Introduction to Human Language Technologies
6. Word Sense Disambiguation

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Outline

Outline

Goal

- Semantic resources provide the possible senses for each word (polisemy)

lema	**PoS**	**sense**
dog	NN	1. animal
		2. (colloquial) wicked person
	VB	1. to follow
		...

- **Goal**: automatically select the right sense for an occurrence of a word in a sentence (for NN, JJ, VB and maybe ADV)

Motivation

- WSD is potentially useful for many NLP applications:
  - Speech Synthesis and Recognition
  - Acquisition of Lexical Knowledge
  - Semantic Parsing
  - Sentiment Analysis
  - IR, IE, QA, MT
  - ...
- WSD has been defined as AI-complete (Ide & Véronis, 1998)
- Unfortunately, this usefulness has not been proven yet

Motivation

- Ex.: Semantic parsing: selecting the right word sense is needed to build the meaning of the sentence

sense	gloss from WordNet 1.5
age 1	the length of time something (or someone) has existed
age 2	a historic period

He was mad about stars at the age of nine .

Motivation

- P.e.: MT: selecting the right word sense is needed to translate a word into the target language.

NOUN

1. (animal)

a. el perro (m), la perra (f)

My dog is a German Shepherd. — Mi perro es un pastor alemán.

2. (colloquial) (wicked person)

a. el bribón (m), la bribona (f)

My coworker is a lazy dog; I'm always having to do his work. — Mi colega es un bribón perezoso; siempre le tengo que estar haciendo el trabajo.

b. el canalla (m), la canalla (f) (colloquial)

That dog started cheating on his girlfriend almost as soon as they started going out. — Ese canalla le pegó cuernos a su novia prácticamente tan pronto empezaron a salir.

3. (pejorative) (unattractive woman)

TRANSITIVE VERB

4. (to follow)

a. seguir

The neighborhood bullies dogged him all the way to his house. — Los matones del vecindario lo siguieron el camino entero hasta llegar a su casa.

5. (to plague)

a. perseguir

He has been dogged by scandal his entire career. — El escándalo lo ha perseguido durante su carrera entera.

Source: `http://www.spanishdict.com`.

Outline

Resources

- Sense Definitions
  - Machine Readable Dictionaries
  - WordNets

Resources

- Sense Definitions
  - Machine Readable Dictionaries
  - WordNets
- Corpora
  - Samples with only one word labeled for each sample
    - SemEval Lexical Sample Task (training/Test corpus)
    - mainly for supervised Machine Learning algorithms

800004
Mr Purves is tight-lipped about what happens then.
He vexed rumour-mongers, who <tag '520051'>bet</> on a bid for Midlan sooner.
800005
Mr Jones loses his <tag '519914'>bet</>:1,000 people attended Cowley pools last year.

Resources

- Sense Definitions
  - Machine Readable Dictionaries
  - WordNets
- Corpora
  - Samples with only one word labeled for each sample
    - SemEval Lexical Sample Task (training/Test corpus)
    - mainly for supervised Machine Learning algorithms

    800004
    Mr Purves is tight-lipped about what happens then.
    He vexed rumour-mongers, who <tag '520051'>bet</> on a bid for Midlan sooner.
    800005
    Mr Jones loses his <tag '519914'>bet</>:1,000 people attended Cowley pools last year.

  - Samples with all the words labeled
    - Semcor, SemEval All Words Task (Test corpus)
    - mainly for unsupervised algorithms

Outline

Outline

Types of WSD Algorithms

- **Based on corpus**:
  - **Supervised approaches**:
    - Occurrences of a particular word in text annotated with their correct senses
    - Ex.: Naïve Bayes, kNN or SVM
    - word embeddings + deep learning, sense embeddings (to see in AHLT)

Types of WSD Algorithms

- **Based on corpus**:
  - **Supervised approaches**:
    - Occurrences of a particular word in text annotated with their correct senses
    - Ex.: Naïve Bayes, kNN or SVM
    - word embeddings + deep learning, sense embeddings (to see in AHLT)
  - **Semisupervised approaches**:
    - Some occurrences of the particular word annotated with their correct senses. Lots of unannotated occurrences.
    - Ex.: Yarowsky Algorithm (Bootstrapping)

