 5 6
lexical translation step
alignment step

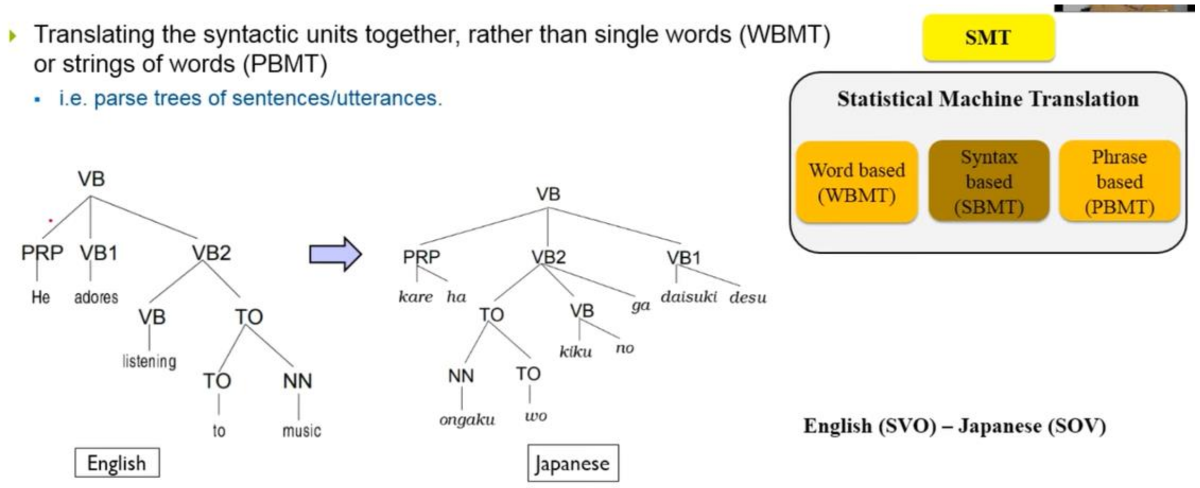


SMT

Statistical Machine Translation

Word based (WBMT) | Syntax based (SBMT) | Phrase based (PBMT)

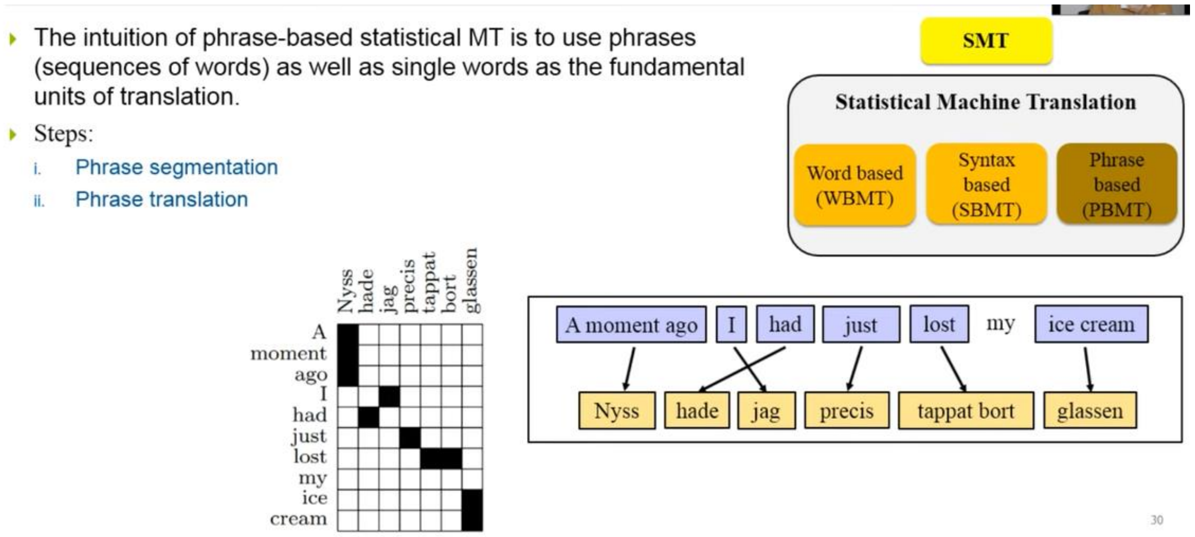
SBMT:



- Translating the syntactic units together, rather than single words (WBMT) or strings of words (PBMT)
  - i.e. parse trees of sentences/utterances.

English (SVO) – Japanese (SOV)

PBMT:

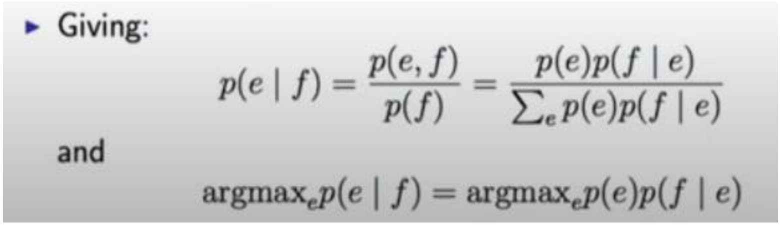


- The intuition of phrase-based statistical MT is to use phrases (sequences of words) as well as single words as the fundamental units of translation.
- Steps:
  i. Phrase segmentation
  ii. Phrase translation

	Nyss	hade	jag	precis	tappat	bort	glassen
A	■						
moment	■						
ago	■						
I			■				
had		■					
just				■			
lost					■	■	
my							
ice							■
cream							■

