

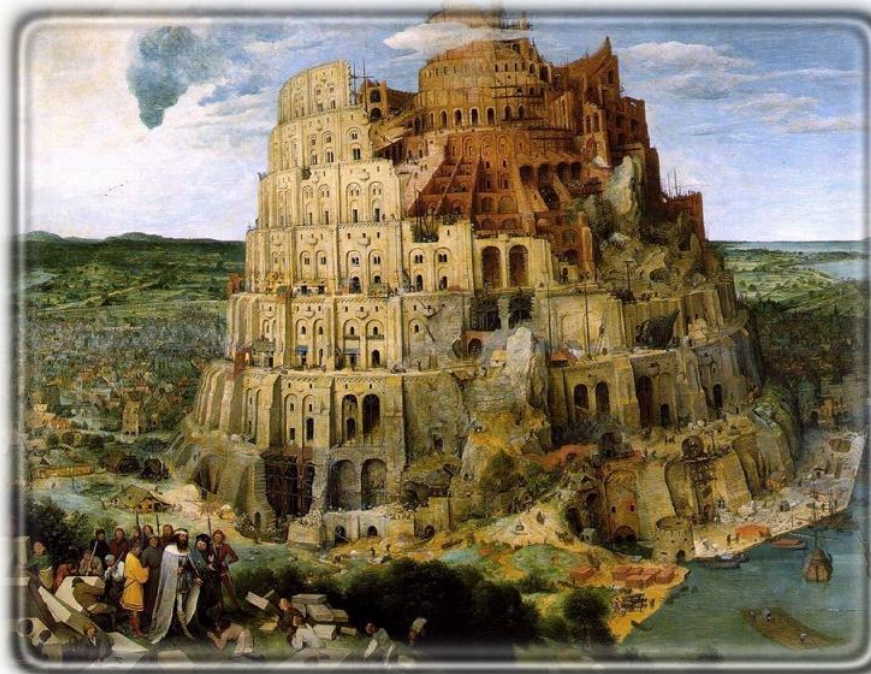
Methods and techniques for NLP



TECHNISCHE
UNIVERSITÄT
DARMSTADT

An introduction to: „ Word Sense Disambiguation“

perform?
game?
maneuver?
play
represent?
turn?
act?
shut?
near?
end?
close
adjacent?
complete?
come together?



stop?
violate?
destroy?
break
fault?
interrupt?
time out?
burn?
trim?
swing?
reduce?
cut
edit out?
split?
stop?

Table of contents



- **Motivation**
- **Introduction**
- **Variants of WSD**
- **Approaches to WSD**
- **Appendix**
- **References**

