



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Lexical Semantics

Lê Thanh Hương

School of Information and Communication Technology

Email: [huonglt@soict.hust.edu.vn](mailto:huonglt@soict.hust.edu.vn)

# Lexical Semantics

- Semantics is the study of the meaning of linguistic utterances.
- Lexical semantics is the study of:
  - (**lexical relations**) systematic meaning-related connections among lexemes, and
  - (**selectional restrictions**) the internal meaning-related structure of individual lexemes

# Content

1. Lexical relations
2. Selectional restrictions
3. Word sense disambiguation
4. Compute related words
5. Word representation
6. Application: Information Retrieval

# Homonymy (đồng âm)

**Homonyms** : words that share a form but have unrelated, distinct meanings:

- bank<sub>1</sub>: financial institution, bank<sub>2</sub>: sloping land
- bat<sub>1</sub>: club for hitting a ball, bat<sub>2</sub>: nocturnal flying mammal

## 1. Homographs (đồng âm, đồng tự):

- dove - dive into water, white bird
- saw

## 2. Homophones (đồng âm, không đồng tự):

write/right, piece/peace, see/sea; meat/meet

# Homonymy causes problems for NLP applications

- Information retrieval
  - “bat care”
- Machine Translation
  - bat: **murciélagos** (animal) or **bate** (for baseball)
- Text-to-Speech
  - bass (stringed instrument) vs. bass (fish)

# Polysemy (đa nghĩa)

1. The **bank** was constructed in 1875 out of local red brick.
  2. I withdrew the money from the **bank**
- Are those the same sense?
    - Sense 2: “A financial institution”
    - Sense 1: “The building belonging to a financial institution”
  - A **polysemous (đa hình)** word has **related** meanings
    - Most non-rare words have multiple meanings

# Metonymy (ẩn dụ) or Systematic Polysemy: A systematic relationship between senses

- Lots of types of polysemy are systematic
  - School, university, hospital
  - All can mean the institution or the building.
- A systematic relationship:
  - Building ↔ Organization
- Other such kinds of systematic polysemy:

Author (Jane Austen wrote Emma)

↔ Works of Author (I love Jane Austen)

Tree (Plums have beautiful blossoms)

↔ Fruit (I ate a preserved plum)

# Synonyms (đồng nghĩa)

- Word that have the same meaning in some or all contexts.
  - filbert / hazelnut
  - couch / sofa
  - big / large
  - automobile / car
  - vomit / throw up
  - Water / H<sub>2</sub>O
- Two lexemes are synonyms
  - if they can be substituted for each other in all situations
  - If so they have the same **propositional meaning**



# Polysemy, Synonymy

- Polysemy: one word – many senses, representing different aspects of an object, or different objects. E.g.,
  - *đi* : walk
  - *đi*: *die*
- Synonymy: several words – one meaning. E.g.,
  - cố, gắng
  - car, automobile

# Synonymy is a relation between senses rather than words

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

# Antonyms (trái nghĩa)

- Senses that are opposites with respect to one feature of meaning

- Otherwise, they are very similar!

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 **reversives**:
    - rise/fall, up/down

# Hyponymy and Hypernymy

- One sense is a **hyponym** (từ lớp con) of another if the first sense is more specific, denoting a subclass of the other
  - *car* is a hyponym of *vehicle*
  - *mango* is a hyponym of *fruit*
- Conversely **hypernym/superordinate** (“hyper is super”) (từ lớp cha)
  - *vehicle* is a **hypernym** of *car*
  - *fruit* is a hypernym of *mango*

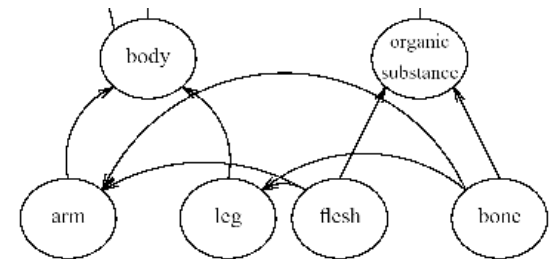
<b>Superordinate/hyper</b>	vehicle	fruit	furniture
<b>Subordinate/hyponym</b>	car	mango	chair

# Hyponyms and Instances

- WordNet has both **classes** and **instances**.
- An **instance** is an individual, a proper noun that is a unique entity
  - San Francisco is an **instance** of city
  - But city is a class
    - city is a **hyponym** of municipality...location...

# WordNet: Introduction

- A lexical database
  - Inspired by psycholinguistic theories of human lexical memory
  - Establishes a massive network of lexical items and lexical relationships
  - English wordnet
    - Four categories: noun, verb, adjective, adverb
    - Nouns: 120,000; Verbs: 22,000; Adjectives: 30,000; Adverbs: 6,000
- Wordnet in other languages [[www.globalwordnet.org](http://www.globalwordnet.org)]
  - Wordnets exist for: Basque, Spanish, Czech, Dutch, Estonian, French, German, Italian, Portuguese, Spanish, Swedish
  - Wordnets are in preparation for: Bulgarian, Danish, Greek, Hebrew, Hindi, Kannada, Latvian, Moldavian, Romanian, Russian, Slovenian, Swedish, Tamil, Thai, Turkish, Icelandic, Norwegian, Persian, Kurdish



# Senses of “bass” in Wordnet

## Noun

- **S: (n) bass** (the lowest part of the musical range)
- **S: (n) bass, bass part** (the lowest part in polyphonic music)
- **S: (n) bass, basso** (an adult male singer with the lowest voice)
- **S: (n) sea bass, bass** (the lean flesh of a saltwater fish of the family Serranidae)
- **S: (n) freshwater bass, bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- **S: (n) bass, bass voice, basso** (the lowest adult male singing voice)
- **S: (n) bass** (the member with the lowest range of a family of musical instruments)
- **S: (n) bass** (nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

## Adjective

- **S: (adj) bass, deep** (having or denoting a low vocal or instrumental range) *"a deep voice"; "a bass voice is lower than a baritone voice"; "a bass clarinet"*

# How is “sense” defined in WordNet?

- The **synset (synonym set)**, the set of near-synonyms, instantiates a sense or concept, with a **gloss**
- Example: **chump** as a noun with the **gloss**:  
“a person who is gullible and easy to take advantage of”
- This sense of “chump” is shared by 9 words:  
chump<sup>1</sup>, fool<sup>2</sup>, gull<sup>1</sup>, mark<sup>9</sup>, patsy<sup>1</sup>, fall guy<sup>1</sup>, sucker<sup>1</sup>,  
soft touch<sup>1</sup>, mug<sup>2</sup>
- Each of **these** senses have this same gloss
  - (Not **every** sense; sense 2 of gull is the aquatic bird)



# WordNet Hierarchies: example

WordNet: example from ver1.7.1

Sense 3: Vancouver

⇒ (city, metropolis, urban center)

⇒ (municipality)

⇒ (urban area)

⇒ (geographical area)

⇒ (region)

⇒ (location)

⇒ (entity, physical thing)

⇒ (administrative district, territorial division)

⇒ (district, territory)

⇒ (region)

⇒ (location)

⇒ (entity, physical thing)

⇒ (port)

⇒ (geographic point)

⇒ (point)

⇒ (location)

⇒ (entity, physical thing)

# WordNet Noun Relations

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

# WordNet 3.0

- Link:
  - <http://wordnetweb.princeton.edu/perl/webwn>
- Libraries
  - Python: WordNet from NLTK
    - <http://www.nltk.org/Home>
  - Java:
    - JWNL, extJWNL on sourceforge

# Selectional Restrictions: Encoding meaning in the grammar

- Predicates impose constraints on their arguments
  - *read (human subject, textual object)*
  - *eat (animate subject)*
  - *kill (animate object)*
- *Use the predicate to disambiguate its arguments*
- *Example "dish":*
  - *plate for eating off*
  - *course of a meal*
  - *communications device*

# Dish Example

- Not unexpectedly, wives, whether working or non-working, did by far the most - about 80% of the shopping, laundry and cooking, and about two-thirds of housecleaning, washing *dishes*, *child care*, and *family paper work*.
- *In her tiny kitchen at home, Ms. Chen works efficiently, stir-frying several simple *dishes*, including braised pig's ears and chicken livers with green peppers.*
- *Installation of satellite *dishes*, TVs and videocassette equipment will cost the company about \$20,000 per school, Mr Whittle said.*

# Selectional Restrictions

- Phrase structure grammars can implement selectional restrictions
  - create an ontology (e.g. human, animate, ...)
  - constrain the PS rules
    - e.g.  $VP \rightarrow V_{kill} NP_{animate}$
  - constrain the semantic interpretation
    - e.g. eat([being], [food])
- Problem?

# Exploring Lexical Relations

Identify lexical relationships between words in the following sentences.

I love domestic animals. I especially like cats because they are very independent pets. Dogs, on the other hand, tend to be quite needy. For example, you've got to walk them everyday.

# Exploring Lexical Relations

- Thesaurus:
  - Synonyms and Antonyms
- Wordnet:
  - Synonyms and Antonyms
  - Hypernyms and Hyponyms
  - ...