Types of WSD Algorithms

- **Based on corpus**:
  - **Supervised approaches**:
    - Occurrences of a particular word in text annotated with their correct senses
    - Ex.: Naïve Bayes, kNN or SVM
    - word embeddings + deep learning, sense embeddings (to see in AHLT)
  - **Semisupervised approaches**:
    - Some occurrences of the particular word annotated with their correct senses. Lots of unannotated occurrences.
    - Ex.: Yarowsky Algorithm (Bootstrapping)
- **Knowledge-based**: from a external knowledge source
  - **Unsupervised approaches**
  - They use lexical knowledge (WordNets, machine readable dictionaries)
  - Ex.: Lesk Algorithm (available at NLTK),
    UKB (available via TextServer)

Outline

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

veo	un	banco	de	peces	desde	el	banco

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

veo	un	banco	de	peces	desde	el	banco
$\downarrow$							
M_{ver}							
$\downarrow$							
ver_1							
ver_2							
ver_3							

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

veo	un	banco	de	peces	desde	el	banco
$\downarrow$		$\downarrow$					
M_{ver}		M_{banco}					
$\downarrow$		$\downarrow$					
ver_3		$banco_1$					
		$banco_2$					
		$banco_3$					
		$banco_4$					
		$banco_5$					
		$\ldots$					

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

veo	un	banco	de	peces	desde	el	banco
$\downarrow$		$\downarrow$		$\downarrow$			
M_{ver}		M_{banco}		M_{pez}			
$\downarrow$		$\downarrow$		$\downarrow$			
ver_3		$banco_4$		pez_1			
				pez_2			
				pez_3			
				pez_4			

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

veo	un	banco	de	peces	desde	el	banco
$\downarrow$		$\downarrow$		$\downarrow$			$\downarrow$
M_{ver}		M_{banco}		M_{pez}			M_{banco}
$\downarrow$		$\downarrow$		$\downarrow$			$\downarrow$
ver_3		$banco_4$		pez_1			$banco_1$
							$banco_2$
							$banco_3$
							$banco_4$
							$banco_5$
							$\ldots$

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- **Set of categories**:

  $\{\text{sense}_1 \ldots, \text{sense}_k\}$

  Ex.:

  44 different senses of word *bajo* in Spanish (NN, JJ, VB)

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- **Set of categories**:

  $\{\text{sense}_1 \ldots, \text{sense}_k\}$

  Ex.:

  44 different senses of word *bajo* in Spanish (NN, JJ, VB)

- **Annotated corpus** :

  $\{<\text{occurrence}_i, \text{context}_i, \text{right\_sense}_i >\}$

  Ex.:

			e_1^+			e_2^+
text:	el	niño	bajo	toca	el	bajo
POS:	DT	NN	JJ	VB	DT	NN
			01206474-a			02803349-n

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- Examples:

  $\{e^+\}$: $\{<$occurence$_i$, context$_i$, correct_sense$_i$ $>\}$

  $\{e^-\}$: $\{<$occurence$_i$, context$_i$, incorrect_sense$_i$ $>\}$

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- Examples:

  $\{e^+\}$: $\{<\text{occurence}_i, \text{context}_i, \text{correct\_sense}_i >\}$

  $\{e^-\}$: $\{<\text{occurence}_i, \text{context}_i, \text{incorrect\_sense}_i >\}$

- Representation with feature vectors:
  - Local context: word+position, lemma+position, POS+position
    Ex.: *come up with* $\rightarrow$ `w+1_up, w+2_with`

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- Examples:

  $\{e^+\}$: $\{<\text{occurence}_i, \text{context}_i, \text{correct\_sense}_i>\}$

  $\{e^-\}$: $\{<\text{occurence}_i, \text{context}_i, \text{incorrect\_sense}_i>\}$

