CS626: Speech, Natural Language Processing and the Web

WSD Techniques

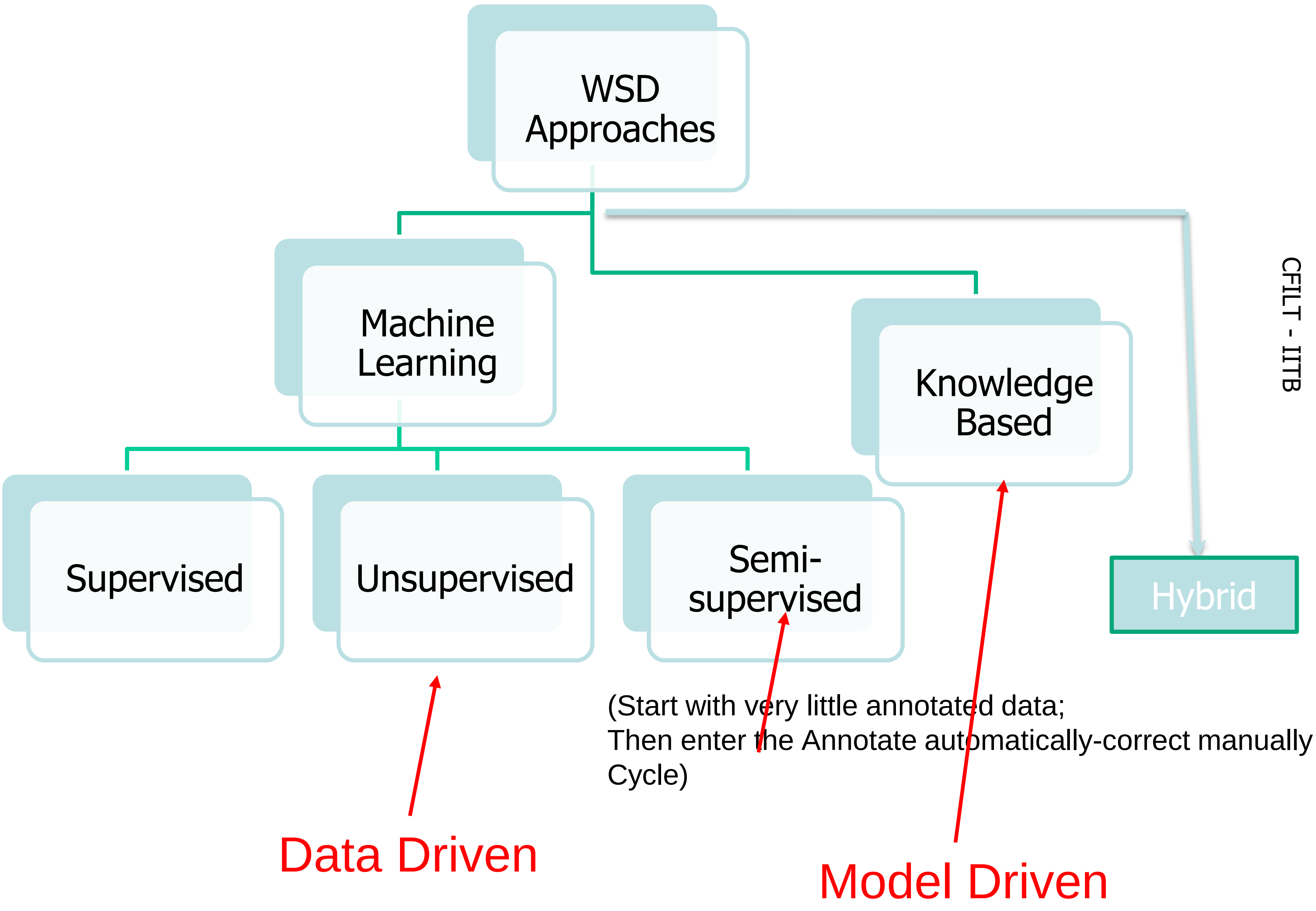
Pushpak Bhattacharyya

Computer Science and Engineering Department

IIT Bombay

Week 5 of 23rd August, 2021

Bird’s eye view of WSD techniques



WSD Approaches

Machine Learning

Knowledge Based

Hybrid

Supervised

Unsupervised

Semi-supervised

(Start with very little annotated data;
Then enter the Annotate automatically-correct manually
Cycle)

Data Driven

Model Driven

Wordnet - Lexical Matrix (with examples)

Word Meanings (IDs)	**Word**				
	$\mathbf{F_1}$	$\mathbf{F_2}$	$\mathbf{F_3}$	**…**	$\mathbf{F_n}$
$\mathbf{M_1}$	(*depend*) $E_{1,1}$	(*bank*) $E_{1,2}$	(rely) $E_{1,3}$		
$\mathbf{M_2}$		(*bank*) $E_{2,2}$		(*embankment*) $E_{2,\ldots}$	
$\mathbf{M_3}$		(*bank*) $E_{3,2}$	$E_{3,3}$		
…				…	
$\mathbf{M_m}$					$E_{m,n}$

Sense tagged corpora (task: sentiment analysis)

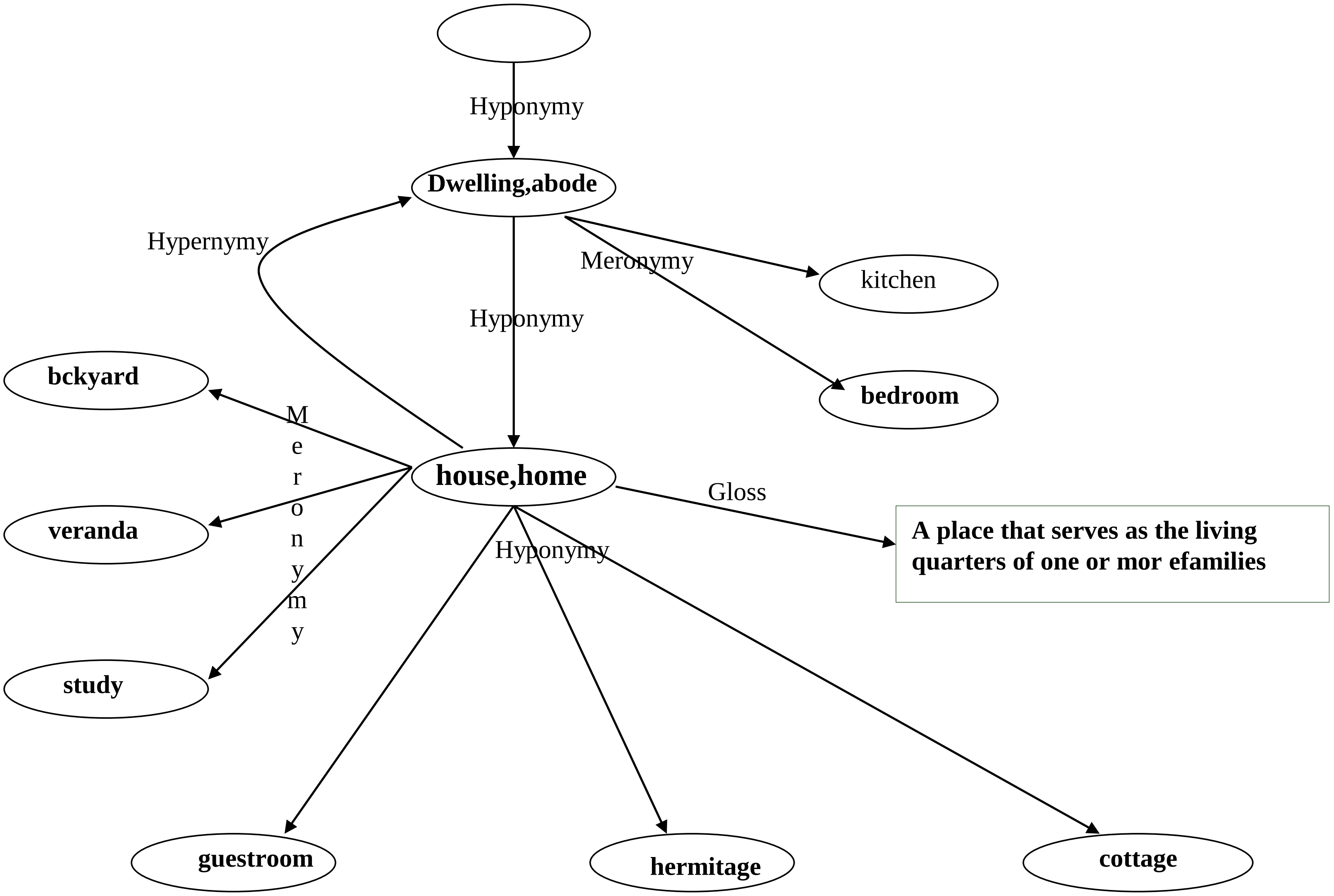
- I have enjoyed_21803158 #LA#_18933620 every_42346474 time_17209466 I have been_22629830 there_3110157 , regardless_3119663 if it was for work_1578942 or pleasure_11057430.
- I usually_3107782 fly_21922384 into #LA#_18933620, but this time_17209466 we decided_2689493 to drive_21912201 .
- Interesting_41394947, to say_2999158 the least_3112746 .