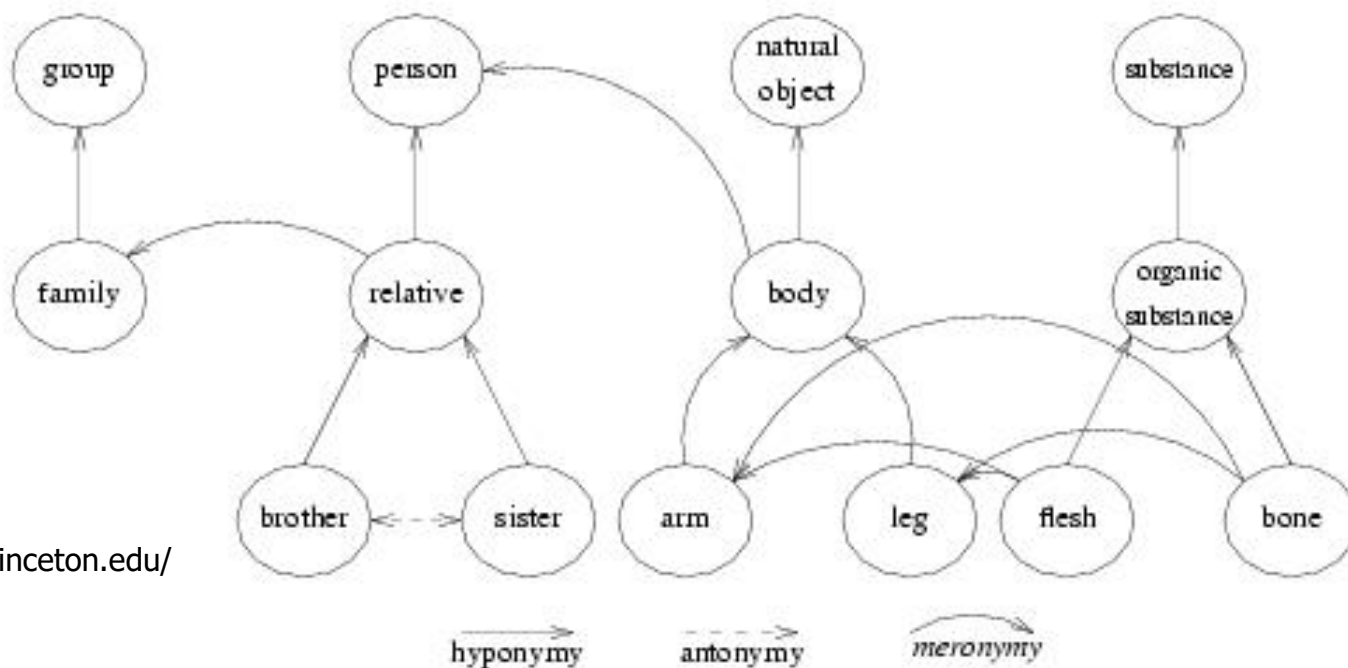
# Disambiguation via Selectional Restrictions

- Disambiguation:
  - Different Predicate select for different thematic roles
    - wash the dishes (theme: washable-thing)
  - An argument can also disambiguate its predicate
    - serve vegetarian dishes (theme: food-type)
- Semantic parsing:
  - Semantic attachment rules are applied as sentences are syntactically parsed – phrase structure grammars
    - “I wanna eat somewhere close to CSSE”
    - Transitive:  $V \rightarrow \text{eat} \langle \text{theme} \rangle \{ \text{theme: food-type} \}$  (VP  $\rightarrow$  V NP)
    - Intransitive:  $V \rightarrow \text{eat} \langle \text{no-theme} \rangle$  (VP  $\rightarrow$  V)
  - Selectional restriction violation: eliminate parse

- Problem: creates brittle grammars
  - Sometimes selectional restrictions don't restrict enough (Which dishes do you like?)
  - Sometimes they restrict too much (I'll eat my hat!)
    - when predicates are used metaphorically

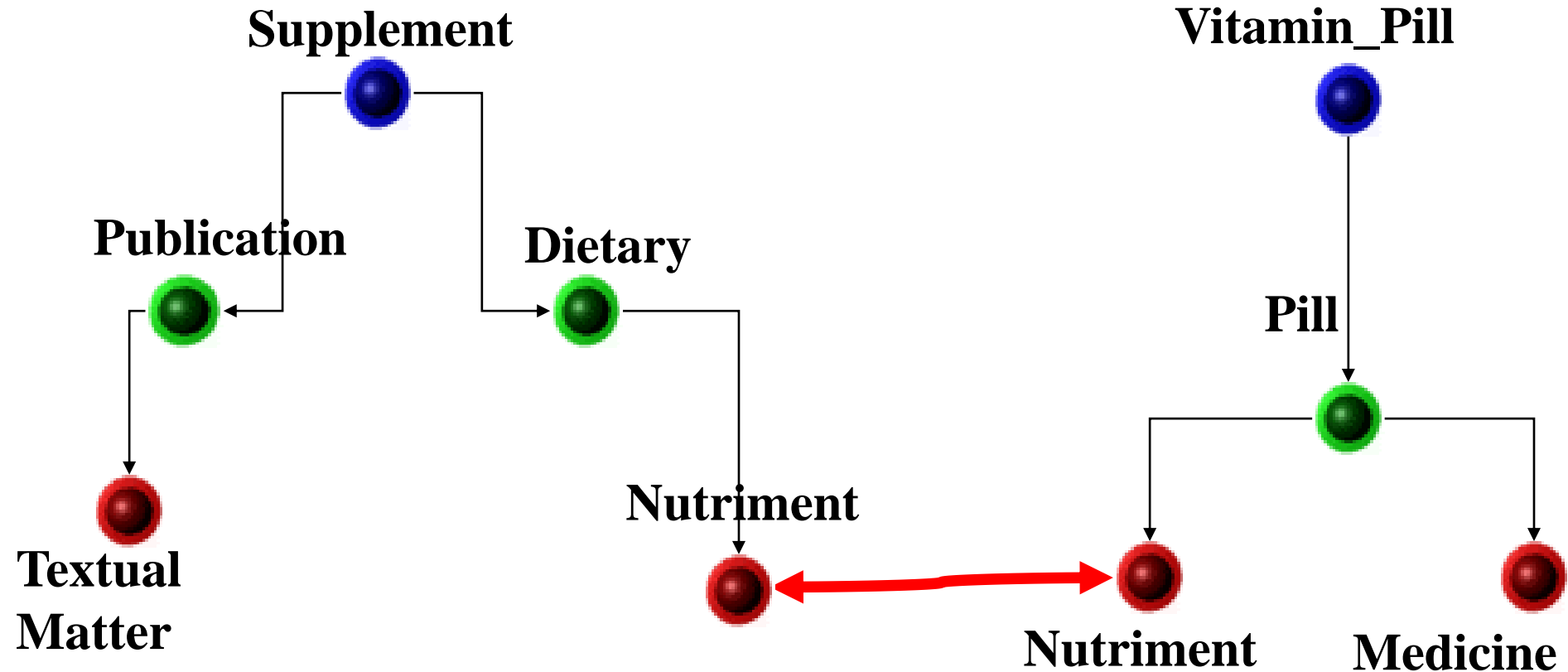
# WordNet Relations





- Words are connected in a vertical fashion via hypernymy (specialisation) and holonymy (generalisation) relationships, and horizontal via meronymy (part\_of) and holonymy (has\_part) relationships.
- Each unique sense of a word is represented by a synset number



<http://wordnet.princeton.edu/>

# Disambiguation via Lexical Relations



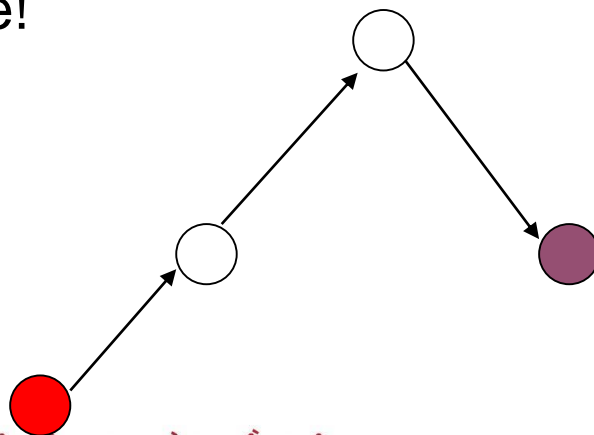
-  SENSE OF WORD
-  KIND-OF (HYPONIMY)
-  HAS-PART (HOLONYMY)
-  PART-OF (MERONYMY)

WordNet Similarity Metrics:

<http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi>

# Measuring Semantic Relatedness

- Not enough to say that two words are related – also need a measure of the strength of this relationship.
- Edge/Node counting in a taxonomy is one method:
  - the distance between two words (nodes) in the WordNet taxonomy is inversely proportion to the strength of the semantic relationship between them.
  - If there are multiple paths between two words take the shortest one!