- Representation with feature vectors:
  - Local context: word+position, lemma+position, POS+position
    Ex.: *come up with* → `w+1_up, w+2_with`
  - Global context: bag of words, lemmas, bigrams or collocations
    Ex.: *I was studing at U.P.C. in Barcelona for 2 years* →
    `l+_year, co+_U.P.C._Barcelona.`

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- Examples:

  $\{e^+\}$: $\{<\text{occurence}_i, \text{context}_i, \text{correct\_sense}_i >\}$

  $\{e^-\}$: $\{<\text{occurence}_i, \text{context}_i, \text{incorrect\_sense}_i >\}$

- Representation with feature vectors:
  - Local context: word+position, lemma+position, POS+position
    Ex.: *come up with* $\rightarrow$ `w+1_up, w+2_with`
  - Global context: bag of words, lemmas, bigrams or collocations
    Ex.: *I was studing at U.P.C. in Barcelona for 2 years* $\rightarrow$
    `l+_year, co+_U.P.C._Barcelona.`
  - Syntax: syntactic functions
    Ex.: *cats eat fish.* $\rightarrow$ `subj_cat, obj_fish`

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

- Examples:

  $\{e^+\}$: $\{<\text{occurence}_i, \text{context}_i, \text{correct\_sense}_i >\}$

  $\{e^-\}$: $\{<\text{occurence}_i, \text{context}_i, \text{incorrect\_sense}_i >\}$

- Representation with feature vectors:
  - Local context: word+position, lemma+position, POS+position
    Ex.: *come up with* $\rightarrow$ `w+1_up, w+2_with`
  - Global context: bag of words, lemmas, bigrams or collocations
    Ex.: *I was studing at U.P.C. in Barcelona for 2 years* $\rightarrow$ `l+_year, co+_U.P.C._Barcelona.`
  - Syntax: syntactic functions
    Ex.: *cats eat fish.* $\rightarrow$ `subj_cat, obj_fish`
  - Semantics: domain, senses of previous words
    p.e.: the example is about *history* $\rightarrow$ `topic_history`

Exercise

We want the sentence below to be represented by local and topical features and be supplied as example for a ML algorithm:

Example He was mad about stars at the age of nine .
age.01

+ PoS ('He', 'PRP'), ('was', 'VBD'), ('mad', 'JJ'), ('about', 'IN'), ('stars', 'NNS'), ('at', 'IN'), ('the', 'DT'), ('age', 'NN'), ('of', 'IN'), ('nine', 'CD'), ('.', '.')

1. Give the bag of open-class words of the left context.
2. Give the local features in a ± 2 word window of the word forms.
3. Give two other possible local or topical features

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

Bottleneck:

- The lack of models for all the words of a given language
- The difficulty of acquiring annotated corpora for learning models

Based on Corpus: Supervised ML Approaches

WSD as a classification problem: learn a model useful to disambiguate occurrences of **a particular word** in text

Bottleneck:

- The lack of models for all the words of a given language
- The difficulty of acquiring annotated corpora for learning models

Alternatives:

- Semisupervised methods (few annotated examples and lots of unannotated ones -ex.: bootstrapping-)
- Knowledge-based methods

Outline

Based on Knowledge: Lesk Algorithm

Lesk algorithm
Disambiguates just one word within a context

$$\mathrm{Lesk}(w) = \operatorname*{argmax}_{s_i \in S(L(w))} \forall_{s_j \in S(C(w))} |\mathrm{Def}(s_i) \cap \mathrm{Def}(s_j)|$$

$\mathrm{L}(w)$: set of lemmas of word w

$\mathrm{C}(w)$: set of lemmas of open-class words in the context of w

$S(X)$: set of senses for all lemmas in X

$\mathrm{Def}(s)$: set of lemmas in the definition of sense s

Based on Knowledge: Lesk Algorithm

Example

Input: "pine cone"

PINE

1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness

CONE

1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of certain evergreen trees

Based on Knowledge: Lesk Algorithm

Example

Input: "pine cone"

PINE

1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness

CONE

1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of certain evergreen trees

Solution (sin contar las *stopwords*)