Noise based translation

- Have a model p(e|f) which estimates conditional probability of any English sentence e given the French sentence f. Use the training corpus to set the parameters.
- A noisy channel model has two components
  P(e) – the language model
  P(f|e) the translation model



- Giving:

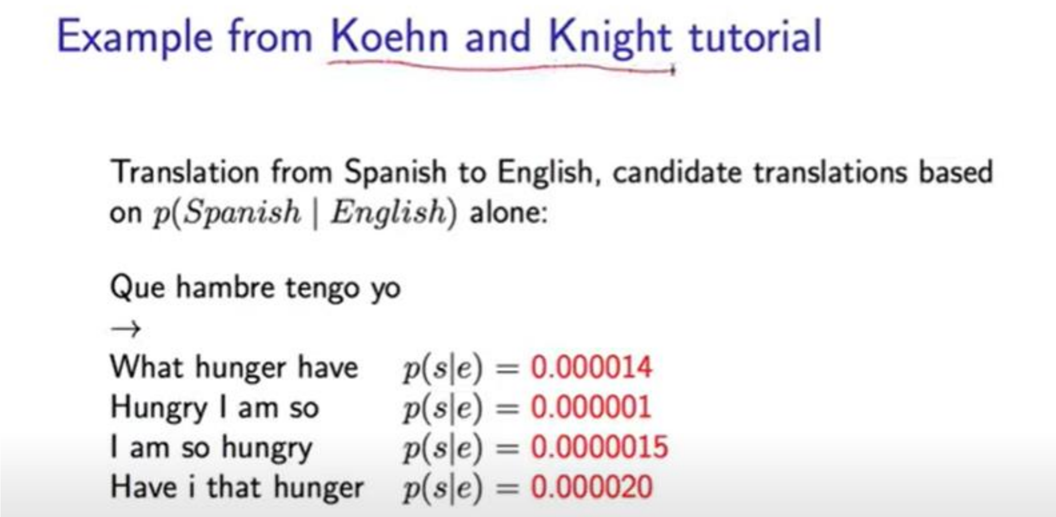
$$p(e \mid f) = \frac{p(e,f)}{p(f)} = \frac{p(e)p(f \mid e)}{\sum_e p(e)p(f \mid e)}$$

and

$$\text{argmax}_e p(e \mid f) = \text{argmax}_e p(e)p(f \mid e)$$

P(e) language model could be a trigram model, estimated from any data (parallel coupus not needed to estimate the parameters)

The translation model p(f|e) is trained from a parallel corpus of French/English pairs.



Example from Koehn and Knight tutorial

Translation from Spanish to English, candidate translations based on $p(Spanish \mid English)$ alone:

Que hambre tengo yo

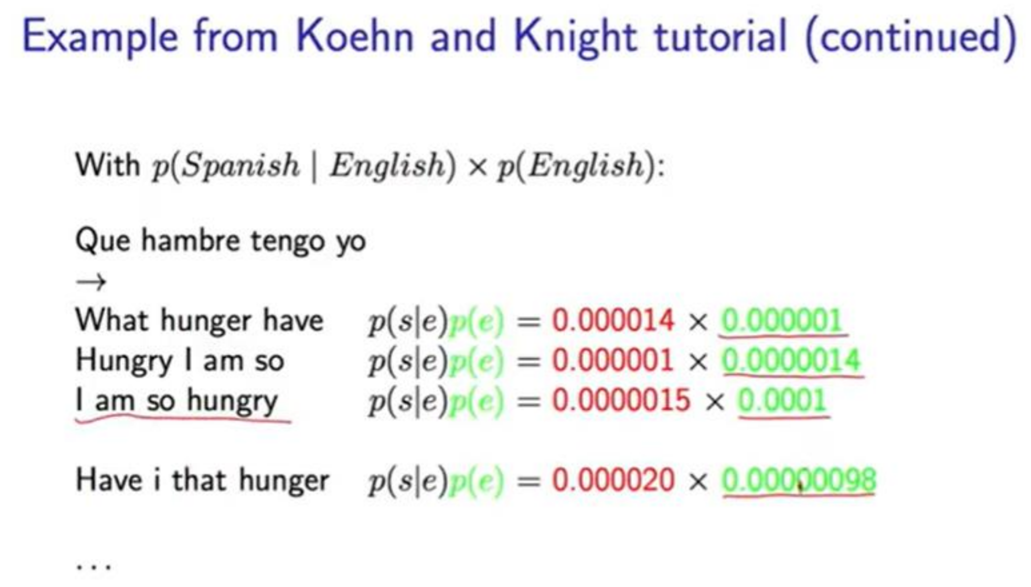
$\rightarrow$

What hunger have $p(s|e) = 0.000014$

Hungry I am so $p(s|e) = 0.000001$

I am so hungry $p(s|e) = 0.0000015$

Have i that hunger $p(s|e) = 0.000020$



Example from Koehn and Knight tutorial (continued)

With $p(Spanish \mid English) \times p(English)$:

Que hambre tengo yo

$\rightarrow$

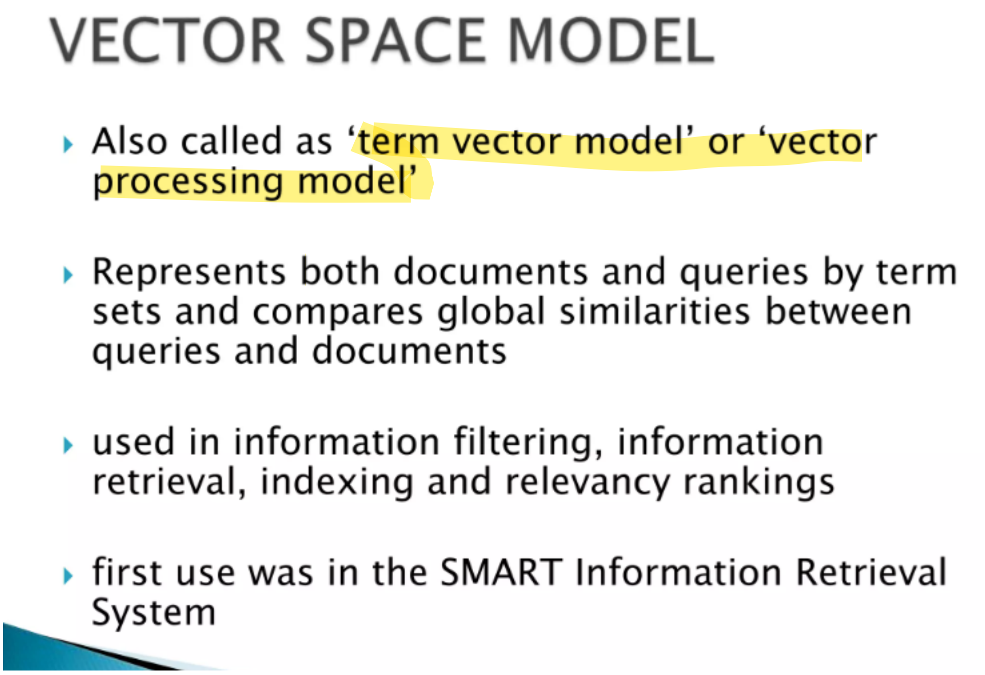
What hunger have $p(s|e)p(e) = 0.000014 \times 0.000001$

Hungry I am so $p(s|e)p(e) = 0.000001 \times 0.0000014$

I am so hungry $p(s|e)p(e) = 0.0000015 \times 0.0001$

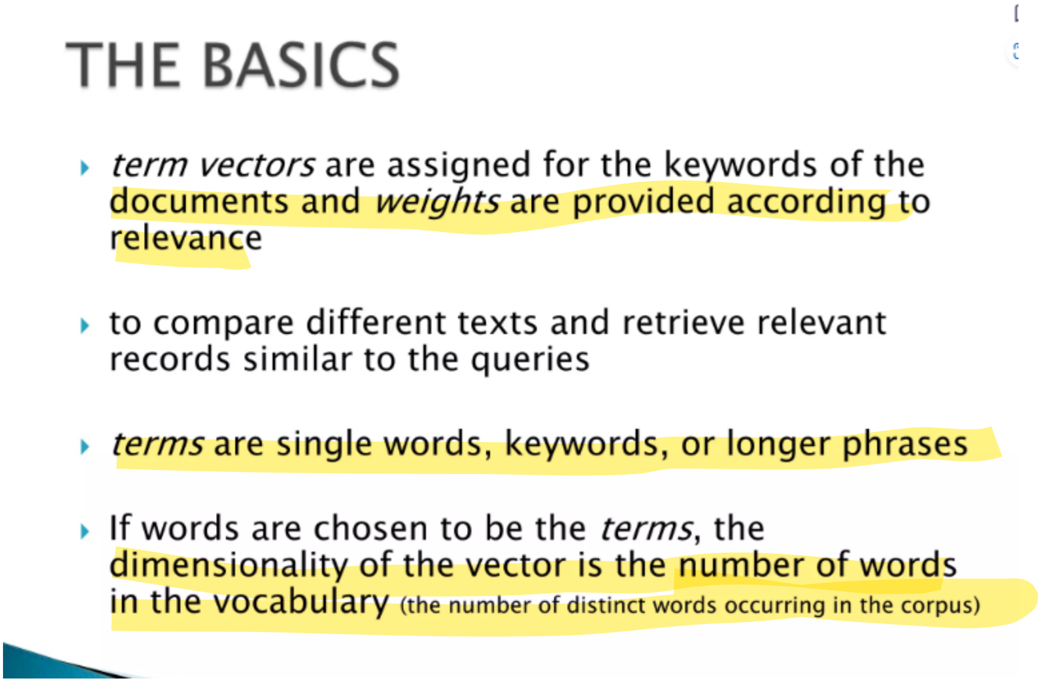
Have i that hunger $p(s|e)p(e) = 0.000020 \times 0.00000098$

...