No Edges = 3

No Nodes = 4

# Which word pair is more similar?

- whale and fish?
- fish and trout?

WordNet Similarity Metrics:

<http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi>



# WordNet::Similarity

Read an overview of [WordNet::Similarity](#).

You may enter any two words in one of three formats:

1. word
2. word#part\_of\_speech (where part\_of\_speech is one of n, v, a, or r)
3. word#part\_of\_speech#sense (where sense is a positive integer)

If words are entered in format 1 or 2, then the relatedness of all valid forms of the words will be computed (e.g., if 'dogs' is entered, then 'dog' will be used to compute relatedness). [More instructions](#).

Word 1:  ☒ Use all senses ☐ Pick a sense by [gloss](#) ☐ Pick a sense by [synset](#)  
Word 2:  ☒ Use all senses ☐ Pick a sense by [gloss](#) ☐ Pick a sense by [synset](#)  
Measure:  [About the measures](#)  
☒ Use [root node](#)?

[Show version info](#)

Created by Ted Pedersen and Jason Michelizzi  
E-mail: tpederse (at) d (dot) umn (dot) edu



# WordNet::Similarity

Read an overview of [WordNet::Similarity](#).

You may enter any two words in one of three formats:

- 1. word
- 2. word#part\_of\_speech (where part\_of\_speech is one of n, v, a, or r)
- 3. word#part\_of\_speech#sense (where sense is a positive integer)

If words are entered in format 1 or 2, then the relatedness of all valid forms of the words will be computed (e.g., if 'dogs' is entered, then 'dog' will be used to relatedness). [More instructions](#).

Word 1:

Word 2:

Measure:

☒ Use all senses    ☐ Pick a sense by [gloss](#)    ☐ Pick a sense by [synset](#)

☒ Use all senses    ☐ Pick a sense by [gloss](#)    ☐ Pick a sense by [synset](#)

[About the measures](#)

☒ Use root nodes

[Show version info](#)

Created by Ted Pedersen  
E-mail: [tpederse \(at\)](mailto:tpederse@at)

Use All Measures

Path Length

Leacock & Chodorow

Wu & Palmer

Resnik

Jiang & Conrath

Lin

Adapted Lesk (Extended Gloss Overlaps)

Gloss Vectors

Gloss Vectors (pairwise)

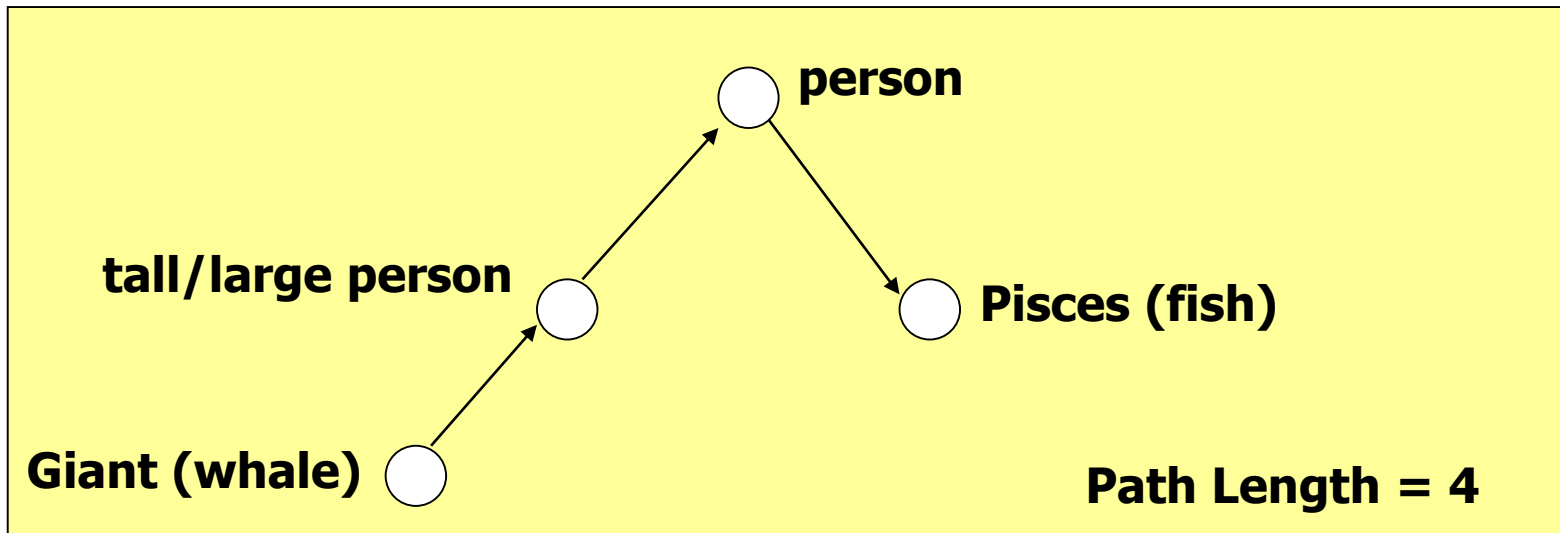
Hirst & St-Onge

Random Measure

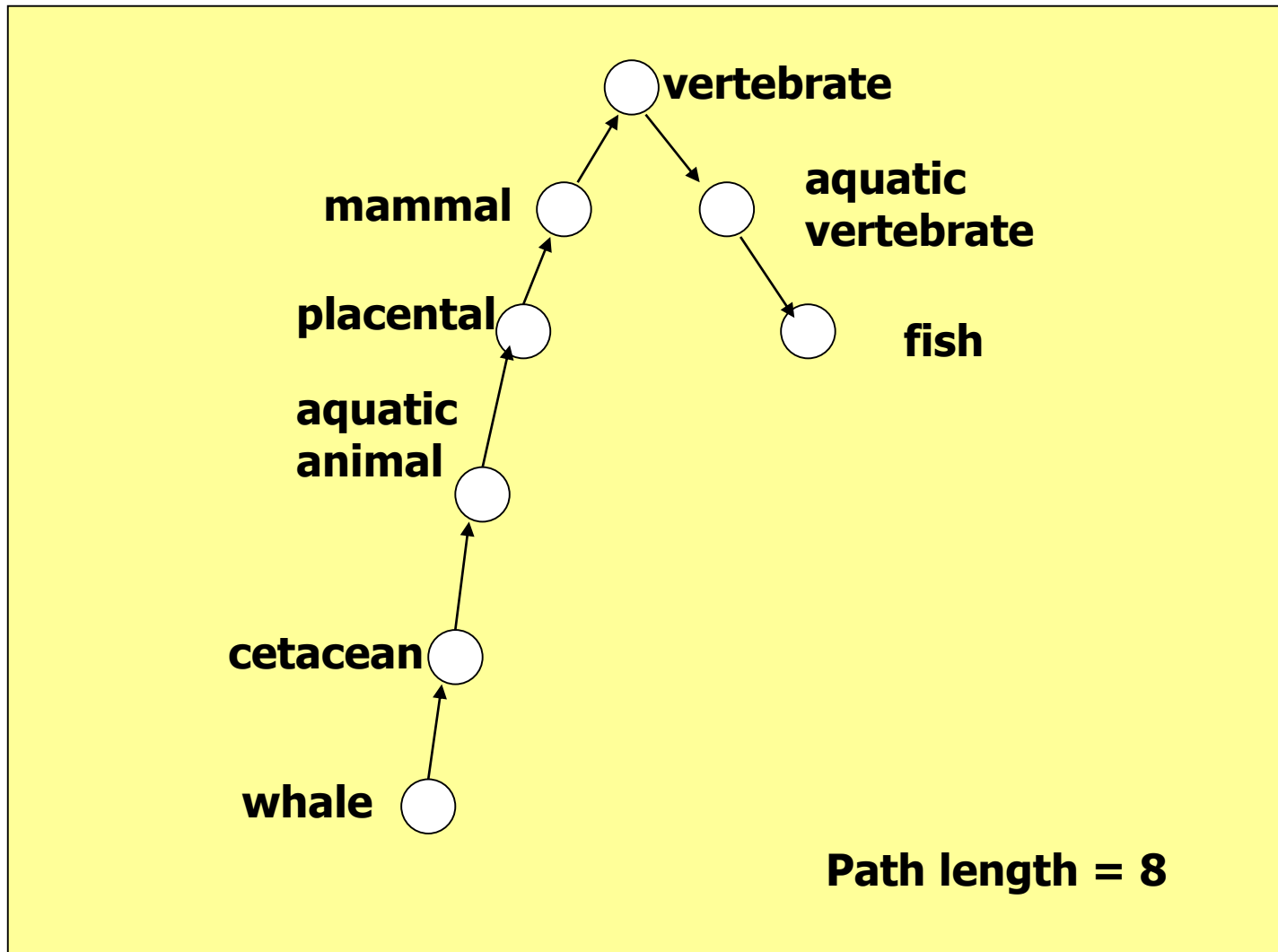


# Sense Disambiguation and Edge Counting

- whale#n#1
  - a very large person; impressive in size or qualities
- fish#n#3
  - (astrology) a person who is born while the sun is in Pisces



# Sense Disambiguation and Edge Counting



# Shortcomings of WordNet for measuring semantic relatedness

- WordNet-based (edge counting) semantic relatedness measures are based around the following incorrect assumptions:
  - Every edge in the taxonomy is of equal length
  - All branches in the taxonomy are equally dense
  - All transitive relationships are valid
- Hence, semantic distance in WordNet isn't always a reliable estimate of the semantic relatedness between 2 words.