Mejor intersección: $\mathrm{Pine}\#1 \cap \mathrm{Cone}\#3 = 2$.

sense for "pine": $\mathrm{Pine}\#1$

Based on Knowledge: Lesk Algorithm

Example

Input: "pine cone"

PINE

1. kinds of evergreen tree with needle-shaped leaves
2. waste away through sorrow or illness

CONE

1. solid body which narrows to a point
2. something of this shape whether solid or hollow
3. fruit of certain evergreen trees

Solution (sin contar las *stopwords*)
Mejor intersección: $Pine\#1 \cap Cone\#3 = 2$.
sense for "cone": $Cone\#3$

Based on Knowledge: Lesk Algorithm Simplification

Simplified Lesk algorithm

$$\mathrm{Lesk}(w) = \operatorname*{argmax}_{s_i \in S(\mathrm{L}(w))} |\mathrm{Def}(s_i) \cap \mathrm{C}(w)|$$

$\mathrm{L}(w)$: set of lemmas of word w

$\mathrm{C}(w)$: set of lemmas of open-class words in the context of w.

$S(\mathrm{X})$: set of senses for all lemmas in X

$\mathrm{Def}(s)$: set of lemmas in the definition of sense s.

In general, better performance than the general Lesk algorithm

Based on Knowledge: Lesk Algorithm Exercise

Given the sentence:

- `I went to the bank to deposit money.`

and the definitions of the two first senses of the word *bank*:

1. `sloping land (especially the slope beside a body of water)`
2. `a financial institution that accepts deposits and channels the money into lending activities`

apply simplified Lesk algorithm to find the most appropriate sense among them.

Based on Knowledge: Lesk's Algorithm Extensions

Lesk algorithm suffers from low recall

Based on Knowledge: Lesk's Algorithm Extensions

Lesk algorithm suffers from low recall

Variants:

- Changing the similarity measure: Cosine
- Use of WordNet instead of a dictionary
- Enrichment with WordNet (Adapted/Extended Lesk) (Banerjee and Pederson, 2002/2003)
  - Use examples of Wordnet Synsets
  - Use the data of hypernyms and/or hyponyms
- Enrichment with WordNet and Wikipedia (Enhanced Lesk) (Basile et al. 2014)

Based on Knowledge: UKB

- Methods to disambiguate one word or all the words at the same time
- Based on **PageRank** algorithm from Google
  - **input**: net of linked webpages
  - **output**: relevance of each webpage included in the net

Based on Knowledge: UKB

- Methods to disambiguate one word or all the words at the same time
- Based on **PageRank** algorithm from Google

  input: net of linked webpages
  output: relevance of each webpage included in the net
- Analogy:

  input: text to disambiguate and graph of word senses defined by their relations (ex. WordNet)
  output: relevance of each sense of each word occurrence included in the text

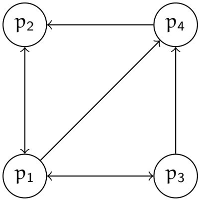
Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?
   * Webpage relevance = prob. of being visited following the links

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

   * Webpage relevance = prob. of being visited following the links



transition matrix

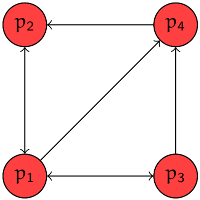
$$H = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

* Webpage relevance = prob. of being visited following the links
* Find the stationary distribution

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



transition matrix

$$H = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix}$$

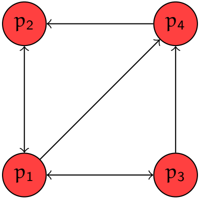
initial relevance vector

$$v_0 = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?
   * Webpage relevance = prob. of being visited following the links
   * Find the stationary distribution

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



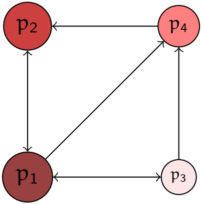
$$v_1 = \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

- Webpage relevance = prob. of being visited following the links
- Find the stationary distribution