Senses of “pleasure”

The noun pleasure has 5 senses (first 2 from tagged texts)

1. (21) pleasure, pleasance -- (a fundamental feeling that is hard to define but that people desire to experience; "he was tingling with pleasure")
2. (4) joy, delight, pleasure -- (something or someone that provides pleasure; a source of happiness; "a joy to behold"; "the pleasure of his company"; "the new car is a delight")
3. pleasure -- (a formal expression; "he serves at the pleasure of the President")
4. pleasure -- (an activity that affords enjoyment; "he puts duty before pleasure")

WordNet Sub-Graph



A place that serves as the living quarters of one or mor efamilies

Vector representation of a synset

- Vector of a synset: < *Hypernymy id, Meronymy id, Hyponymy id, Representation for the gloss, Representation for example sentence, and so on >

- Hypernymy id – Id of the synset which is linked by hypernymy to the given node

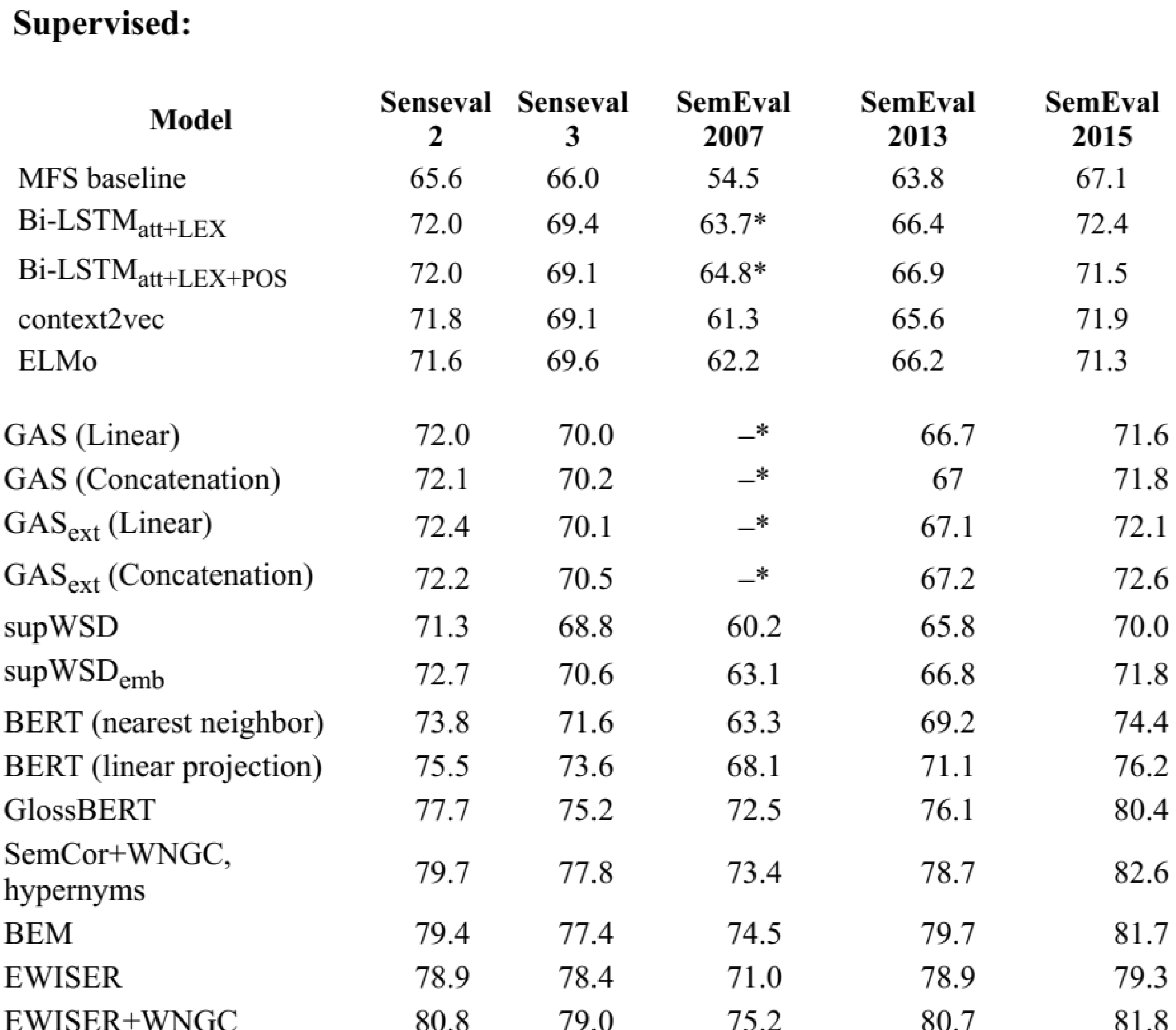
Definition of WSD

- The task of Word Sense Disambiguation (WSD) consists of associating words in context with their most suitable entry in a pre-defined sense inventory.
- The de-facto sense inventory for English in WSD is WordNet. For example, given the word “mouse” and the following sentence:

Training Data for WSD

- The most widely used training corpus used is SemCor, with 226,036 sense annotations from 352 documents manually annotated.
- Some supervised methods, particularly neural architectures, usually employ the SemEval 2007 dataset.
- The most usual baseline is the Most Frequent Sense (MFS) heuristic, which selects for each target word the most frequent sense in the training data.