# Dictionary Approaches

- Machine readable dictionaries (Lesk '86)
  - Retrieve all definitions of content words occurring in context of target (e.g. I've often caught bass while out at sea)
  - Compare for overlap with sense definitions of target entry (bass<sup>2</sup>: a type of fish that lives in the sea)
  - Choose sense with most overlap
- Limits: Entries are short --> expand entries to 'related' words

# WSD Machine Learning Approaches

- Learn a classifier to assign one of possible word senses for each word
  - Acquire knowledge from labeled or unlabeled corpus
  - Human intervention *only* in labeling corpus and selecting set of features to use in training
- Input: feature vectors
  - Target (word to be disambiguated)
  - Context (features that can be used to predict the correct sense tag)
- Output: classification rules for unseen text

# Input Features for WSD

- POS tags of target and neighbors
- Surrounding context words (stemmed or not)
- Punctuation, capitalization and formatting
- Partial parsing to identify thematic/grammatical roles and relations
- Collocational information:
  - How likely are target and left/right neighbor to co-occur
- Co-occurrence of neighboring words
  - Intuition: How often does **sea** occur with **bass**

# Example

- Tôi ăn cơm với cá.
  - DT ĐgT DT GT DT
  - (C (CN (ĐaT Tôi)) (VN (ĐgN (ĐgN (ĐgT ăn) (DT cơm)) (GN (GT với) (DT cá))))))
- Em bé chỉ thích ăn kẹo thôi.
  - DT TT TT ĐgT DT PT
  - (C (CN (DT Em bé)) (VN (TN (TN (TT chỉ) (TN (TT thích) (ĐgN (ĐgT ăn) (DT kẹo)))) (PT thôi))))))
- Nó ăn nhiều hoa hồng quá.
  - ĐaT ĐgT TT DT TT
  - (C (CN (ĐaT Nó)) (VN (ĐgN (ĐgN (ĐgT ăn) (TT nhiều) (DT hoa hồng)) (TT quá))))))
- Tôi tên là Hoa.

# Types of Classifiers

- Naïve Bayes: Best sense is most probable sense given input vector

- $\hat{s} = \arg \max_{s \in S} p(s|V)$ , or  $\arg \max_{s \in S} \frac{p(V|s)p(s)}{p(V)}$

- Where  $s$  is one of the senses possible and  $V$  the input vector of features
  - Limited data available associating specific vectors to senses =>
  - Assume features independent, so probability of  $V$  given  $s$  is the product of probabilities of each feature

$$p(V|s) = \prod_{j=1}^n p(v_j|s)$$

- and  $p(V)$  same for any  $\hat{s}$  (doesn't affect final ranking)

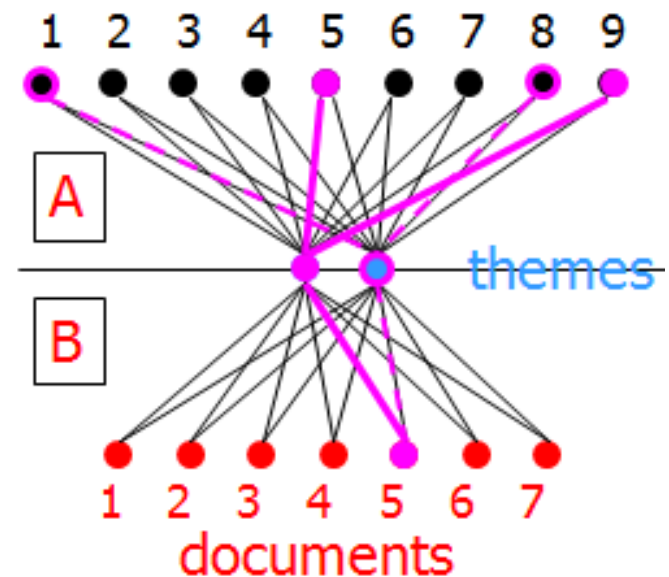
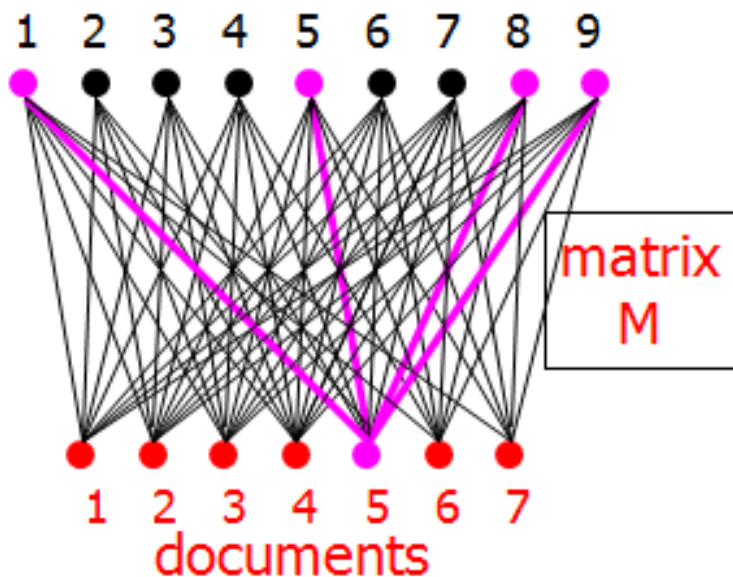


# Types of Classifiers

- **Naïve Bayes:** Best sense is most probable sense given input vector
  - Then 
$$\hat{s} = \arg \max_{s \in S} p(s) \prod_{j=1}^n p(v_j | s)$$
  - $P(s)$  is prior for each sense = proportion of each sense in tagged data
  - $P(v,s)$  = count of occurrence of bass with sea

# ML approach to find related words

- Latent semantic analysis method:
  - SVD (Singular Value Decomposition)



# ML approach to find related words

- Latent semantic analysis method:
  - LSA (Latent Semantic Analysis)

$$\begin{array}{ccccccc}
 & X & & U & & \Sigma & & V^T \\
 & (\mathbf{d}_j) & & & & & & (\hat{\mathbf{d}}_j) \\
 & \downarrow & & & & & & \downarrow \\
 (\mathbf{t}_i^T) \rightarrow & \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} & = & (\hat{\mathbf{t}}_i^T) \rightarrow & \begin{bmatrix} \begin{bmatrix} \mathbf{u}_1 \end{bmatrix} & \dots & \begin{bmatrix} \mathbf{u}_l \end{bmatrix} \end{bmatrix} & \cdot & \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_l \end{bmatrix} & \cdot & \begin{bmatrix} \begin{bmatrix} \mathbf{v}_1 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \mathbf{v}_l \end{bmatrix} \end{bmatrix}
 \end{array}$$

# ML approach to find related words

- LDA (Latent Dirichlet Allocation)

$\alpha$  is the parameter of the Dirichlet prior on the per-document topic distributions,

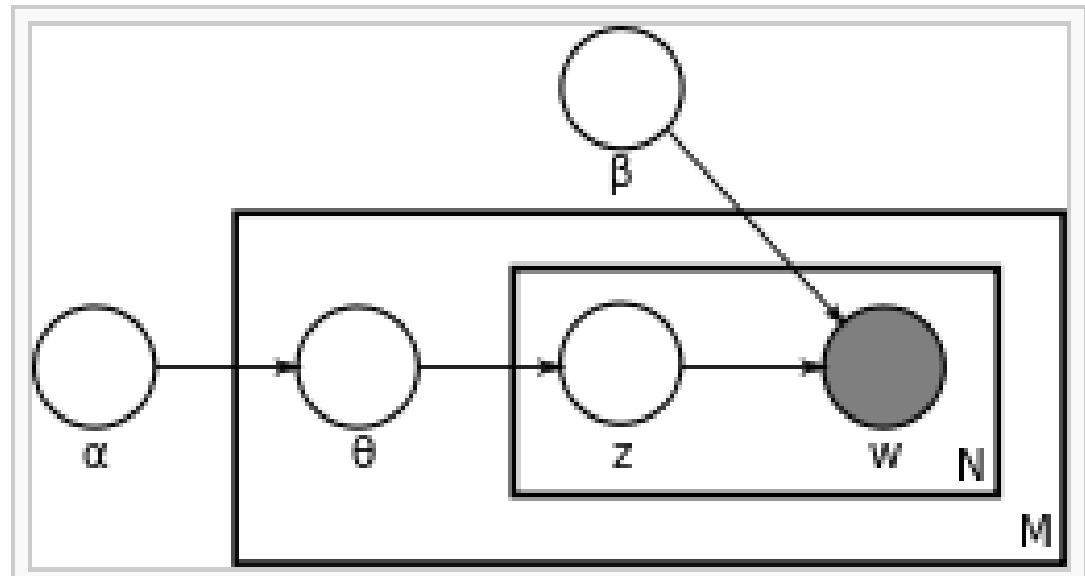
$\beta$  is the parameter of the Dirichlet prior on the per-topic word distribution,

$\theta_i$  is the topic distribution for document  $i$ ,

$\varphi_k$  is the word distribution for topic  $k$ ,

$z_{ij}$  is the topic for the  $j$ th word in document  $i$ , and

$w_{ij}$  is the specific word.