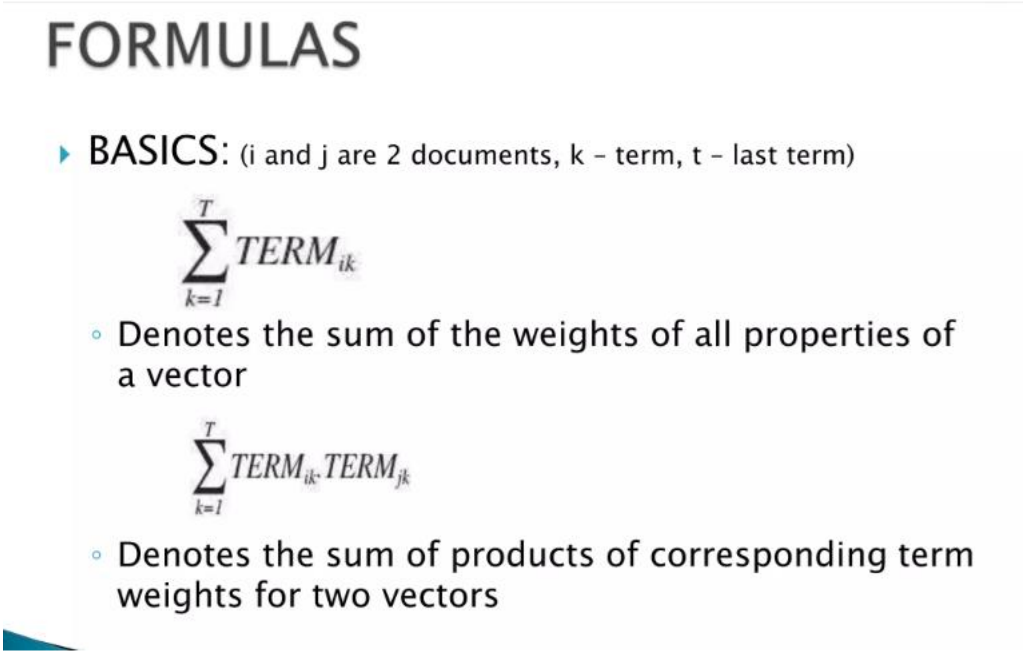
VECTOR SPACE MODEL

- Also called as ‘term vector model’ or ‘vector processing model’
- Represents both documents and queries by term sets and compares global similarities between queries and documents
- used in information filtering, information retrieval, indexing and relevancy rankings
- first use was in the SMART Information Retrieval System



THE BASICS

- *term vectors* are assigned for the keywords of the documents and *weights* are provided according to relevance
- to compare different texts and retrieve relevant records similar to the queries
- *terms* are single words, keywords, or longer phrases
- If words are chosen to be the *terms*, the dimensionality of the vector is the number of words in the vocabulary (the number of distinct words occurring in the corpus)



FORMULAS

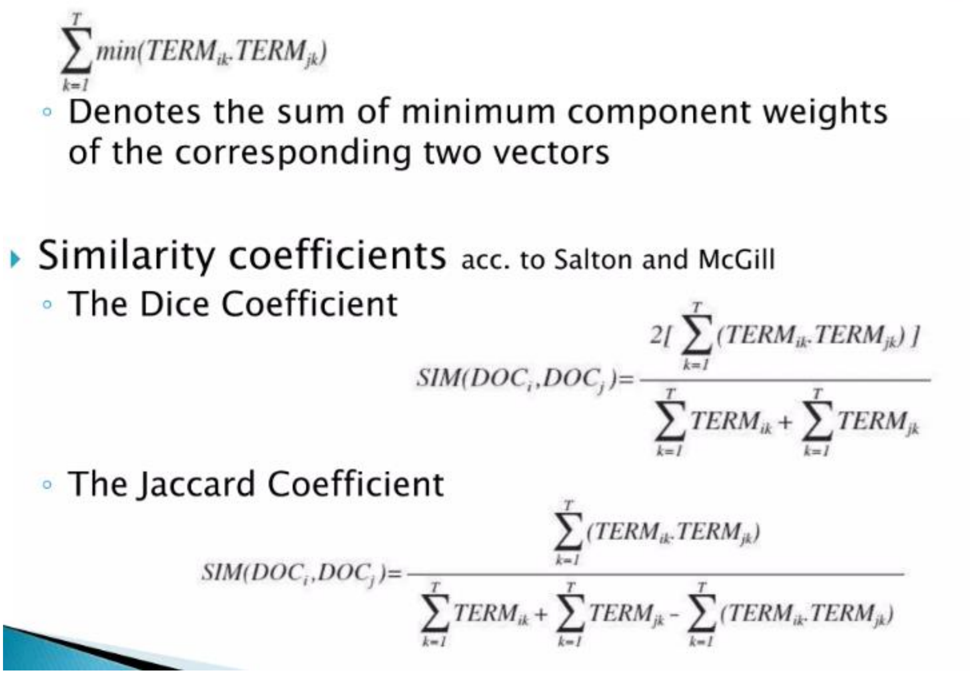
- BASICS: (i and j are 2 documents, k - term, t - last term)

$$\sum_{k=1}^{T} TERM_{ik}$$

  - Denotes the sum of the weights of all properties of a vector

$$\sum_{k=1}^{T} TERM_{ik}.TERM_{jk}$$

  - Denotes the sum of products of corresponding term weights for two vectors



$$\sum_{k=1}^{T} min(TERM_{ik}.TERM_{jk})$$

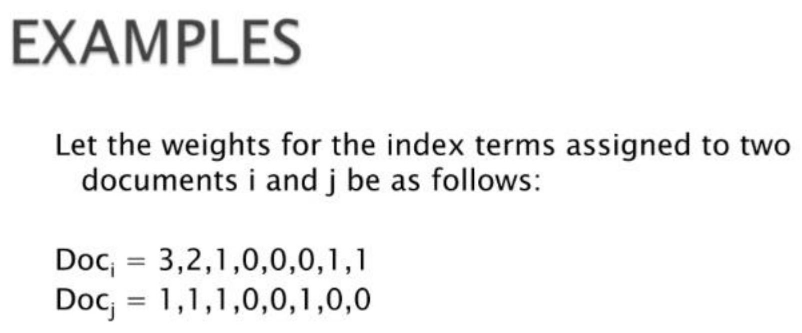
- Denotes the sum of minimum component weights of the corresponding two vectors

- Similarity coefficients acc. to Salton and McGill
  - The Dice Coefficient

$$SIM(DOC_i,DOC_j)=\frac{2[\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})]}{\sum_{k=1}^{T} TERM_{ik}+\sum_{k=1}^{T} TERM_{jk}}$$