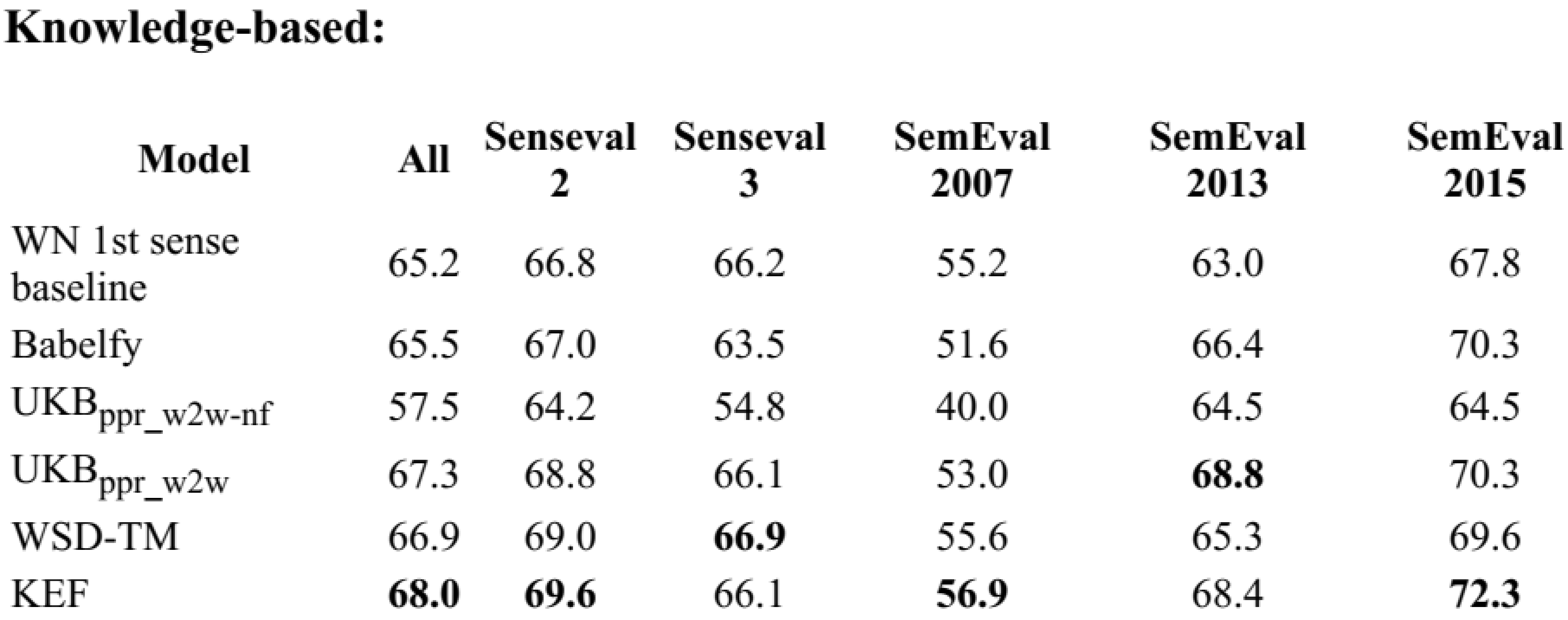
WSD: State of Art (1/2)



Supervised:

Model	Senseval 2	Senseval 3	SemEval 2007	SemEval 2013	SemEval 2015
MFS baseline	65.6	66.0	54.5	63.8	67.1
Bi-LSTM$_{att+LEX}$	72.0	69.4	63.7*	66.4	72.4
Bi-LSTM$_{att+LEX+POS}$	72.0	69.1	64.8*	66.9	71.5
context2vec	71.8	69.1	61.3	65.6	71.9
ELMo	71.6	69.6	62.2	66.2	71.3
GAS (Linear)	72.0	70.0	–*	66.7	71.6
GAS (Concatenation)	72.1	70.2	–*	67	71.8
GAS$_{ext}$ (Linear)	72.4	70.1	–*	67.1	72.1
GAS$_{ext}$ (Concatenation)	72.2	70.5	–*	67.2	72.6
supWSD	71.3	68.8	60.2	65.8	70.0
supWSD$_{emb}$	72.7	70.6	63.1	66.8	71.8
BERT (nearest neighbor)	73.8	71.6	63.3	69.2	74.4
BERT (linear projection)	75.5	73.6	68.1	71.1	76.2
GlossBERT	77.7	75.2	72.5	76.1	80.4
SemCor+WNGC, hypernyms	79.7	77.8	73.4	78.7	82.6
BEM	79.4	77.4	74.5	79.7	81.7
EWISER	78.9	78.4	71.0	78.9	79.3
EWISER+WNGC	80.8	79.0	75.2	80.7	81.8

WSD: SOTA (2/2)



Knowledge-based:

Model	All	Senseval 2	Senseval 3	SemEval 2007	SemEval 2013	SemEval 2015
WN 1st sense baseline	65.2	66.8	66.2	55.2	63.0	67.8
Babelfy	65.5	67.0	63.5	51.6	66.4	70.3
$\text{UKB}_{\text{ppr\_w2w-nf}}$	57.5	64.2	54.8	40.0	64.5	64.5
$\text{UKB}_{\text{ppr\_w2w}}$	67.3	68.8	66.1	53.0	**68.8**	70.3
WSD-TM	66.9	69.0	**66.9**	55.6	65.3	69.6
KEF	**68.0**	**69.6**	66.1	**56.9**	68.4	**72.3**

Modeling of WSD- sense *S* given word *W* and context *C*

$$S^* = \arg\max_{S} P(S \mid w, C) \qquad w \in C$$

$$P(S \mid w, C) = \frac{\#(w\_tagged\_as\_S\_in\_context\ C)}{\#(w\_in\_context\ C)}$$

Isolate "*prior*" probability

$$
\begin{aligned}
& P(S \mid w, C) \\
&= \frac{P(S, w, C)}{P(w, C)} \\
&= \frac{P(w)P(S, C \mid w)}{P(w)P(C \mid w)} \\
&= \frac{P(S, C \mid w)}{P(C \mid w)} \\
&= \frac{P(S \mid w)P(C \mid S, w)}{P(C \mid w)}
\end{aligned}
$$

Constant in *argmax*
calculation

$$S^* = \arg\max_S (P(S \mid w, C)) = \arg\max_S (P(S \mid w) P(C \mid S, w))$$

Prior

$$P(S \mid w) = \frac{\#(w\_tagged\_as\_S)}{\# w}$$

Likelihood

Let W^S = W in sense S

Apply chain rule and make Markov assumption

$$P(C \mid w^S) = \prod_{i=1}^{K} P(c_i \mid w^S)$$

K=#words in context C, leaving out w

Example: modelling of WSD (1/3)

- Sentence - *He has Jupiter in the seventh house of his horoscope*, w: house, C: All words other than house
  - (*He, has, Jupiter, in, the, seventh, of, his, horoscope*)
- Word house has 3 senses (Astrological, Family, Dwelling)
- S* =$\arg\max_{S} P(S|w, C)$, where $w \in C$