Table of contents

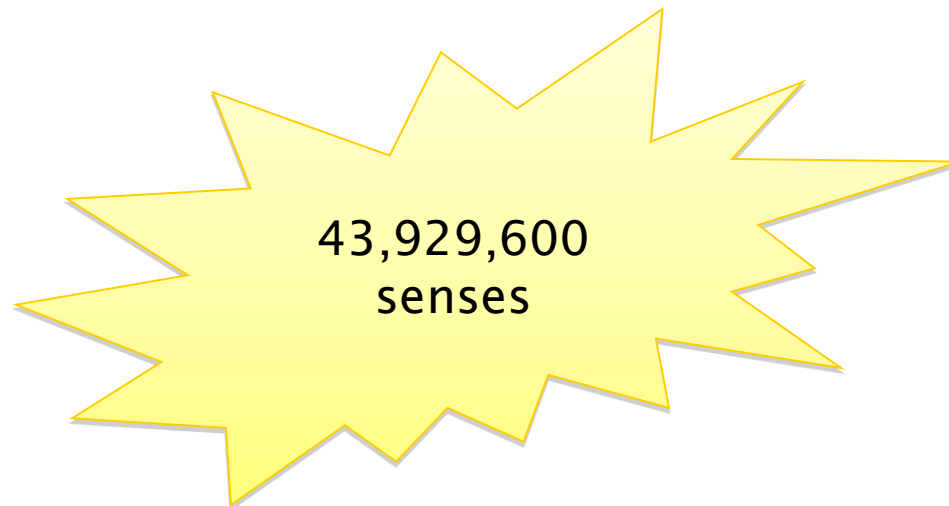
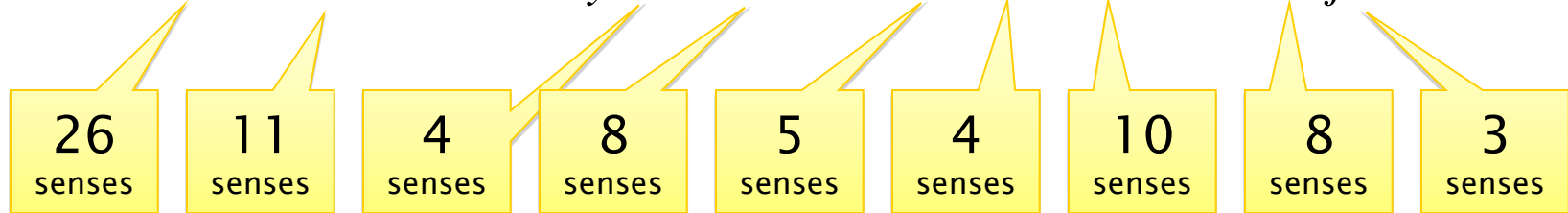


- **Motivation**
- Introduction
- Variants of WSD
- Approaches to WSD
- Appendix
- References

Motivation

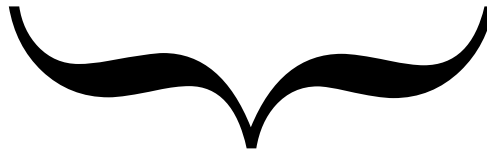
Assume a computer should “try” to understand the following sentence...

I saw a man who is 98 years old and can still walk and tell jokes



[BMCWSD09]

- Computer must **disambiguate** the **senses** for all the ambiguous **words** to understand the whole sentence...
- Put it all together: **words** + **senses** + **disambiguate**



Word Sense Disambiguation (WSD)

- ...and we get one of the central challenges in NLP !
(WSD is declared as a "**Open problem**")

"In science and mathematics, an open problem or an open question is a known problem that can be accurately stated, and has not yet been solved (no solution for it is known)..." [[Wikipedia](#)]

Motivation = Demotivation ???



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Are you still motivated...?

Motivation = Demotivation ???



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Are you still motivated...?

Really ?

Motivation = Demotivation ???



Are you still motivated...?

Really ?

OK, give me ~40 minutes 😊

Table of contents

✓ Motivation

➤ Introduction

- What is WSD ?
- What is WSD used for ?
- Ambiguity for humans and computers

➤ Variants of WSD

➤ Approaches to WSD

➤ Appendix

➤ References

What is WSD ?

- WSD is the task of assigning sense labels to occurrences of an ambiguous word...
- The WSD Problem can be divided into **two** subproblems:

- 1. Sense discrimination** (**simple** to handle)

Determining the class to which an occurrence belongs

- 2. Sense labeling** (**difficult** ! → focus of this presentation!)

Determining the sense of each class

[AWSDHS98]

What is WSD ?

- Note

“WSD itself is not a standalone application !”



...however, WSD is **acutely necessary** to accomplish NLP tasks →

What is WSD used for ?



- **Machine translation**

WSD is essential for the proper translation of words such as the French “grille”, which (depending on the context) can be translated as: railings, gate, bar, grid, scale, schedule, etc.

- **Information retrieval & hypertext navigation**

When searching for specific keywords, it's desirable to eliminate occurrences in documents where the word/words are used in an inappropriate sense, e.g. searching for judicial references → eliminate documents containing the word “court” as associated with “royalty”, rather than with “law”

- **Text processing**

WSD is necessary for spelling correction, e.g. to determine when diacritics should be inserted (e.g. in French, changing *comte* to *comté*), case changes (“HE READ THE TIMES” → “He read the Times”) and also for lexical access of Semitic languages (where vowels aren't written), etc.