# Example - Output of LDA

Topic0:	Topic1:	Topic2:	Topic3:	Topic4:	Topic5:	Topic6:	Topic7:	Topic8:	Topic9:
cstag	cstag	cstag	cstag	cstag	cstag	cstag	cstag	cstag	cstag
credit	agenttag	agenttag	agenttag	agenttag	agenttag	agenttag	agenttag	agenttag	agenttag
agenttag	credit	credit	credit	credit	interest	credit	interest	credit	credit
interest	interest	interest	interest	interest	credit	interest	credit	interest	interest
pay	pay	time	pay	payment	pay	pay	pay	pay	time
time	low	payment	time	pay	time	payment	payment	time	low
payment	rate	pay	low	time	low	time	time	low	payment
low	time	rate	payment	low	payment	low	rate	payment	rate
rate	payment	low	rate	rate	rate	use	low	use	pay
use	charg	use	use	use	use	rate	use	rate	use

# Machine Learning Approach to find related words

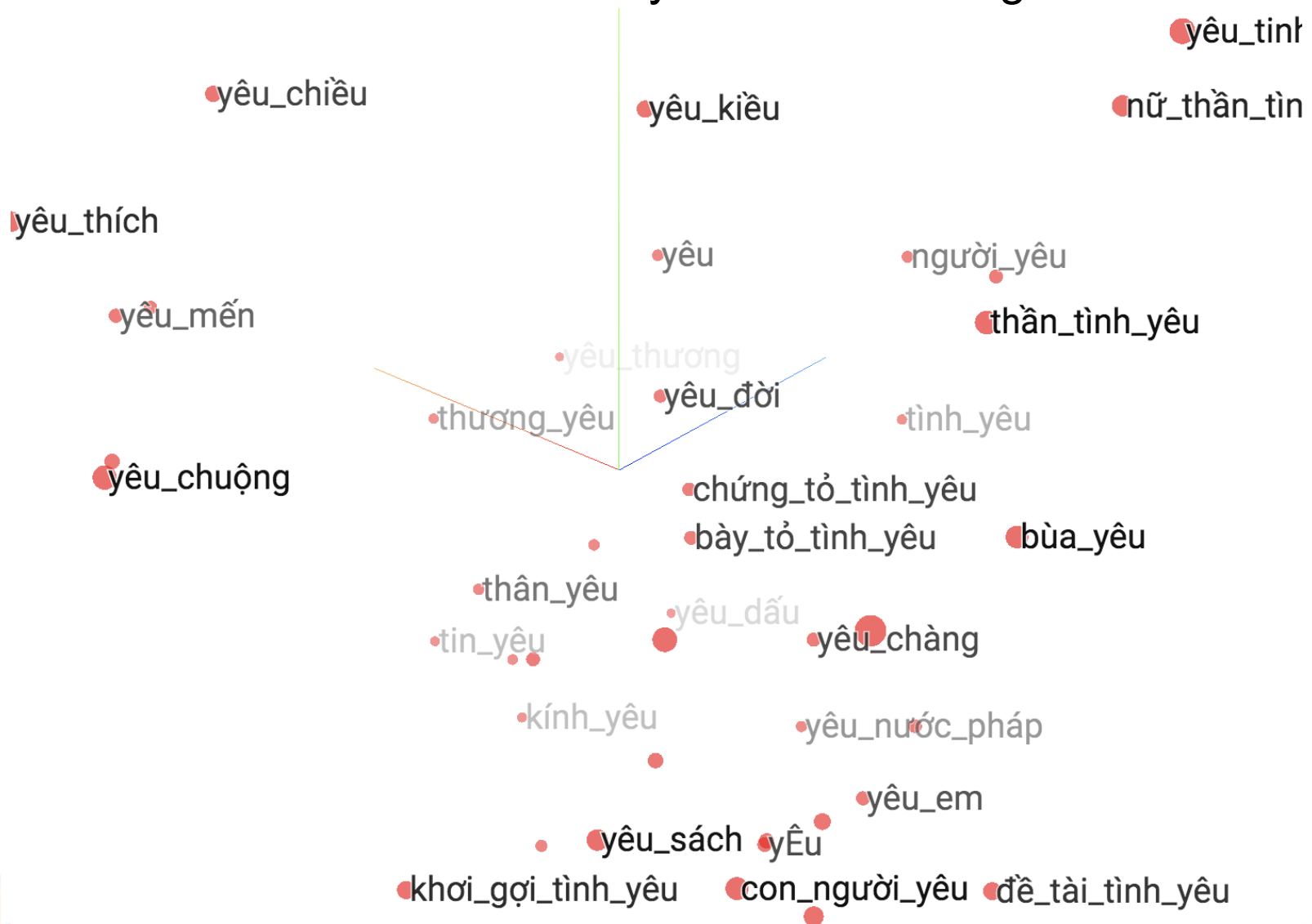
- Word embeddings are a type of word representation that allows words with similar meaning to have a similar representation.
- Previous word representation methods:
  - One-hot encoding
  - Co-occurrence matrix

# One-hot encoding

- Dataset:
  - Tôi đang đi tìm một\_nửa của mình
  - Tôi đã ăn một\_nửa quả táo
  - Tôi đã đi tìm một\_nửa quả táo
- Vocabulary  $V = \{ \text{tôi\_1, đang\_2, đi\_3, tìm\_4, một-nửa\_5, của\_6, mình\_7, đã\_8, ăn\_9, quả\_10, táo\_11} \}$
- Word representation
  - Tôi = [1 0 0 0 0 0 0 0 0 0 0]
  - đang = [0 1 0 0 0 0 0 0 0 0 0]
  - ...
  - mình = [0 0 0 0 0 0 1 0 0 0 0]
  - táo = [0 0 0 0 0 0 0 0 0 0 1]
- Disadvantages: consumes resources, cannot represent semantic relationships between words

# Co-occurrence matrix

- “You will understand a word by the words that go with it”





# Co-occurrence matrix

Tôi đang đi tìm một\_nửa của mình  
Tôi đã ăn một\_nửa quả táo  
Tôi đã đi tìm một\_nửa quả táo

- Text level provides general information on topics towards LSA methods
- The "word window" level provides information about both the syntactic and semantic function of the word

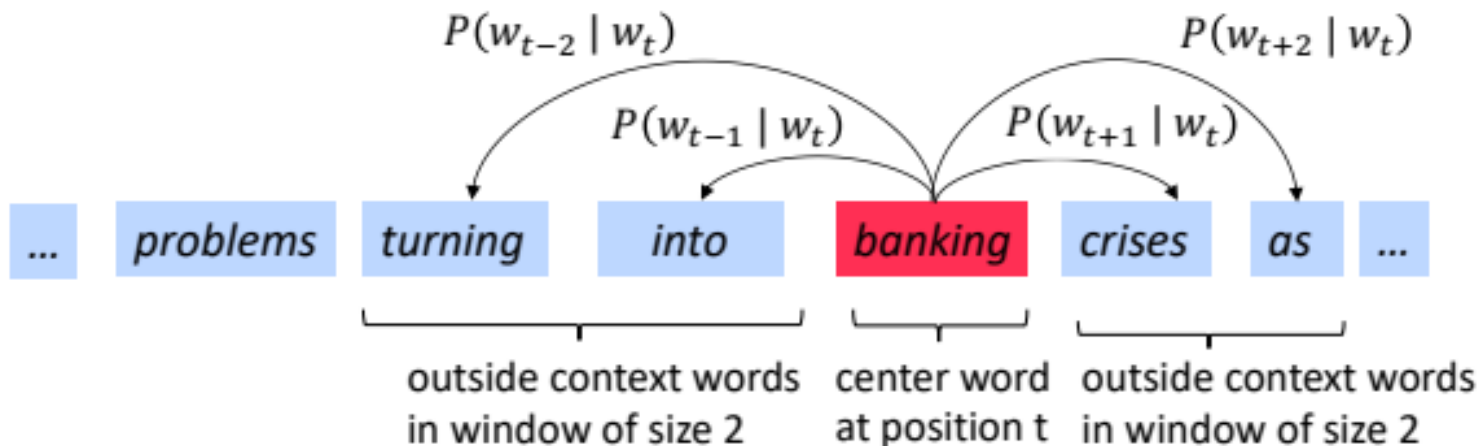
Counts	tôi	đang	đi	tìm	một_nửa	của	mình	đã	ăn	quả	táo
tôi	0	1	0	0	0	0	0	2	0	0	0
đang	1	0	1	0	0	0	0	0	0	0	0
đi	0	1	0	2	0	0	0	1	0	0	0
tìm	0	0	2	0	2	0	0	0	0	0	0
một_nửa	0	0	0	2	0	1	0	0	1	2	0
của	0	0	0	0	1	0	1	0	0	0	0
mình	0	0	0	0	0	1	0	0	0	0	0
đã	2	0	1	0	0	0	0	0	1	0	0
ăn	0	0	0	0	1	0	0	1	0	0	0
quả	0	0	0	0	2	0	0	0	0	0	2
táo	0	0	0	0	0	0	0	0	0	2	0