  $= \arg\max_{S} P(S|w, c) = P(S|w) * P(C|S, w)$

Example: Modelling of WSD (2/3)

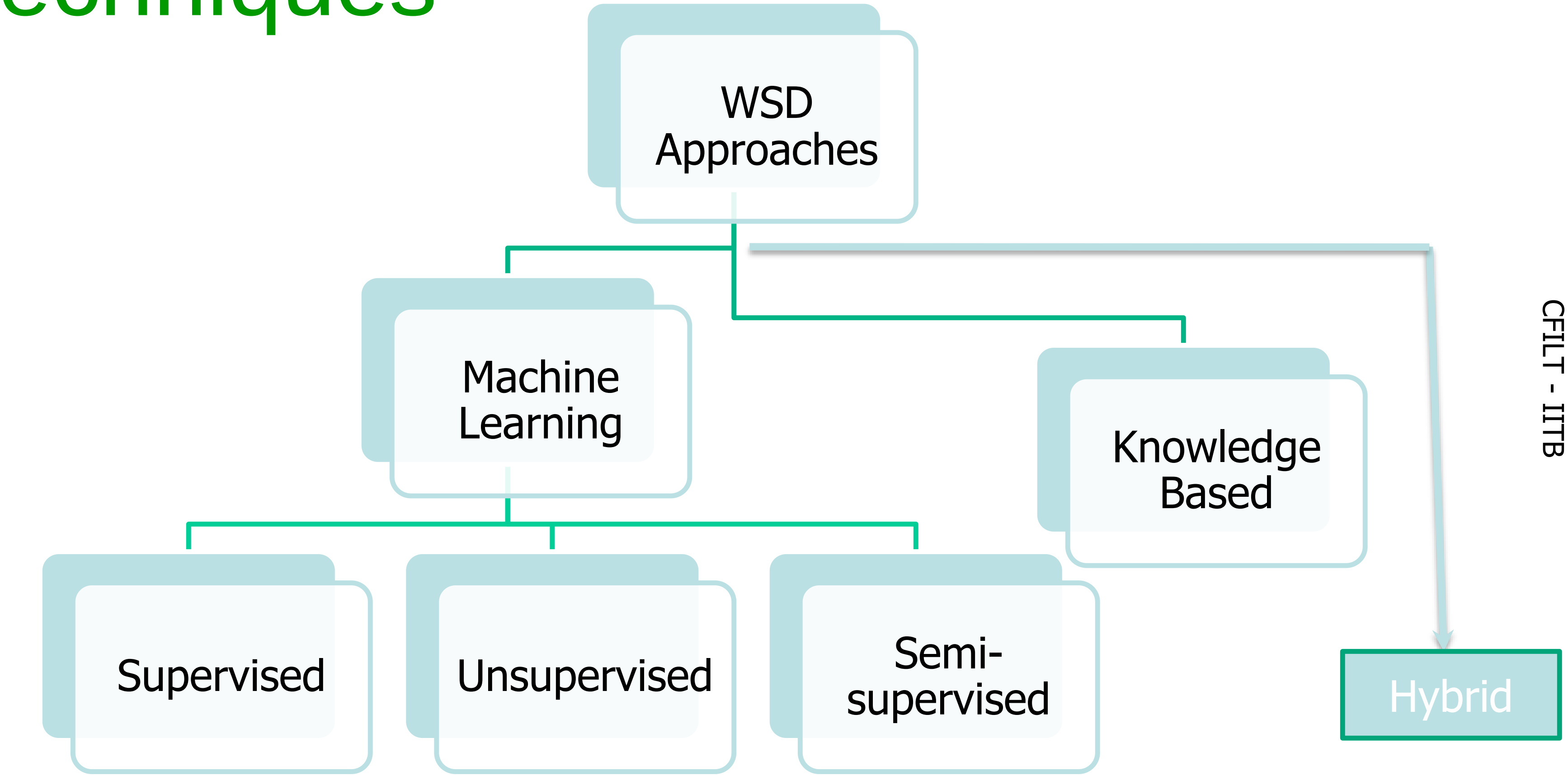
- Let S = Sense expressed by the synset id for particular sense(ex: Astrological)
- Prior : $P(S|w) = \frac{\text{number of times word house tagged in astrological sense}}{\text{number of times house appears in corpus}}$
- Likelihood :

$P(C|S,w)$ = P(He, has, Jupiter, in, the, seventh, of, his, horoscope | word house in astrological sense)

Example: Modelling of WSD (3/3)

- W^s = Word w in sense S (here S = Astrological)
- Apply chain rule
  - P(he | W^s) * P(has | he,W^s)…..P(horoscope | He,has,Jupiter,in,the,seventh,of,his, W^s)
- Make Markov assumption (Bi-gram)
  - P(he| W^s)*P(has|He, W^s)……P(horoscope|his, W^s)

Revisit: Bird's eye view of WSD techniques



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).
- These features could be sense definitions, example sentences, hypernyms etc.
- The features could also be given weights.
- The sense which has the maximum overlap is selected as the contextually appropriate sense.

LESK’S ALGORITHM

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

Context Bag: *contains the words in the context.*

E.g. “On burning ***coal*** we get ***ash***.”

From Wordnet

- The noun ash has 3 senses (first 2 from tagged texts)
- 1. (2) ash -- (the residue that remains when something is burned)
- 2. (1) ash, ash tree -- (any of various deciduous pinnate-leaved ornamental or timber trees of the genus Fraxinus)
- 3. ash -- (strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats)
- The verb ash has 1 sense (no senses from tagged texts)
- 1. ash -- (convert into ashes)

LESK'S ALGORITHM (contd..)

- Note the **importance of lower layer tasks** in NLP stack for a higher layer task like **Word Sense Disambiguation**
  - **Morphological Analysis**: Comparing the root words while finding overlap could be useful
  - Ex: 'burned' and 'burning' have the same root word in the previous example
  - **POS Tagging:** Identifying the POS tag of a

CRITIQUE

- Many times there may not be any overlap: sparsity problem
  - The ash from the combustion
- Overlap may be spurious leading to “drift”
  - *The ash tree was burned*
- Proper nouns as as strong disambiguators, but not present in WN

  E.g. **“Sachin Tendulkar”** will be a strong indicator of the category **“sports”.**

  Sachin Tendulkar plays **cricket.**

- Typical Accuracy
  - 50% when tested on 10 highly polysemous English words.

Word Vector Based WSD

Assignment

- Build an OVERLAP BASED wsd system using word embeddings aka word vectors

- Use word vectors (also called word embeddings) to do the disambiguation
- Use word2vec embeddings

Tools

a. Keras

b. TensorFlow

c. PyTorch

d. Huggingface

Keras

Keras is an open source neural network library

- written in Python
- and runs on top of TensorFlow.
- designed to enable fast experimentation with deep neural networks.
- Keras provides 2 APIs for building a NN model
  - Sequential
  - Functional.
- lets us build, train, and evaluate DNs quickly

TensorFlow

TensorFlow is open-source software library for machine learning applications

- written in C++, CUDA, Python
- designed to enable both low and high level control over the deep neural network.
- is famous for excellent functionality and high performance.