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



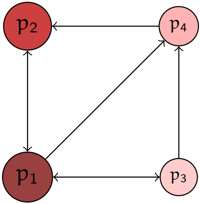
$$v_1 = \begin{bmatrix} 0.375 \\ 0.333 \\ 0.083 \\ 0.208 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

   * Webpage relevance = prob. of being visited following the links
   * Find the stationary distribution

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



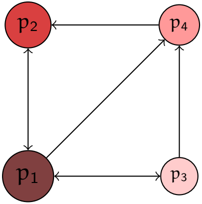
$$v_2 = \begin{bmatrix} 0.374 \\ 0.333 \\ 0.125 \\ 0.166 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.375 \\ 0.333 \\ 0.083 \\ 0.208 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

- Webpage relevance = prob. of being visited following the links
- Find the stationary distribution

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



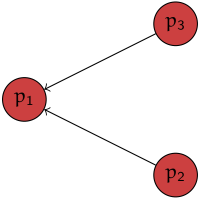
$$v_3 = \begin{bmatrix} 0.395 \\ 0.291 \\ 0.125 \\ 0.187 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1/2 & 0 \\ 1/3 & 0 & 0 & 1 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 1/2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.374 \\ 0.333 \\ 0.125 \\ 0.166 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

DRAWBACK: webpages without outcoming links and disconnected graphs

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



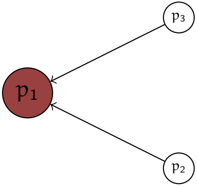
$$v_1 = \begin{bmatrix} - \\ - \\ - \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

DRAWBACK: webpages without outcoming links and disconnected graphs

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



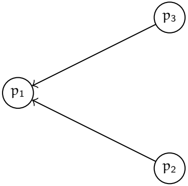
$$v_1 = \begin{bmatrix} 0.66 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

DRAWBACK: webpages without outcoming links and disconnected graphs

$$v_{(i+1)} = H \cdot v_i \qquad v_0 = [1/n]_n$$



$$v_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.66 \\ 0 \\ 0 \end{bmatrix}$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

DRAWBACK: webpages without outcoming links and disconnected graphs
SOLUTION: select a webpage randomly

$$v_{(i+1)} = H \cdot v_i \quad v_0 = [1/n]_n$$

Based on Knowledge: UKB

1. PRELIMINARY: How does **PageRank** perform?

DRAWBACK: webpages without outcoming links and disconnected graphs
SOLUTION: select a webpage randomly

$$v_{(i+1)} = M \cdot v_i \qquad v_0 = [1/n]_n$$

$$M = (1 - \alpha) \cdot H + \alpha \cdot B$$

M: PageRank matrix

H: transition matrix

α: probability of random selection (default 0.15)