# Co-occurrence matrix

- Acquiring information about co-occurrence of words in the learning data
- Problem :
  - The vector 's dimension increases with the dictionary' size.
  - Need large memory space to store information.
  - Subsequent classification models based on this representation will encounter sparsity issues.
- Solution: Singular Value Decomposition

# Word embedding

- Instead of storing the occurrence information of words by directly counting as co-occurrence matrix, word2vec learns to guess the neighbors of all words.
- Method:
  - Guess the neighboring words in the window size  $m$  of each word:
  - For each word  $t = 1 \dots T$ ,
    - Guess the words within the window size  $m$  of all the words.



# Target function

For each position  $t = 1, \dots, T$ , predict context words within a window of fixed size  $m$ , given center word  $w_j$ .

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^T \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(w_{t+j} | w_t; \theta)$$

$\theta$  is all variables  
to be optimized

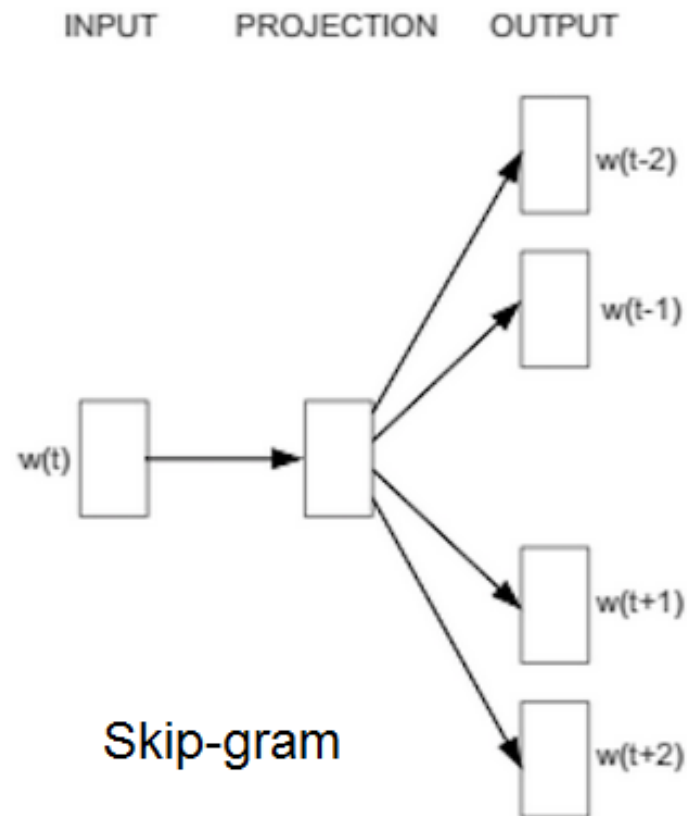
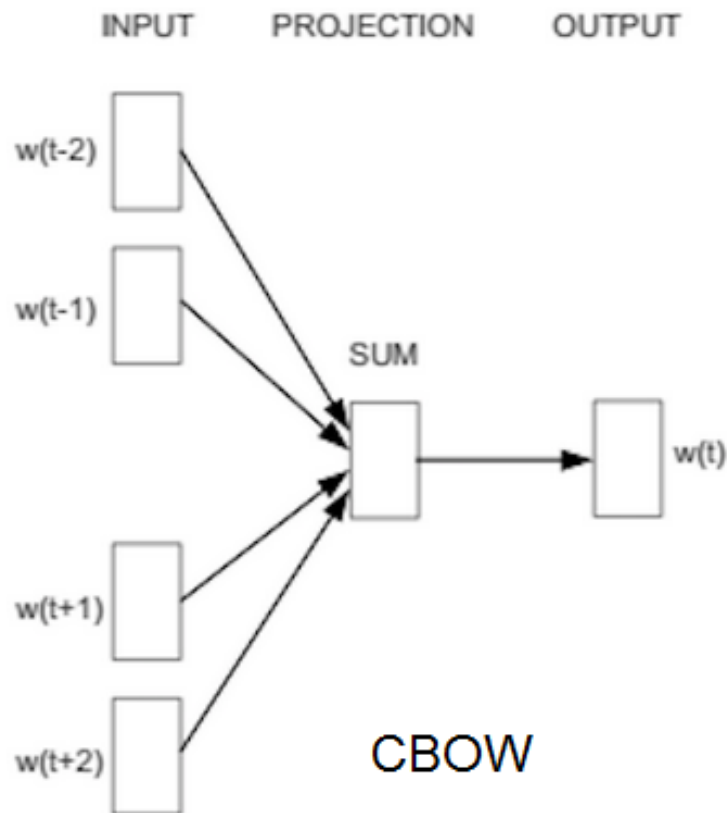
sometimes called *cost* or *loss* function

The *objective function*  $J(\theta)$  is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^T \sum_{\substack{-m \leq j \leq m \\ j \neq 0}} \log P(w_{t+j} | w_t; \theta)$$

Loss/cost function

# Word embedding



## CBoW:

- Given context words
- Guess probability of a target word

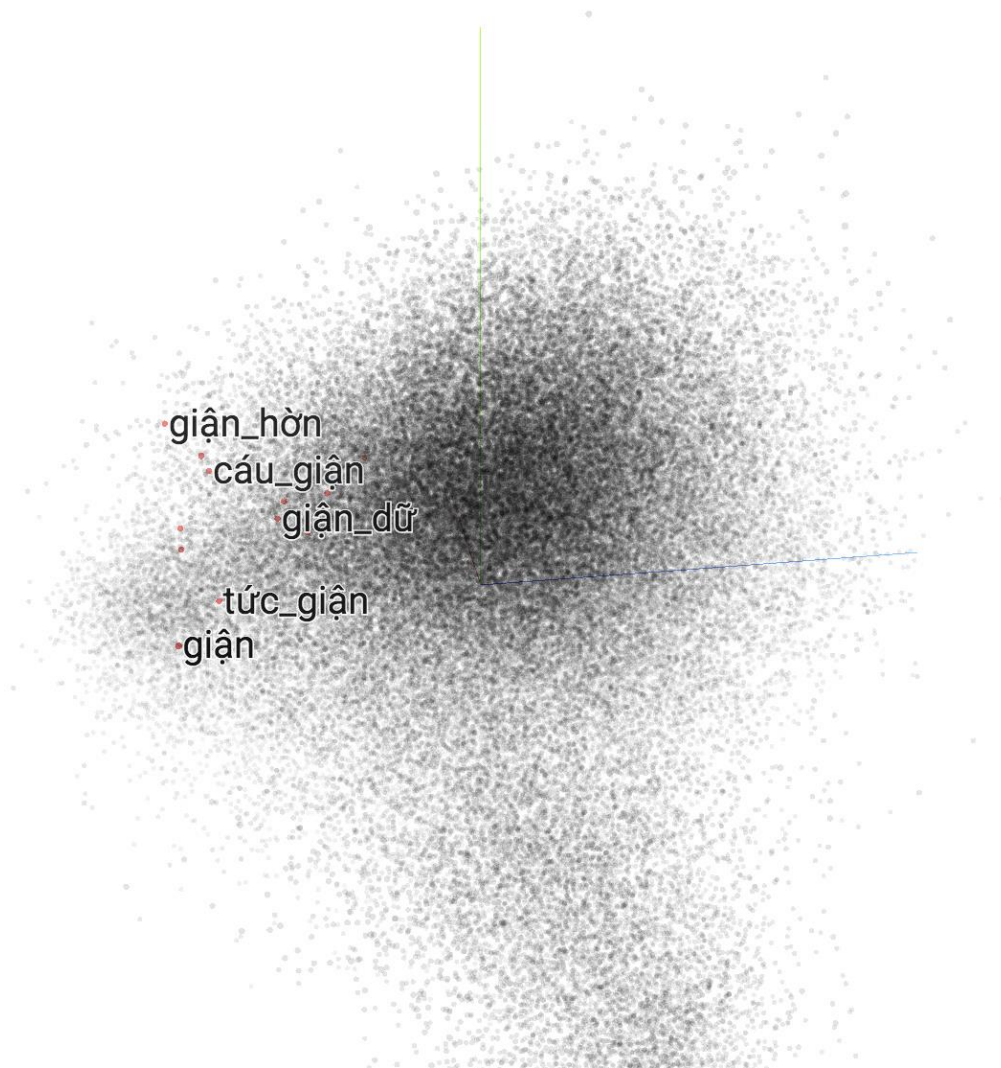
## Skip-gram:

- Given the target word
- Guess probability of context words

# Word embedding

- Different versions of word embedding:
  - Gensim: input is words. Better than Fasttext in terms of semantics
  - Fasttext: interested in word structure → separates words into syllables. Better than gensim in terms of syntax
- Disadvantages:
  - Word representation vectors are context-insentitive → BERT

# word2vecVN



Search	by
giận	label
13 matches.	
tức_giận	
giận	
giận_dữ	
nổi_giận	
nóng_giận	
giận_dối	
giận_hờn	
cáu_giận	
chọc_giận	
hờn_giận	
hả_giận	
căm_giận	
oán_giận	

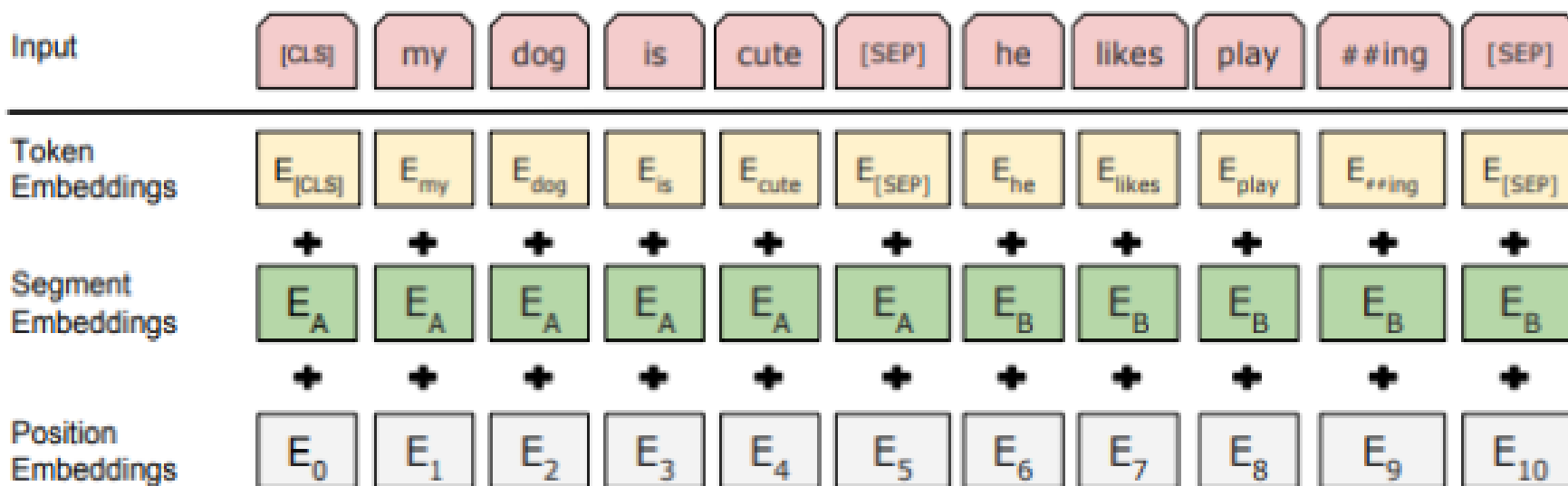
# Bidirectional Encoder Representations from Transformers (BERT)

- Bert is a language representation model from Google, using pre-training and fine-tuning to generate language models for many tasks: Question Answering, sentiment analysis,.....
- BERT was trained by bidirectional context of Transformer



# BERT

- Input: 1 sentence or 1 pair of sentences (e.g., [question, answer])

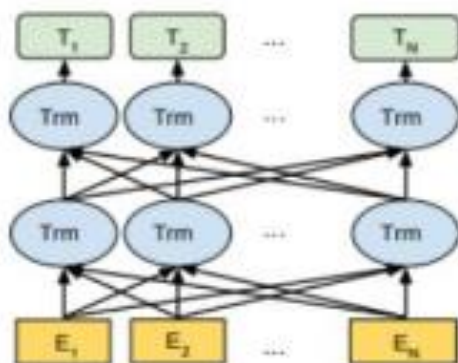


# BERT

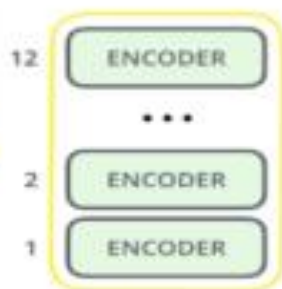
- **Position Embeddings:** token position in a sentence.
- **Token Embeddings:** tokens from the input text. 1<sup>st</sup> token is [CLS]. Last token is [SEP].
- **Segment Embeddings:** distinguish 2 sentences in case the input is a sentence pair. Sentence A is 0 values; sentence B is 1 values

# BERT architecture

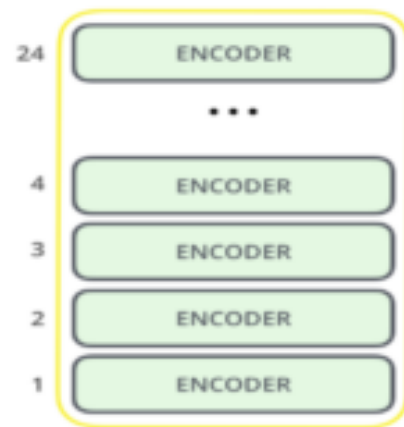
- BERT uses bidirectional Transformer encoder with several layers. Self-attention layer is performed following bidirectional way
- 2 variants available:
  - BERT Base: 12 layers (transformer blocks), 12 attention heads, 110M parameters
  - BERT Large: 24 layers (transformer blocks), 16 attention heads, 340M parameters



BERT Architecture



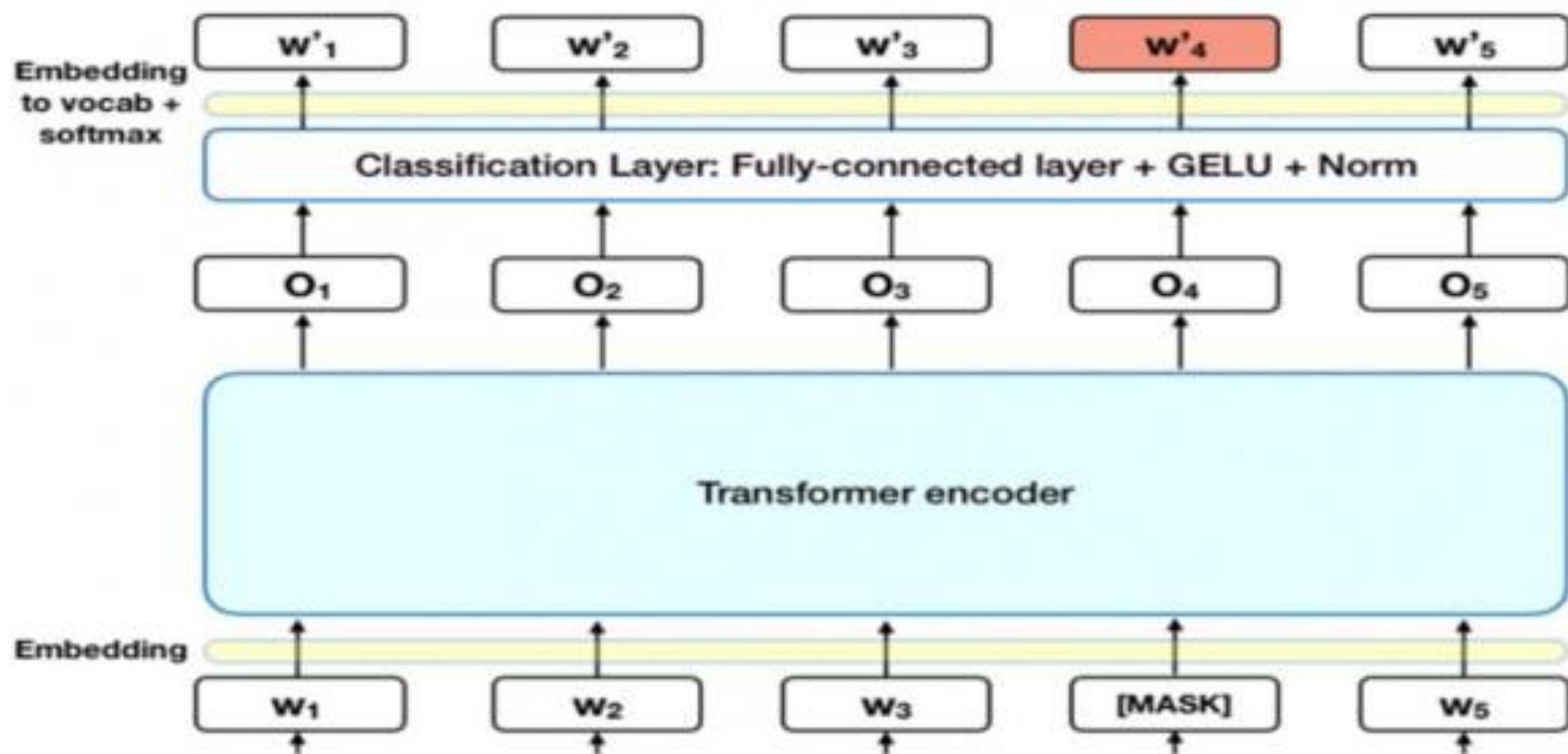
BERT<sub>BASE</sub>



BERT<sub>LARGE</sub>

# Training BERT

- BERT is pre-trained using 2 unsupervised prediction tasks
  - Masked Language Modeling (MLM)
  - Next Sentence Prediction (NSP)