  - The Jaccard Coefficient

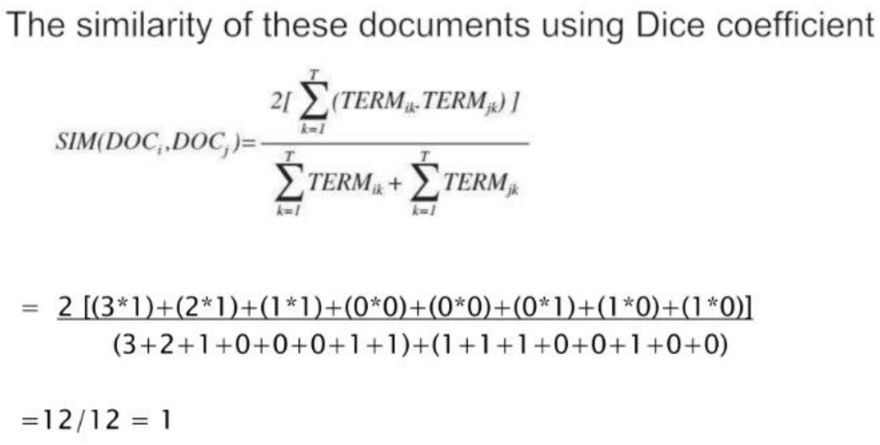
$$SIM(DOC_i,DOC_j)=\frac{\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})}{\sum_{k=1}^{T} TERM_{ik}+\sum_{k=1}^{T} TERM_{jk}-\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})}$$



EXAMPLES

Let the weights for the index terms assigned to two documents i and j be as follows:

Doc_i = 3,2,1,0,0,0,1,1
Doc_j = 1,1,1,0,0,1,0,0

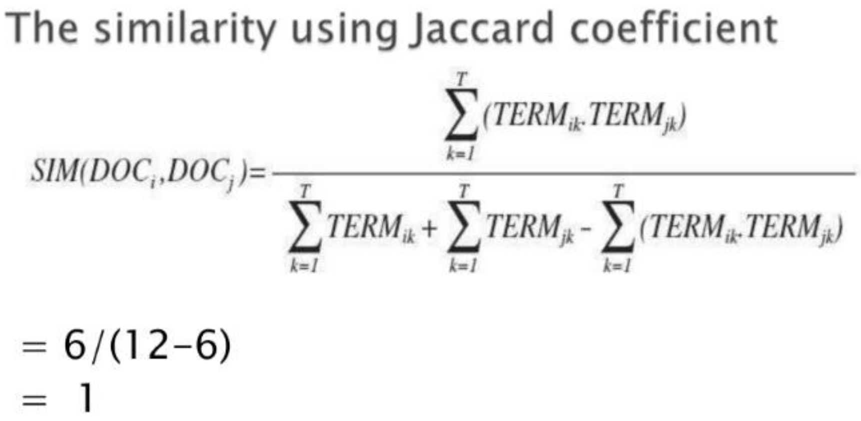


The similarity of these documents using Dice coefficient

$$SIM(DOC_i,DOC_j)=\frac{2[\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})]}{\sum_{k=1}^{T}TERM_{ik}+\sum_{k=1}^{T}TERM_{jk}}$$

$$=\frac{2\,[(3*1)+(2*1)+(1*1)+(0*0)+(0*0)+(0*1)+(1*0)+(1*0)]}{(3+2+1+0+0+0+1+1)+(1+1+1+0+0+1+0+0)}$$

$=12/12 = 1$



The similarity using Jaccard coefficient

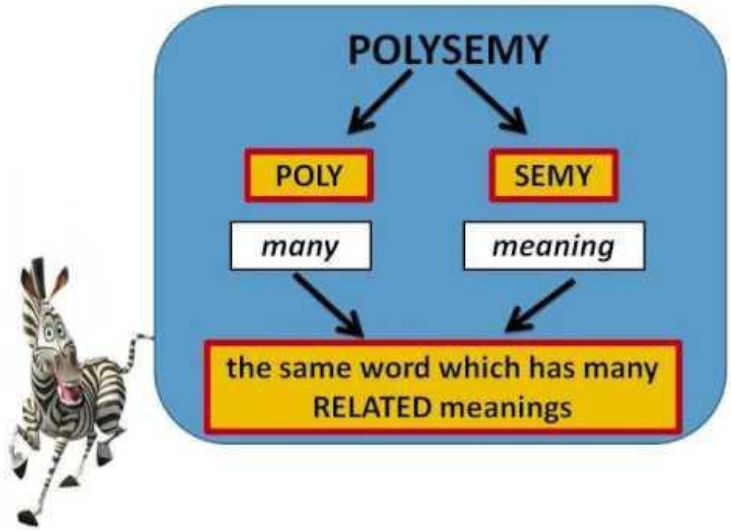
$$SIM(DOC_i,DOC_j)=\frac{\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})}{\sum_{k=1}^{T}TERM_{ik}+\sum_{k=1}^{T}TERM_{jk}-\sum_{k=1}^{T}(TERM_{ik}.TERM_{jk})}$$

$= 6/(12-6)$

$= \ 1$

homonymy, polysemy, synonymy

Polysemy is the association of one word with two or more distinct meanings, and a *polyseme* is a word or phrase with multiple meanings.