[SOTAWSD98]

What is WSD used for ?

- **Grammatical analysis**

WSD is useful for POS Tagging, e.g. in the French sentence: "L'étagère plie sous les livres" ("The shelf is bending under [the weight of] the books"), it's necessary to disambiguate the sense of livres (which can mean books or pounds and is masculine in the former sense, feminine in the latter) to properly tag it as a masculine noun. WSD is also necessary for certain syntactic analyses, such as prepositional phrase attachment

- **Speech processing**

WSD is required for correct phonetization of words in speech synthesis, e.g. the word conjure in "He conjured up an image" or in "I conjure you to help me" and also for word segmentation and homophone discrimination in speech recognition

- **Content & thematic analysis**

A common approach → analyze distribution of pre-defined categories of words i.e., words indicative of a given concept, idea, theme, etc. across a text. The need for WSD in such analysis has long been recognized in order to include only those instances of a word in its proper sense

[SOTAWSD98]

What is WSD used for ?

- **Note**

- Different NLP applications require **different degrees** of disambiguation, e.g.:
- Information Retrieval → demands shallow WSD
- Machine Translation → requires a much higher WSD-precision to generate translations, that sounds “natural” in target language

Ambiguity for humans and computers

Conclusion so far: Polysemy →

"Many words have many possible meanings"

- **Computer vs. human**

- A **computer** has no basis for knowing which "sense" is appropriate for a given word (even if it is obvious to a human...)
- For **humans** ambiguity is rarely a problem in their day-to-day communication (except in extreme cases...)

- **Question**

- How is it possible for a computer to distinguish between several senses of a given word?

Ambiguity for humans and computers



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Answer

- Cannot be centralized within one simple sentence ☹️
- Therefore: "***Divide et impera***"
 - Decompose question into smaller parts and try to answer them...
- What does a computer need in order to start a disambiguation process...?

Ambiguity for humans and computers

- Generally a computer relies on **two** major sources of information:

- **1. Context**

- Together with extra-linguistic information about the text such as situation

→ **data-driven**

- **2. External knowledge sources**

- Dictionaries
- Thesauri
- Parallel corpora
- Hand-labeled training sets
- Lexical databases

→ **knowledge-driven**

Table of contents

✓ Motivation

✓ Introduction

➤ **Variants of WSD**

- Targeted WSD
- All Words WSD

➤ Approaches to WSD

➤ Appendix

➤ References

Variants of WSD

- Before looking at the algorithms in detail, it should be clear to know **WHAT** exactly has to be disambiguated...
- What does it mean?
- WSD is a very expensive task, e.g.:
 - Execution time, querying external knowledge base sources, etc.
- Save complexity → disambiguate only what is important for a given task...
- Useful to distinguish two variants of the generic WSD task:
 - 1) Targeted WSD → **one** specific word in a sentence
 - 2) All Words WSD → **any** open-class word (similar to POS-tagging)

Targeted WSD

- Disambiguate only **one** target word **X**

"An electric guitar and **bass** player stand off to one side..."

- Before a disambiguation process can start, it's very important to "look" around **X** and collect some potentially useful information...
- Use a so-called "Context-Window" consisting of n word(s) around **X**

"An electric guitar and **bass** player stand off to one side..."

- Then, annotate all **words** except the target word
 - Typical annotations are: lemmas, POS-tags, frequency, ...
 - These annotations can be used in a later process...