PyTorch

PyTorch is an open source machine learning library for Python

- written in Lua
- designed to enable both low and high level control over the deep neural network.
- is famous for excellent functionality and high performance.

Comparison

	Keras	**PyTorch**	**TensorFlow**
Written In	Python	Lua	C++, CUDA, Python
Speed	Slow	Fast	Fast
Implementation	Simple, concise	Harder	Complex
Debugging	Simple	Good debugging capabilities	Difficult to conduct debugging
API Level	High	Low	High and Low

Useful Links

1. https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html
2. https://keras.io/guides/
3. https://keras.io/guides/writing_a_training_loop_from_scratch/
4. https://indicnlp.org/
5. https://spacy.io/usage/spacy-101
6. https://github.com/nltk/nltk/wiki
7. https://stanfordnlp.github.io/CoreNLP/
8. https://www.tensorflow.org/tutorials

Extended Lesk's algorithm

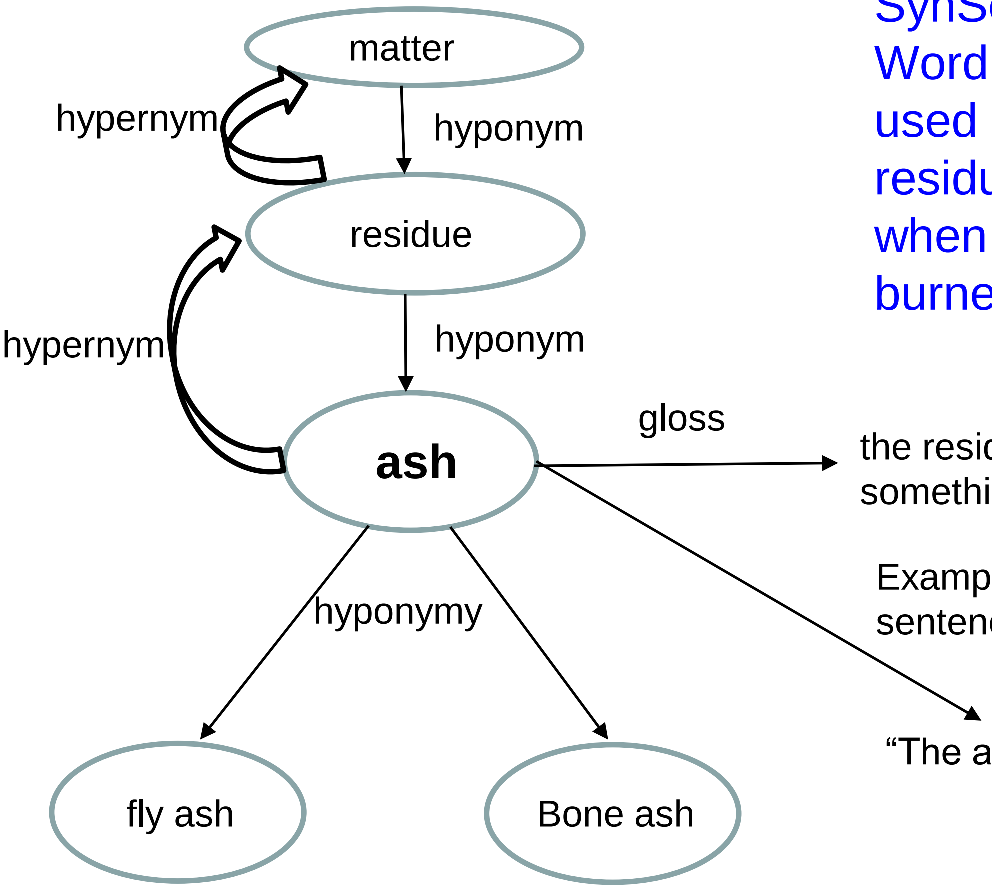
- Extension includes glosses of semantically related senses from WordNet (e.g. *hypernyms*, *hyponyms*, etc.).

- The scoring function now computes the overlap of context bag with not only the words local to the synset but also words occurring in neighjboring synsets

- Vide next slide

WordNet Sub-graph

SynSet relations from WordNet for 'ash', when used in the sense of 'the residue that remains when something is burned'



matter

hypernym

hyponym

residue

hypernym

hyponym

ash

gloss

the residue that remains when something is burned

Example sentence

“The ash tray was on the table”

hyponymy

fly ash

Bone ash

fine solid particles of ash that are carried into the air when fuel is combusted

ash left when bones burn

Example: Extended Lesk

- *“On combustion of coal we get ash”*

From Wordnet

- The noun ash has 3 senses (first 2 from tagged texts)
- 1. (2) ash -- (the residue that remains when something is burned)
- 2. (1) ash, ash tree -- (any of various deciduous pinnate-leaved ornamental or timber trees of the genus Fraxinus)
- 3. ash -- (strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats)
- The verb ash has 1 sense (no senses from tagged texts)
- 1. ash -- (convert into ashes)

Example: Extended Lesk (cntd)

- *“On combustion of coal we get ash”*

From Wordnet (through hyponymy)

- ash -- (the residue that remains when something is burned)

  => fly ash -- (fine solid particles of ash that are carried into the air when fuel is combusted)

  => bone ash -- (ash left when bones burn; high in calcium phosphate; used as fertilizer and in bone china)

Critique of Extended Lesk

- Larger region of matching in WordNet
  - Increased chance of Matching

    BUT
  - Increased chance of Topic Drift

  - E.g. for “there were some bones under the ash tree”→ Spurious overlap with bone under “bone ash”

What is overlaps tie?

- There is “tree” also in the context
- Both “bone” and “tree” will contribute equally to overlap
- Then we will invoke other factors like PROXIMITY which is also called SANNIDHI in Indian linguistic tradition (SANNIDHI means “proximity”)
- AKANGJSHA (desire), YOGYATA (suitability) and SANNIDHI (proximity) are fundamental disambiguators
- Since “tree” is *CLOSER* to “ash”, ash tree

Argument Frame Selection Preference

- “eat” and “rice”
- Eat needs and object→ akangksha (argument)
- Object should be edible, rice is edible→ yogyata (selectional preference)

WSD using Sense Embedding

- We will create the **sense embedding** by averaging the word vector for each word in the Gloss.

E.g. “On burning *coal* we get ***ash***.”