POLYSEMY

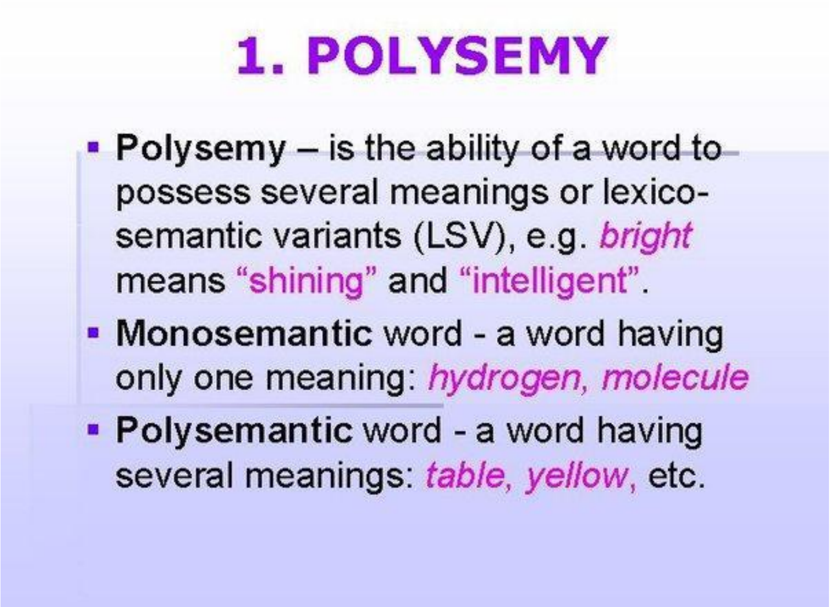
POLY

SEMY

many

meaning

the same word which has many RELATED meanings

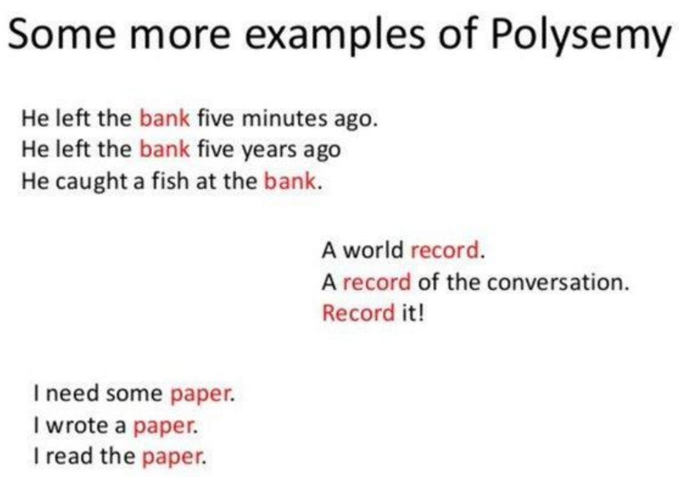


1. POLYSEMY

- **Polysemy** – is the ability of a word to possess several meanings or lexico-semantic variants (LSV), e.g. *bright* means "shining" and "intelligent".
- **Monosemantic** word - a word having only one meaning: *hydrogen, molecule*
- **Polysemantic** word - a word having several meanings: *table, yellow*, etc.

Here are a few examples of ***polysemous*** words. They are shown first in a *primary-meaning* context and followed by a *secondary-meaning* context.

- *Arms* bend at the elbow.
  Germany sells *arms* to Saudi Arabia.
- Boil the solution *once* with salt and *once* with sugar.
  Once Germany had surrendered, the Soviets were free to enter the conflict against Japan.



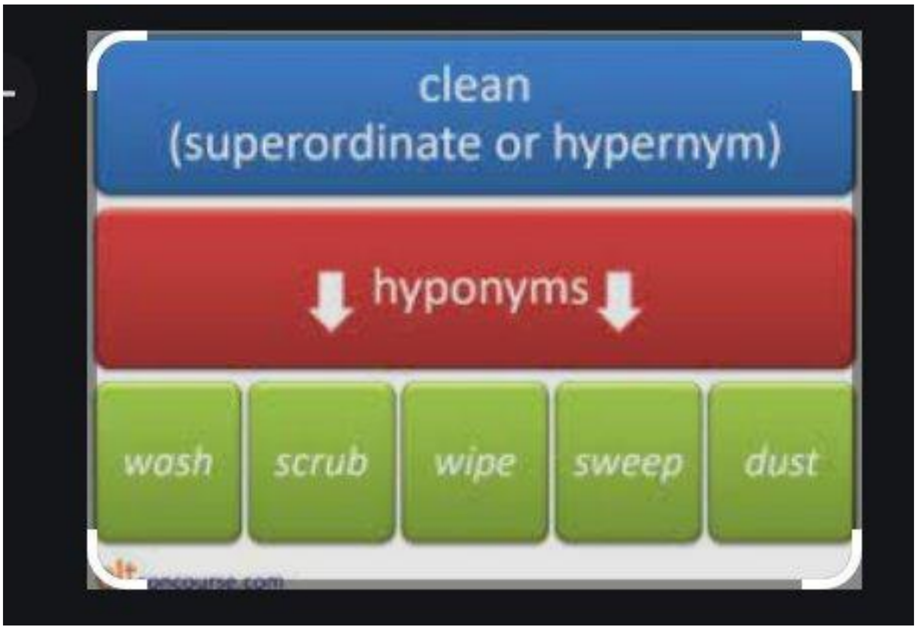
Some more examples of Polysemy

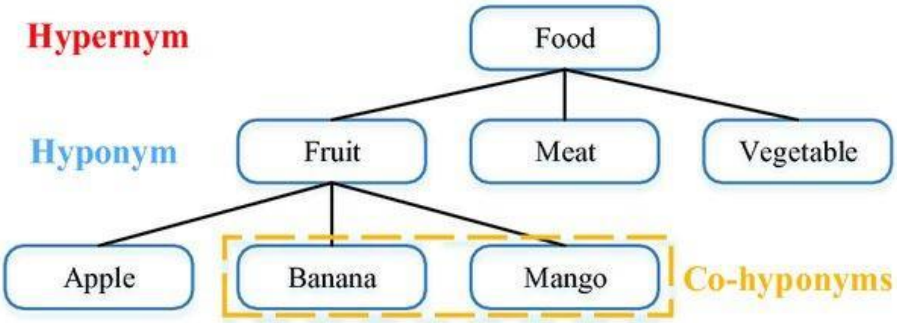
He left the bank five minutes ago.
He left the bank five years ago
He caught a fish at the bank.