Targeted WSD

- Why is a Context-Window so important?
- Provides **evidence** of local syntactic context
- Gives general topical cues of the context
 - **Improving Context-Window**
 - Use feature selection to determine a smaller set of words that help discriminate possible senses
 - Remove common “stop words” such as articles, prepositions, etc.
 - **Typical to include**
 - Single-word, Two-word, Three-word Context Window
 - Some authors mention to take a Context-Window of 2^n+1 words

All Words WSD

- Attempt to disambiguate **all** open-class words in a text:

*“He **put** his **suit** over the **back** of the **chair**”*

- Knowledge-based approach:
 - Use information from dictionaries
 - definitions / examples for each meaning
 - find similarity between definitions and current context
- Position in a semantic network
 - Find that “**table**” is closer to “**chair/furniture**” than to “**chair/person**”
- Use discourse properties
 - A word exhibits the same sense in a discourse / in a **collocation**

All Words WSD



- Attempt to disambiguate **all** open-class words in a text:

“He **put**...”

- Knowledge-based

- Use information

→ definitions /

→ find similarity

Collocation → means the co-occurrence of two (or more) words, which only make sense if they're combined together...

Example: **fast food**, **hot pants**, etc...

- Position in a semantic

- Find that “**table**” is closer to “**chair**” than to “**chair/person**”

- Use discourse properties

- A word exhibits the same sense in a discourse / in a **collocation**

Table of contents

✓ Motivation

✓ Introduction

✓ Variants of WSD

➤ **Approaches to WSD**

- Knowledge-Based Disambiguation
- Supervised Disambiguation
- Unsupervised Disambiguation

➤ Appendix

➤ References

Table of contents

✓ Motivation

✓ Introduction

✓ Variants of WSD

➤ **Approaches to WSD**

- Knowledge-Based Disambiguation
- Supervised Disambiguation
- Unsupervised Disambiguation

➤ Appendix

➤ References

Approaches to WSD (Overview)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **Knowledge-Based Disambiguation (KBD)**

- Rely on **external knowledge** resources (e.g. WordNet, Thesaurus, etc.)
- May use grammar rules for disambiguation
- May use hand coded rules for disambiguation

- **Supervised Disambiguation**

- Based on a **labeled training set**
- The learning system has:
 - A training set of feature-encoded inputs
AND their appropriate sense label (category)

- **Unsupervised Disambiguation**

- Based on **unlabeled corpora**
- The learning system has:
 - A training set of feature-encoded inputs
BUT NOT their appropriate sense label (category)

Approaches to WSD (Overview)

- **Knowledge-Based Disambiguation (KBD)**

- Rely on external knowledge resources (e.g. WordNet, Thesaurus, etc.)
- May use grammar rules for disambiguation
- May use hand coded rules for disambiguation

- **Supervised Disambiguation**

- Based on a labeled training set
- The learning system has:
 - A training set of feature-encoded inputs
AND their appropriate sense label (category)

- **Unsupervised Disambiguation**

- Based on unlabeled corpora
- The learning system has:
 - A training set of feature-encoded inputs
BUT NOT their appropriate sense label (category)

Note: besides these, a variety of other approaches exists for WSD

See Appendix for more details...

KBD: Task Definition

- KBD = class of WSD methods relying mainly on knowledge drawn from dictionaries and/or raw text...
- **Resources**
 - Machine Readable Dictionaries (MRD)
 - Raw corpora (pure textual data → **not manually** annotated!)
- **Scope**
 - All open-class words (nouns, verbs, adjectives, etc.)

Machine Readable Dictionaries

- In recent years, most dictionaries made available in Machine Readable format, e.g.:
 - [Oxford English Dictionary](#)
 - [Collins COBUILD](#)
 - [Longman Dictionary of Ordinary Contemporary English \(LDOCE\)](#)
- Thesauruses – add synonymy information
 - [Roget Thesaurus](#)
- Semantic networks – add semantic relations
 - [WordNet](#) (→ **next slides...**)
 - [Wortschatz \(University of Leipzig\)](#)
 - [EuroWordNet](#)