- We have three senses from Wordnet

1. ash -- (the residue that remains when something is burned)

2. ash, ash tree -- (any of various deciduous pinnate-leaved ornamental or timber trees of the genus Fraxinus)

3. ash -- (strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats)

- sense_emb = sum of word vector of each word in Gloss /# of words in Gloss

- context_emb = sum of word vector of each word in input /# of words in input

WSD using Sense Embedding (cont'd…)

- sense_emb = sum of word vector of each word in Gloss /# of words in Gloss
- context_emb = sum of word vector of each word in input /# of words in input
- Compute the cosine similarity between each sense embedding and context embedding:

  *similarity_with_sense_1 = cosine_similarity(sense_emb_1, context_emb)=**0.4675***

  similarity_with_sense_2 = cosine_similarity(sense_emb_2, context_emb) =0.4315

  similarity_with_sense_3 = cosine_similarity(sense_emb_3, context_emb)=0.4019
- The sense having the maximum cosine similarity will be the disambiguated sense for the given context word.

$$best\_sense = argmax\ (\ similarity\_with\_sense\_i\ )\ \ \forall i$$

Best sense: ash -- (the residue that remains when something is burned)

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***

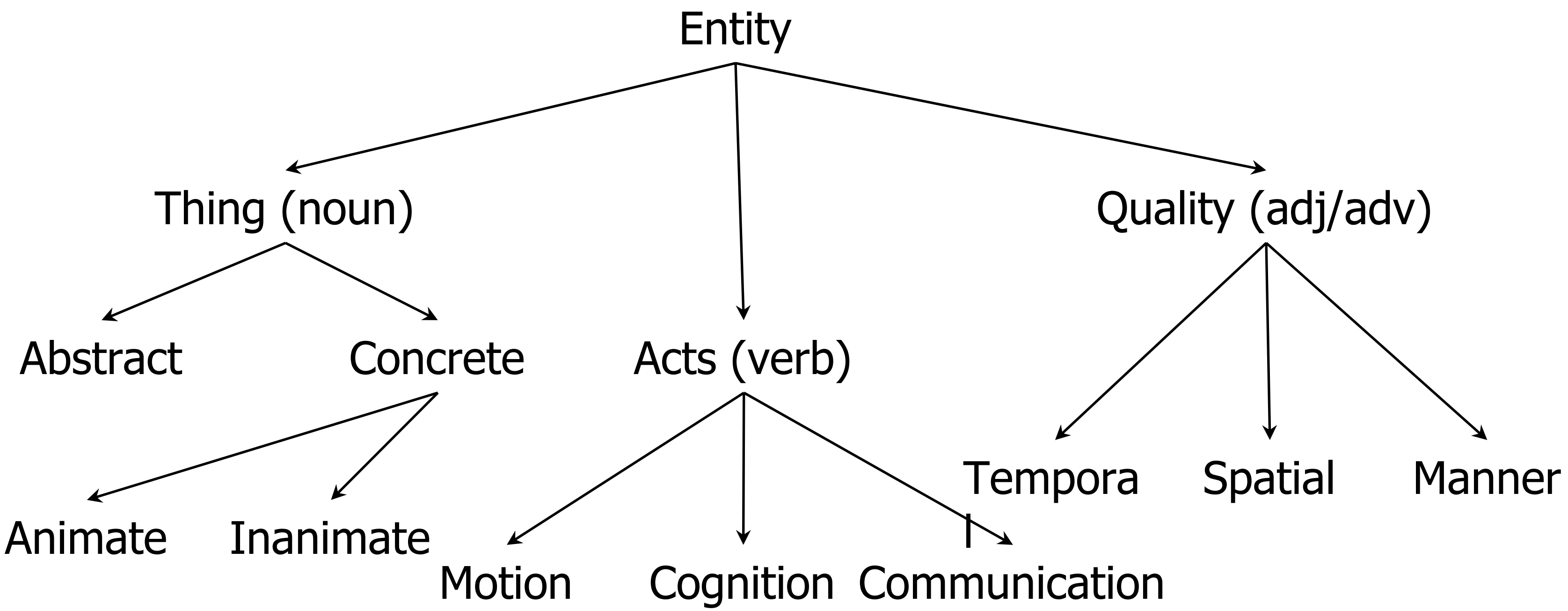
	Sense1: Finance	**Sense2: Location**
Money	+1	0
Interest	+1	0
Fetch	0	0
Annum	+1	0
Total	3	0

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

WSD USING CONCEPTUAL DENSITY *(Agirre and Rigau, 1996)*

- Select a sense based on the *relatedness* of that word-sense to the context.
- Relatedness is measured in terms of conceptual distance
  - (i.e. how close the concept represented by the ***word*** and the concept represented by its ***context words*** are)
- This approach uses a structured hierarchical semantic net (*WordNet*) for finding the conceptual distance.
- Smaller the conceptual distance higher will be the conceptual density.
  - (i.e. if all words in the context are strong indicators of a particular concept then that concept will have a higher density.)

Fundamental ontology (starting part)



Entity

Thing (noun)

Acts (verb)

Quality (adj/adv)

Abstract

Concrete

Animate

Inanimate

Motion

Cognition

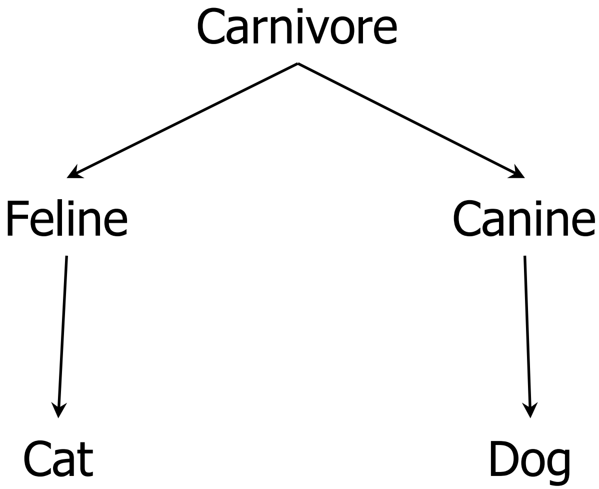
Communication

Temporal

Spatial

Manner

Path length and concept “height”



path_length(cat, dog) = 4

path_length(animate, inanimate) = 2

Animate and inanimate are more similar?

- Higher the concept, less specific it is
- Feature vector has less number of components
- Child concept inherits everything of parent plus adds its own
- Entropy is higher at higher levels of conceptual hierarchy (more heterogeneity)
- Semantic similarity will reduce at higher levels

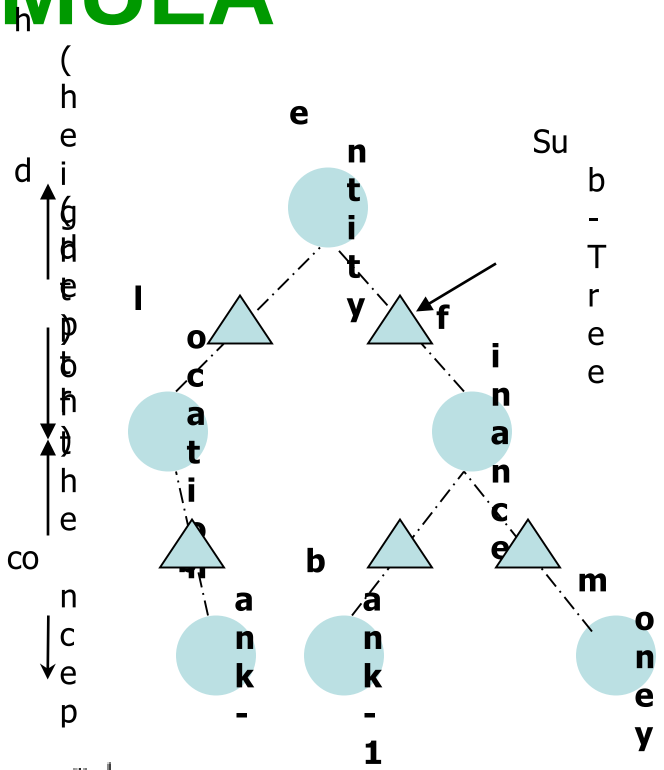
Relevance in the era of DL-NLP

- The notion of conceptual density is important for DL-NLP too
- Similarity in DL-NLP is computed by cosine similarity of word vectors
- Word vectors are created exploiting SYNTAGMATIC relations (coming from corpus)
- Ontology based similarity is computed using PARADIGMATIC relations