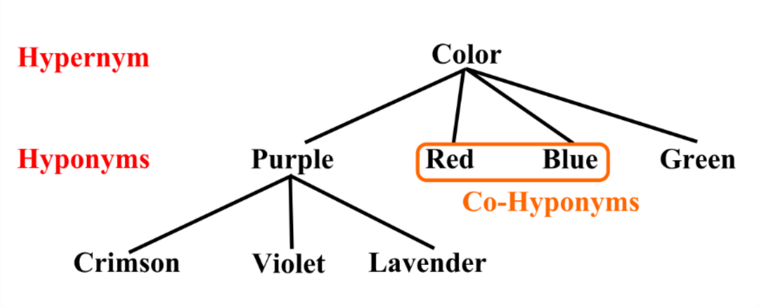
A world record.
A record of the conversation.
Record it!

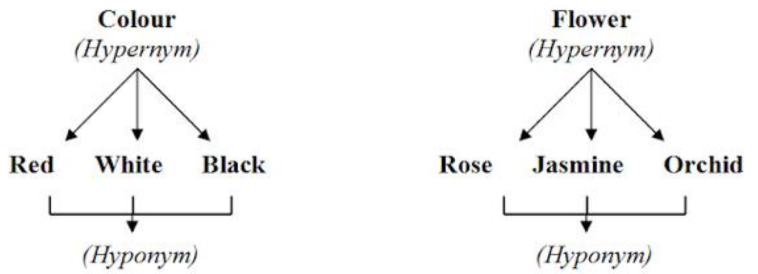
I need some paper.
I wrote a paper.
I read the paper.

In linguistics and lexicography, ***hyponym*** is a term used to designate a particular member of a broader class. For instance, *daisy* and *rose* are hyponyms of *flower*. Also called a *subtype* or a *subordinate term*.



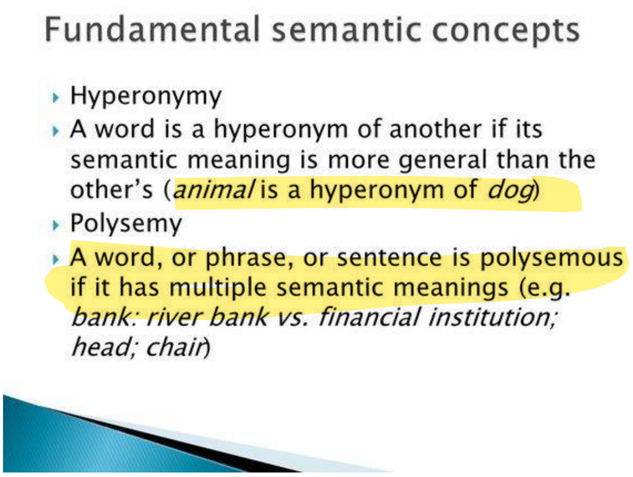






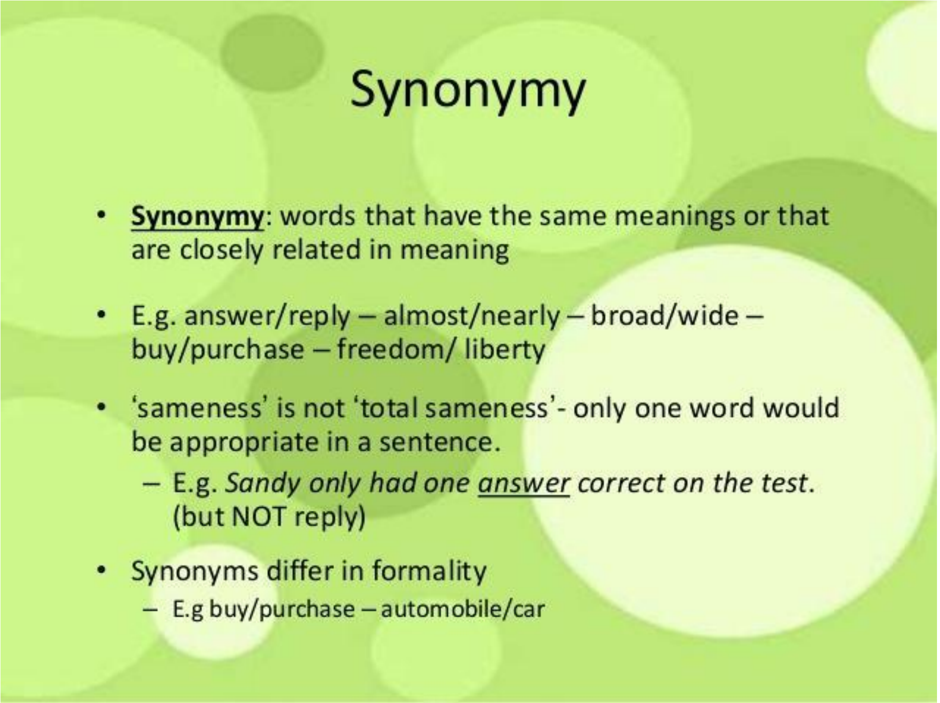
Homonymy – There are many pairs or groups of words, which, though different in meaning, are pronounced alike, or spelt alike, or both. Such words are called homonyms, and such words are in homonymy.