- A detailed **lexical database** of semantic relationships between English words (developed at the Princeton University)
- **Some technical facts**
 - WordNet's latest version is 3.0 (released: 2006)
 - Contains about 150,000 English words
 - Distinguishes between 4 POS types: { Nouns, Adjectives, Verbs, Adverbs }
 - Grouped into about 115,000 synonym sets called **synsets** for a total of 207,000 word-sense pairs
 - Size of database (in compressed form) about 12 Mbyte
 - Many wrappers for many programming languages freely available (→ Appendix)

WordNet: Synset relationships



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- **Antonym** **male** → **female** (opposite)
- **Attribute** **benevolence** → **good** (noun to adjective)
- **Pertainym** **alphabetical** → **alphabet** (adjective to noun)
- **Synonym** **buy** → **purchase** (diff. words with “similar meanings”)
- **Cause** **killed** → **dead** (**A** sugg. truth of **B**, but **doesn't require** it)
- **Entailment** **assassinated** → **dead** (**A requires** truth of **B**)
- **Holonym** **chapter** → **text** (part-of)
- **Meronym** **computer** → **cpu** (whole-of)
- **Hyponym** **tree** → **plant** (specialization)
- **Hypernym** **fruit** → **apple** (generalization)

WordNet: Conclusion

- The literature shows that WordNet has been used successfully for the WSD task...
- Some authors mention using WordNet with their systems leads to correct solutions up to **57%**
- Hmm...doesn't satisfy!
 - Any other possibilities to get higher accuracy?
- Yes! Mihalcea and Moldovan [RDWSD99] report better results when WordNet is combined and cross-checked with other sources → improving up to **92%**

Keep in mind: Different corpora leads often to different senses (rely on 1 corpus is easier...)

Algorithms based on MRD: Lesk Algorithm

- In 1986 the Lesk algorithm has first been implemented in its simple form by Michael Lesk [MLESK04]
- Assumption: words in a given neighbourhood tend to share a common topic...
- Use a (scored) overlap for their **dictionary definitions**
- **Pseudo- Algorithm**

Step 1: Retrieve from MRD all sense definitions of the words to be disambiguated

Step 2: Determine the definition overlap for all possible sense combinations

Step 3: Choose senses that **lead to highest** overlap

Algorithms based on MRD: Lesk Algorithm

- In 1986 the Lesk algorithm has first been implemented in its simple form by Michael Lesk [MLESK04]
- Assumption: words in a given neighbourhood tend to share a common topic...
- Use a (scored) overlap for their dictionary definitions
- **Pseudo- Algorithm**

Note: these definitions are good indicators of the senses they define !

Step 1: Retrieve from MRD all sense definitions of the words to be disambiguated

Step 2: Determine the definition overlap for all possible sense combinations

Step 3: Choose senses that lead to highest overlap

Algorithms based on MRD: Lesk Algorithm

- **Example**

- Assume we have the following word group: "...Pine Cone..."
- Our task here is to disambiguate **Pine** and **Cone**
- MRD provides for both the following definitions...

- **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

Algorithms based on MRD: Lesk Algorithm

- **Example**

- Assume we have the following word group: "...Pine Cone..."
- Our task here is to disambiguate **Pine** and **Cone**
- MRD provides for both the following definitions...

- **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

Calculate overlap:

Pine#1	∩	Cone#1	=	0
Pine#2	∩	Cone#1	=	0
Pine#1	∩	Cone#2	=	1
Pine#2	∩	Cone#2	=	0
Pine#1	∩	Cone#3	=	2
Pine#2	∩	Cone#3	=	0

Algorithms based on MRD: Lesk Algorithm

• Example

- Assume we have the following word group: "...Pine Cone..."
- Our task here is to disambiguate **Pine** and **Cone**
- MRD provides for both the following definitions...