CONCEPTUAL DENSITY FORMULA

Wish list

- The conceptual distance between two word senses should be proportional to the length of the path between the two words in the hierarchical tree (WordNet).
- The conceptual distance between two word senses should be inversely proportional to the depth of the common ancestor concept in the hierarchy.



$$CD(c,m) = \frac{\sum_{i=0}^{m-1} nhyp^{i^{0.20}}}{descendants_c}$$

where,

c= concept

nhyp = mean number of hyponyms

h= height of the sub-hierarchy

m= no. of senses of the word and senses of context words contained in the sub-hierarchy

CD= Conceptual Density

and 0.2 is the smoothing factor

CONCEPTUAL DENSITY (cntd)

- The dots in the figure represent the senses of the word to be disambiguated or the senses of the words in context.
- The CD formula will yield highest density for the sub-hierarchy containing more senses.
- The sense of W contained in the sub-hierarchy with the highest CD will be chosen.

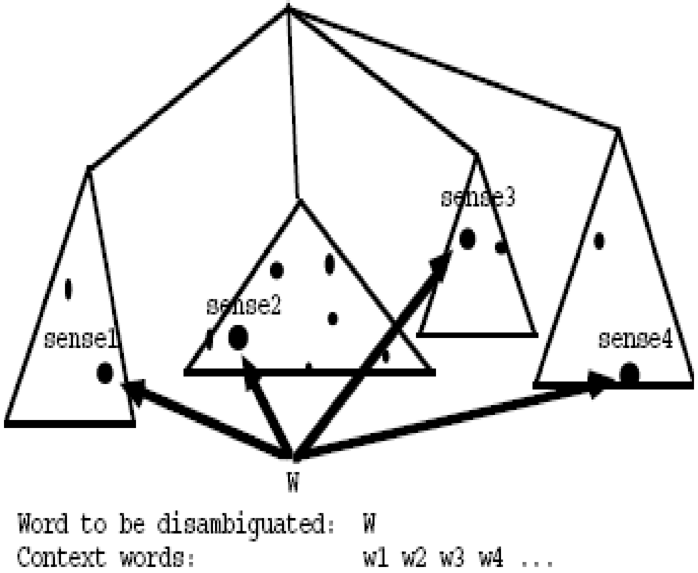
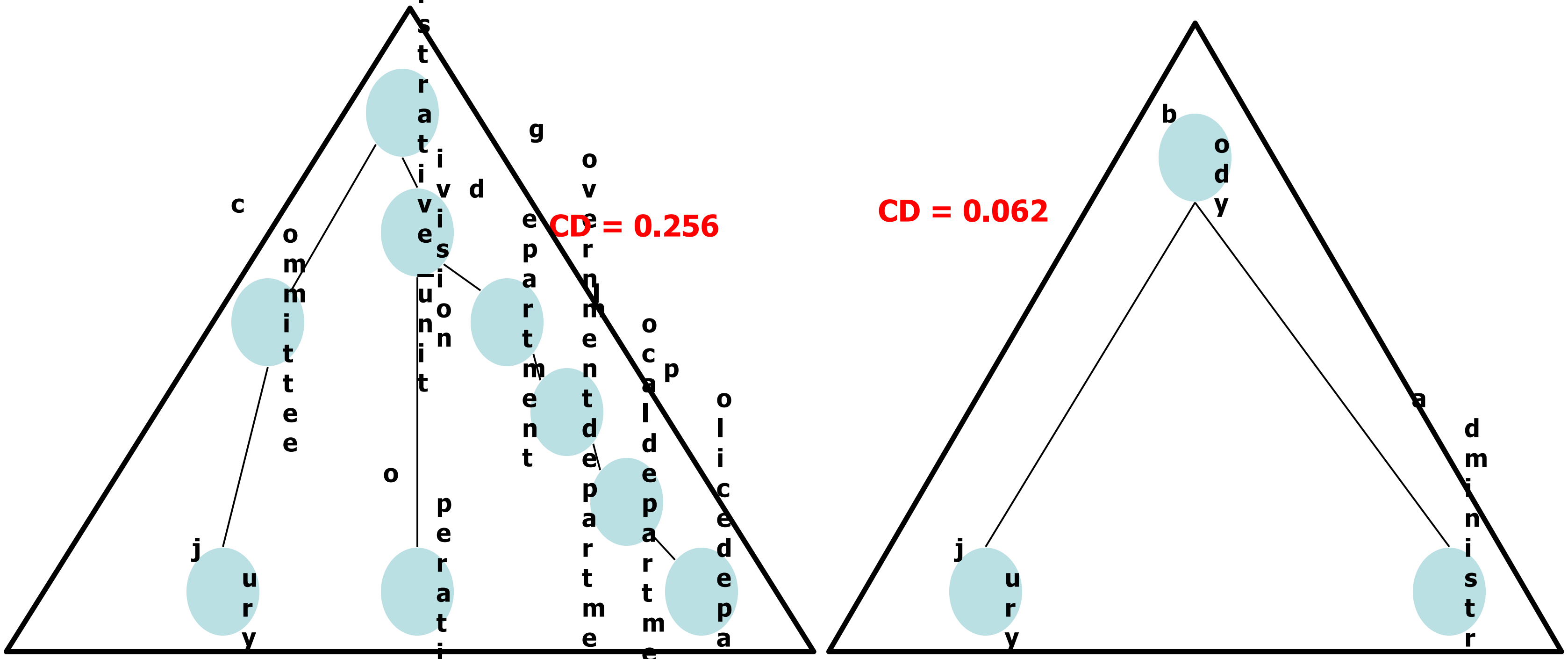


Figure 1: senses of a word in WordNet

CONCEPTUAL DENSITY (EXAMPLE)



The jury(2) praised the administration(3) and **operation** (8) of Atlanta Police Department(1)

Step 1: Make a lattice of the nouns in the context, their senses and hypernyms.

Step 2: Compute the conceptual density of resultant concepts (sub-hierarchies).

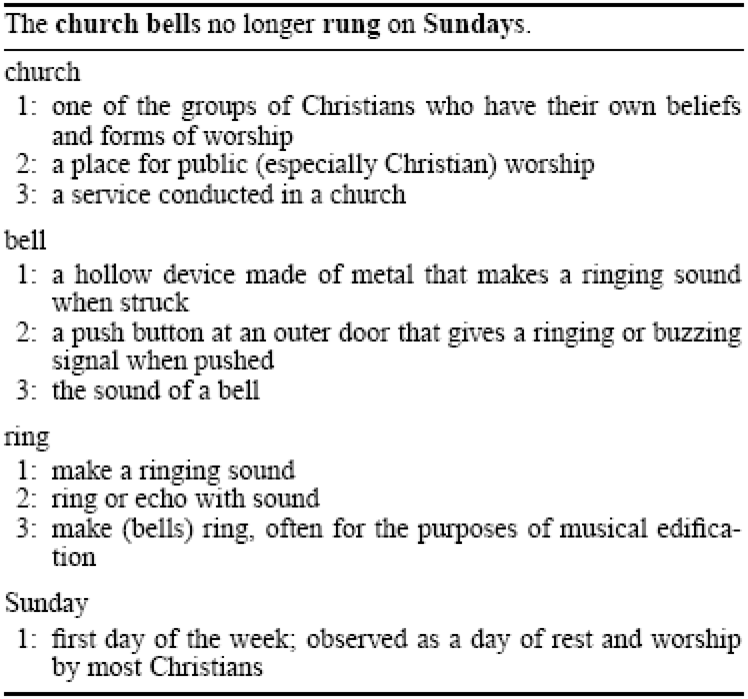
Step 3: The concept with the highest CD is selected.