Lexical items which have the same sound or spelling or both, but differ in meaning, are called homonyms. Most of the homonyms are monosyllabic.



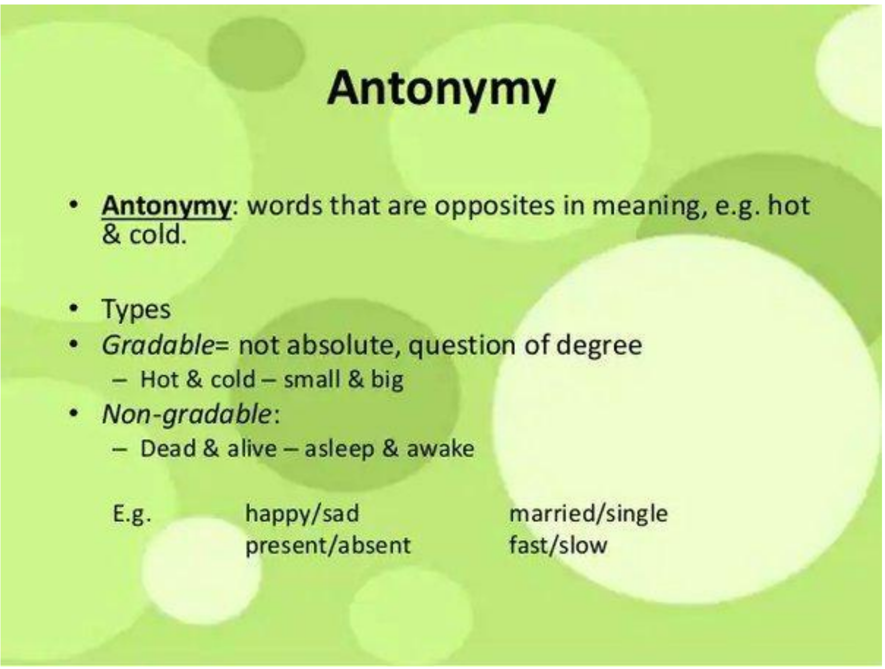
Fundamental semantic concepts

- Hyperonymy
- A word is a hyperonym of another if its semantic meaning is more general than the other's (*animal* is a hyperonym of *dog*)
- Polysemy
- A word, or phrase, or sentence is polysemous if it has multiple semantic meanings (e.g. *bank: river bank vs. financial institution; head; chair*)



Synonymy

- **<u>Synonymy</u>**: words that have the same meanings or that are closely related in meaning
- E.g. answer/reply – almost/nearly – broad/wide – buy/purchase – freedom/ liberty
- 'sameness' is not 'total sameness'- only one word would be appropriate in a sentence.
  - E.g. *Sandy only had one answer correct on the test.* (but NOT reply)
- Synonyms differ in formality
  - E.g buy/purchase – automobile/car



Antonymy

- **<u>Antonymy</u>**: words that are opposites in meaning, e.g. hot & cold.
- Types
- *Gradable*= not absolute, question of degree
  - Hot & cold – small & big
- *Non-gradable*:
  - Dead & alive – asleep & awake

E.g. happy/sad married/single
present/absent fast/slow

User Query Improvement

The primary goal of any information retrieval system must be accuracy – to produce relevant documents as per the user's requirement. However, the question that arises here is how can we improve the output by improving user's query formation style. Certainly, the output of any IR system is dependent on the user's query and a well-formatted query will produce more accurate results. The user can improve his/her query with the help of ***relevance feedback***, an important aspect of any IR model.

Relevance Feedback

Relevance feedback takes the output that is initially returned from the given query. This initial output can be used to gather user information and to know whether that output is relevant to perform a new query or not. The feedbacks can be classified as follows −

Explicit Feedback

It may be defined as the feedback that is obtained from the assessors of relevance. These assessors will also indicate the relevance of a document retrieved from the query. In order to improve query retrieval performance, the relevance feedback information needs to be interpolated with the original query.

Assessors or other users of the system may indicate the relevance explicitly by using the following relevance systems −

- **Binary relevance system** − This relevance feedback system indicates that a document is either relevant (1) or irrelevant (0) for a given query.
- **Graded relevance system** − The graded relevance feedback system indicates the relevance of a document, for a given query, on the basis of grading by using numbers, letters or descriptions. The description can be like "not relevant", "somewhat relevant", "very relevant" or "relevant".

Implicit Feedback

It is the feedback that is inferred from user behavior. The behavior includes the duration of time user spent viewing a document, which document is selected for viewing and which is not, page browsing and scrolling actions, etc. One of the best examples of implicit feedback is ***dwell time***, which is a measure of how much time a user spends viewing the page linked to in a search result.

Pseudo Feedback

It is also called Blind feedback. It provides a method for automatic local analysis. The manual part of relevance feedback is automated with the help of Pseudo relevance feedback so that the user gets improved retrieval performance without an extended interaction. The main advantage of this feedback system is that it does not require assessors like in explicit relevance feedback system.

Consider the following steps to implement this feedback −

- **Step 1** − First, the result returned by initial query must be taken as relevant result. The range of relevant result must be in top 10-50 results.

- **Step 2** – Now, select the top 20-30 terms from the documents using for instance term frequency(tf)-inverse document frequency(idf) weight.
- **Step 3** – Add these terms to the query and match the returned documents. Then return the most relevant documents.