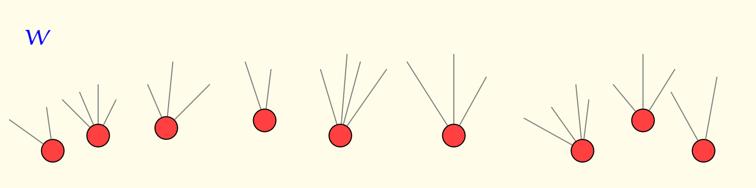
B: matrix $[1/n]_n^n$

Based on Knowledge: UKB

2. WSD using PageRank

* Use of WordNet as graph

$$v_{(i+1)} = M_W \cdot v_i \qquad v_0 = [1/|W|]_{|W|}$$



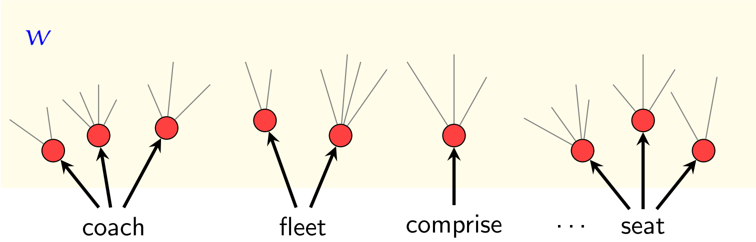
Based on Knowledge: UKB

2. WSD using PageRank

- Use of WordNet as graph

$$v_{(i+1)} = M_W \cdot v_i \qquad v_0 = [1/|W|]_{|W|}$$

- Focused on the synsets of the input words



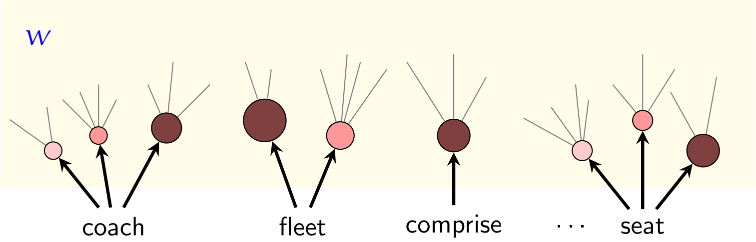
Based on Knowledge: UKB

2. WSD using PageRank

* Use of WordNet as graph

$$v_{(i+1)} = M_W \cdot v_i \qquad v_0 = [1/|W|]_{|W|}$$

* Focused on the synsets of the input words

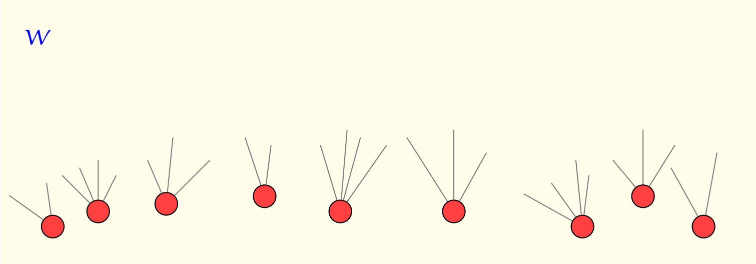


Based on Knowledge: UKB

2. WSD using PageRank

How does it focus on the synsets of the k input words?

$$v_{(i+1)} = M_W \cdot v_i \qquad v_0 = [1/|W|]_{|W|}$$



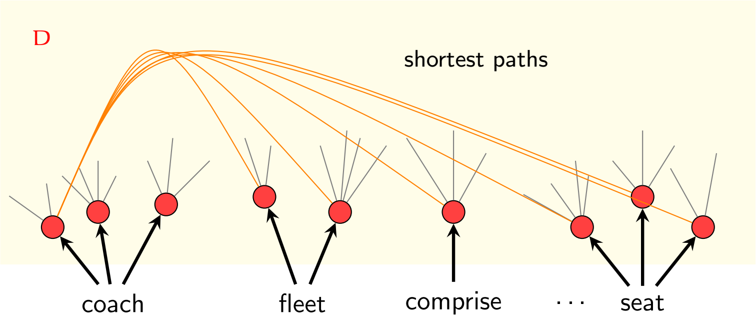
Based on Knowledge: UKB

2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 1. Restrict W to the disambiguation graph D

$$v_{(i+1)} = M_D \cdot v_i \qquad v_0 = [1/|D|]_{|D|}$$



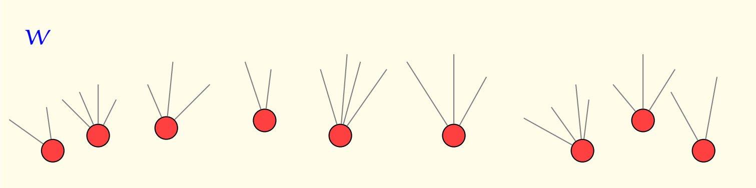
Based on Knowledge: UKB

2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 2. Personalize B to the k input words

$$v_{(i+1)} = M_W \cdot v_i \qquad v_0 = [1/|W|]_{|W|}$$



Based on Knowledge: UKB

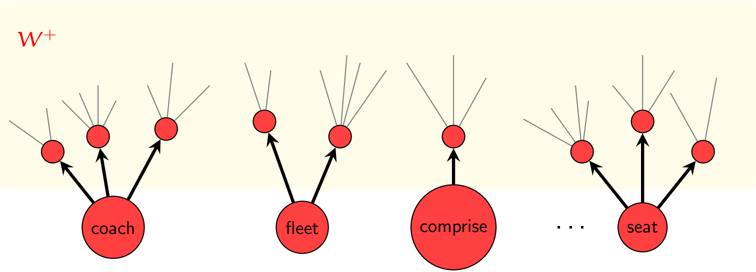
2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 2. Personalize B to the k input words