Step 4: Select the senses below the selected concept as the correct sense for the respective words.

CRITIQUE

- Resolves lexical ambiguity of ***nouns*** by finding a combination of senses that maximizes the total Conceptual Density among senses.
- The Good
  - Does not require a tagged corpus.
- The Bad
  - Fails to capture the strong clues provided by proper nouns in the context.
- Accuracy
  - 54% on Brown corpus.

WSD USING RANDOM WALK ALGORITHM (Page Rank) *(sinha and Mihalcea, 2007)*



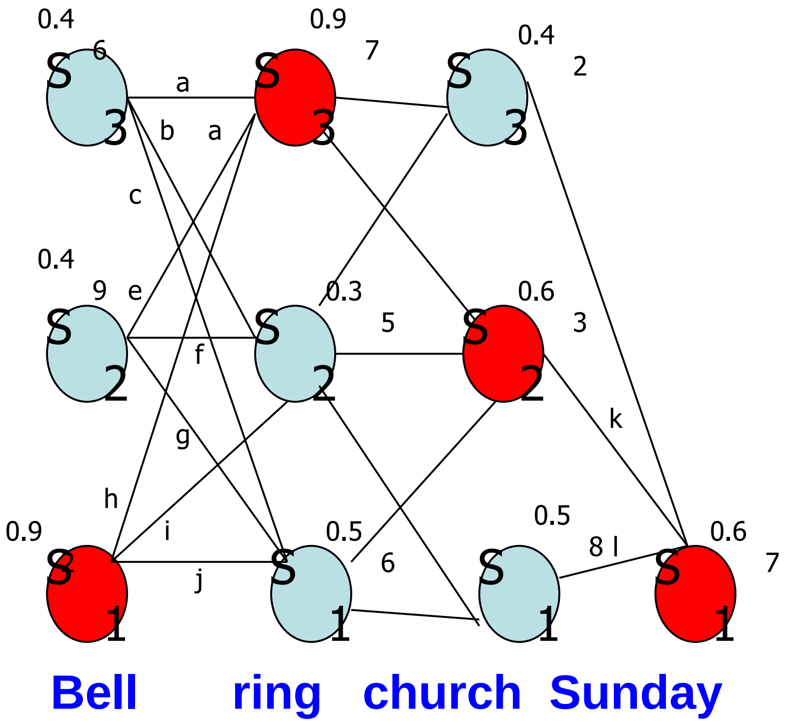
The **church bells** no longer **rung** on **Sundays**.

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.

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

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

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

A look at Page Rank (from Wikipedia)

Developed at Stanford University by Larry Page (hence the name *Page*-Rank) and Sergey Brin as part of a research project about a new kind of search engine

The first paper about the project, describing PageRank and the initial prototype of the Google search engine, was published in 1998

Shortly after, Page and Brin founded Google Inc., the company behind the Google search engine

While just one of many factors that determine the ranking of Google search results, PageRank continues to provide the basis for all of Google's web search tools

A look at Page Rank (cntd)

PageRank is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page.

Assume a small universe of four web pages: **A**, **B**, **C** and **D**.

The initial approximation of PageRank would be evenly divided between these four documents. Hence, each document would begin with an estimated PageRank of 0.25.

If pages **B**, **C**, and **D** each only link to **A**, they would each confer 0.25 PageRank to **A**. All PageRank **PR( )** in this simplistic system would thus gather to **A** because all links would be pointing to **A.**

PR(A)=PR(B)+PR(C)+PR(D)

This is 0.75.

A look at Page Rank (cntd)

Suppose that page **B** has a link to page **C** as well as to page **A**, while page **D** has links to all three pages

The *value of the link-votes is divided among all the outbound links on a page*.

Thus, page **B** gives a vote worth 0.125 to page **A** and a vote worth 0.125 to page **C**.

Only one third of **D**'s PageRank is counted for A's PageRank (approximately 0.083).

$$\mathbf{PR(A)=PR(B)/2+PR(C)/1+PR(D)/3}$$

In general,

$$\mathbf{PR(U)=\sum_{V\varepsilon B(U)} PR(V)/L(V),}$$

where B(u) is the set of pages u is linked to, and L(V) is the number of links from V

A look at Page Rank (damping factor)

The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking.

The probability, at any step, that the person will continue is a damping factor *d*.

$$\mathbf{PR(U)= (1\text{-}d)/N + d.\sum_{V \varepsilon B(U)} PR(V)/L(V),}$$

N=size of document collection

For WSD: Page Rank

- Given a graph G = (V,E)
  - In(V_i) = predecessors of V_i
  - Out(V_i) = successors of V_i

$$S(V_i) = \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

- In a weighted graph, the walker randomly selects an outgoing edge with higher probability of selecting edges with higher weight.

$$WS(V_i) = \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

Other Link Based Algorithms

- *HITS* algorithm invented by Jon Kleinberg (used by Teoma and now Ask.com)
- IBM *CLEVER project*
- *TrustRank* algorithm.

CRITIQUE

- Relies on random walks on graphs encoding label dependencies.
- The Good
  - Does not require any tagged data (a wordnet is sufficient).
  - The weights on the edges capture the definition based semantic similarities.
  - Takes into account global data recursively drawn from the entire graph.
- The Bad
  - Poor accuracy
- Accuracy
  - 54% accuracy on SEMCOR corpus which has a baseline accuracy of 37%.

KB Approaches– Comparisons

Algorithm	Accuracy
WSD using Selectional Restrictions	44% on Brown Corpus
Lesk’s algorithm	50-60% on short samples of *“Pride and Prejudice”* and some *“news stories”*.
Extended Lesk’s algorithm	32% on Lexical samples from Senseval 2 (Wider coverage).
WSD using conceptual density	54% on Brown corpus.
WSD using Random Walk Algorithms	54% accuracy on SEMCOR corpus which has a baseline accuracy of 37%.
Walker’s algorithm	50% when tested on 10 highly polysemous English words.

KB Approaches– Summary

- Drawbacks of WSD using Selectional Restrictions
  - Needs exhaustive Knowledge Base.

- Drawbacks of Overlap based approaches
  - Dictionary definitions are generally very small.
  - Dictionary entries rarely take into account the distributional constraints of different word senses (e.g. selectional preferences, kinds of prepositions, etc. □ ***cigarette*** and ***ash*** never co-occur in a dictionary).
  - Suffer from the problem of sparse match.
  - Proper nouns are not present in a MRD. Hence these approaches fail to capture the strong clues provided by proper nouns.