# Training BERT

- **Next Sentence Prediction (NSP)**

- BERT uses sentence pairs as training data. E.g., using a dataset of 100.000 sentences for pre-training a language model => 50.000 samples (sentence pairs) are used for training
- With 50% sentence pairs, 2<sup>nd</sup> sentence is the next sentence for the 1<sup>st</sup> sentence. The label is “IsNext”
- With the rest 50%, 2<sup>nd</sup> sentence is a random sentence from the dataset. The label is “notNext”
- **Note:** When training BERT, MLM and NSP are trained together to reduce errors

# BERT

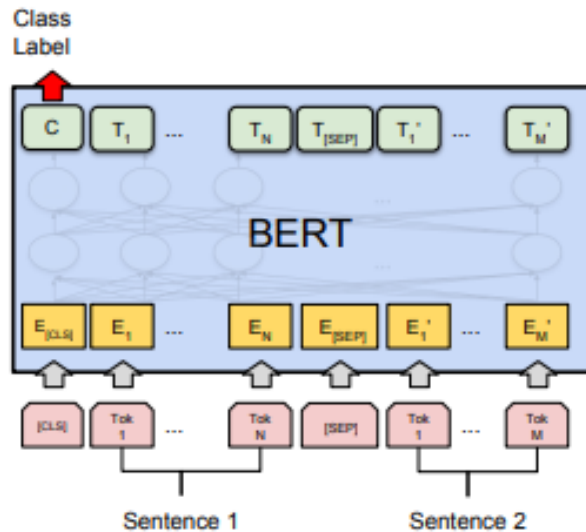
**Input:** [CLS] người đàn\_ông làm [MASK] tại cửa\_hàng [SEP] anh\_ta rất [MASK] và thân\_thiện [SEP]

**Label:** isNext

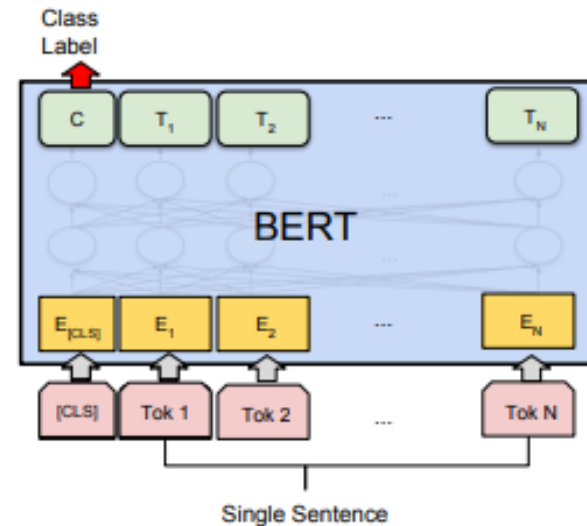
**Input:** [CLS] người đàn\_ông làm [MASK] tại cửa\_hàng [SEP] cô\_ta đang cầm súng [SEP]

**Label:** notNext

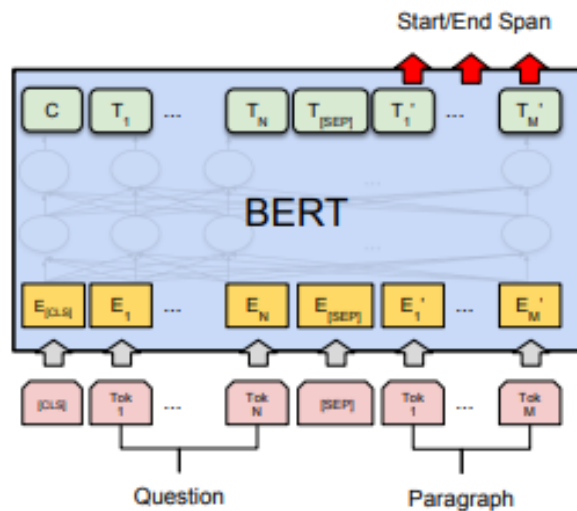
# Some applications of BERT



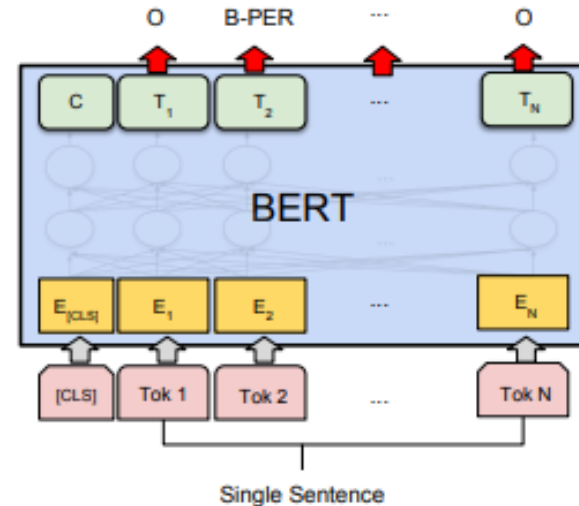
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



(b) Single Sentence Classification Tasks:  
SST-2, CoLA



(c) Question Answering Tasks:  
SQuAD v1.1



(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

# Applications of lexical semantics

- Text Summarization
- Text classification
- Sentiment Analysis
- Contextual advertising
- Text Matching
- Search Engines
- Dialogue system
- Question answering
- ...



# WSD and IR

- Motivation
  - Homonymy = Bank (financial, river)
  - Polysemy = Bat ((the club used in playing cricket), (a small racket with a long handle used for playing squash))
  - Synonymy = doctor, doc, physician, MD, medico (a licensed medical practitioner)
- How do these linguistic phenomena affect IR performance?
  - Homonymy and Polysemy: tend to decrease precision
  - Synonymy: tends to decrease recall

# Two IR applications of WSD

- Query-based retrieval experiments using WordNet (Voorhees, 1998):
  - Using WSD for query expansion: disambiguate the query and add in related hyponyms etc..
  - Using WSD for conceptual indexing: disambiguate document collection and build an index of synset numbers rather than an index of stemmed words
  - Vector Space IR model: find cosine similarity between query vector and each document vector
- Conceptual Indexing Experimental Results
  - Sense based vectors (in all experiments) performed more poorly than stemmed word based vectors
  - Causes??
  - Disambiguation errors
    - in document collection and
    - short queries due to lack of context

# Two IR applications of WSD

- Query Expansion Experimental Results
  - Again no performance gains
  - BUT!!
  - Manually query disambiguation and expansion did help
- Why?
  - *furniture*: table, chair, board, refectory (specialisations)
  - “Only certain lexicographically related words are useful in the expansion process, because a hypernym path between words in the WordNet taxonomy does not always indicate a useful query expansion term.”

# WSD Accuracy and IR

- Gold standard WSD evaluation datasets: SensEval and SemCor
- Alternative to hand annotated data: Pseudowords
  - Take two words (at random) with the same part of speech, and replace both by a single artificial word. For example, 'door' and 'banana' might both be replaced in a corpus by the word 'donana'.
  - The WSD accuracy: determine whether each instance of donana was originally 'door' or 'banana'. (Yarowsky, 1993)
- (Sanderson, 1997) showed that adding ambiguity to queries and collections has little effect on IR performance compared to the effect of adding disambiguation errors to the collection
  - only low levels of disambiguation error (less than 10%) would result in improvements over a basic word stem-based IR model.

# WSD Accuracy and IR

- Reasons why polysemy/homonymy isn't as big a problem as we suspected:
  - Query word collocation effect: query terms implicitly disambiguate each other
  - Skewed sense distributions: Applies to specific domains
  - Stemmed VSM is very effective: only 5% of stems are unrelated (Buitelaar, 1998)

# WSD Accuracy and IR

- Synonymy may have a bigger effect:
  - Gonzalo et al. (1998; 1999): using SemCor (Brown corpus documents with WordNet sense tags) showed that if disambiguation accuracy 100%
    - Synset indexing (e.g. synset number) = IR accuracy of 62%
    - Word sense indexing (e.g. canine<sub>1</sub>) = IR accuracy 53.2%
    - Stemmed word indexing = IR accuracy 48%
  - Gonzalo et al. also suggest 90% minimum WSD accuracy for IR is too high – nearer 60% - pseudowords don't always behave like real ambiguous words