$$v_{(i+1)} = M_{W^+} \cdot v_i \qquad v_0 = [1/|W| + k]_{|W|+k}$$

- Add the k words as new nodes linked to their n' synsets



Based on Knowledge: UKB

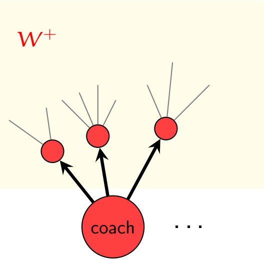
2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 2. Personalize B to the k input words

$$M_{W^+} = (1-\alpha) \cdot H_{W^+} + \alpha \cdot B_{W^+}$$

- Add the k words as new nodes linked to their n' synsets



$$H_{W^+} = \begin{array}{c} \begin{array}{cc} \overbrace{\qquad\quad}^{|W|} & \overbrace{\qquad}^{k} \end{array} \\ \left[\begin{array}{c:cc} & 0 & \dots \\ & 1/3 & \dots \\ & 1/3 & \dots \\ H_W & 0 & \dots \\ & \dots & \\ & 0 & \dots \\ & 1/3 & \dots \\ \hdashline 0 & 0 & \end{array}\right] \begin{array}{l} \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\right\} |W| \\ \left.\begin{array}{c} \\ \end{array}\right\} k \end{array} \end{array}$$

Based on Knowledge: UKB

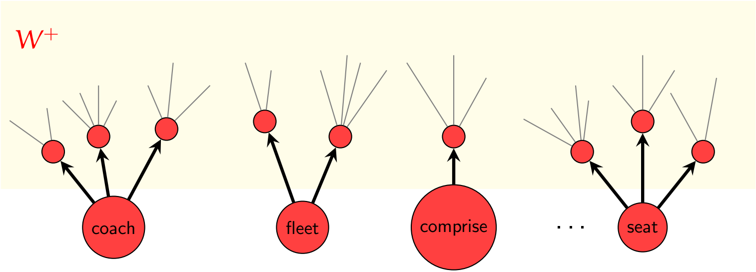
2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 2. Personalize B to the k input words

$$M_{W^+} = (1-\alpha) \cdot H_{W^+} + \alpha \cdot B_{W^+}$$

- Concentrate the random selection prob. on the n' synsets



Based on Knowledge: UKB

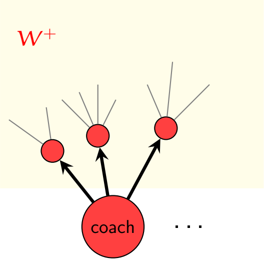
2. WSD using PageRank

How does it focus on the synsets of the k input words?

OPTION 2. Personalize B to the k input words

$$M_{W^+} = (1-\alpha) \cdot H_{W^+} + \alpha \cdot B_{W^+}$$

- Concentrate the random selection prob. on the n' synsets



$$B_{W^+} = \begin{array}{cc} \begin{array}{cc} \overbrace{\hphantom{0000000}}^{|W|} & \overbrace{\hphantom{0000000}}^{k} \end{array} & \\ \left[\begin{array}{c:cc} & 0 & \cdots \\ & 1/n' & \cdots \\ & 1/n' & \cdots \\ 0 & 0 & \cdots \\ & \cdots & \\ & 0 & \cdots \\ & 1/n' & \cdots \\ \hdashline 0 & \multicolumn{2}{c}{0} \end{array}\right] & \begin{array}{l} \left.\vphantom{\begin{array}{c}0\\0\\0\\0\\0\\0\\0\end{array}}\right\}|W| \\ \}\,k \end{array} \end{array}$$