• 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

Calculate overlap:

Pine#1 \cap Cone#1 = 0
Pine#2 \cap Cone#1 = 0
Pine#1 \cap Cone#2 = 1
Pine#2 \cap Cone#2 = 0
Pine#1 \cap Cone#3 = 2
Pine#2 \cap Cone#3 = 0

Algorithms based on MRD: Lesk Algorithm

- How does the overlap exactly work?
- Actually the \cap is not a “real” intersection...
- First: **clean up** words in the Context-Window
(e.g. apply regular expressions, replace/remove **noise**)
- After that, letter-cases have to be **ignored** (e.g. lowercase)
- And last but not least: stemm the tokens (necessary to **avoid flexion**)
- Now use a “real” intersection and score each match by adding +1

Algorithms based on MRD: Lesk Algorithm (variants)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

• Simplified Lesk

- Retrieve all sense definitions of target word from MRD
- Compare with words in context (instead: "sense definitions of words")
- Choose the sense with the most overlap

• Corpus Lesk

- Include SEMCOR sentences ([next slide](#)) in signature for each sense
- Weight words by inverse document frequency (IDF)
- $IDF(w) = -\log P(w)$
- Best-performing Lesk variant
- Used as a (strong) baseline in [SENSEVAL](#)

term = t
document frequency = df
total number of documents in a collection = N
inverse document frequency $idf = \log\left(\frac{N}{df_t}\right)$

Algorithms based on MRD: Lesk Algorithm (variants)

Semcor sentence: "In fig. 6) are slipped into place across the roof beams,"

only 1 sense in wordnet

Indicates: synset assigned to this word by the human annotators that created SEMCOR

```
<contextfile concordance="brown">
<context filename="br-h15" paras="yes">
.....
<wf cmd="ignore" pos="IN">in</wf>
<wf cmd="done" pos="NN" lemma="fig" wnsn="1" lexsns="1:10:00::">fig.</wf>
<wf cmd="done" pos="NN" lemma="6" wnsn="1" lexsns="1:23:00::">6</wf>
<punc>)</punc>
<wf cmd="done" pos="VBP" ot="notag">are</wf>
<wf cmd="done" pos="VB" lemma="slip" wnsn="3" lexsns="2:38:00::">slipped</wf>
<wf cmd="ignore" pos="IN">into</wf>
<wf cmd="done" pos="NN" lemma="place" wnsn="9" lexsns="1:15:05::">place</wf>
<wf cmd="ignore" pos="IN">across</wf>
<wf cmd="ignore" pos="DT">the</wf>
<wf cmd="done" pos="NN" lemma="roof" wnsn="1" lexsns="1:06:00::">roof</wf>
<wf cmd="done" pos="NN" lemma="beam" wnsn="2" lexsns="1:06:00::">beams</wf>
<punc>,</punc>
```

Algorithms based on MRD: Lesk Algorithm

- **Question:** Does the Lesk Algorithm works for more than two words?
- Recall the sentence from the intro:

"I saw a man who is 98 years old and can still walk and tell jokes"

- 43,929,600 sense combinations
 - Lesk Algorithm will take a while here 😞
- In 1992 J. Cowie, J. & L. Guthrie invented a acceptable workaround:
"Simulated Annealing Algorithm" [LDJJG92]

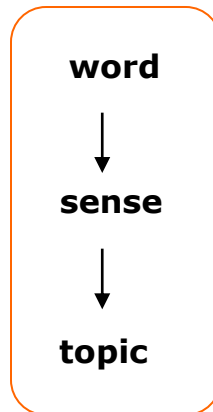
Excluded in this presentation...

Walker's Algorithm



- A thesaurus based approach (invented by: **Walker, 1987**)
- Exploits semantic categorization provided by a thesaurus (e.g. [Roget Thesaurus](#))

Word	Sense	Thesaurus category (Roget)
bass	musical senses	music
	fish	animal, insect
star	space object	universe
	celebrity	entertainer
	star shaped object	insignia
interest	curiosity	reasoning
	advantage	injustice
	financial	debt
	share	property



- Each word is assigned one or more **subject codes** in the dictionary
- If the word is assigned several subject codes, then:
 - assume that they corresponds to different senses of the word

Walker's Algorithm

• Algorithm

Step 1: For each sense of the target word → find thesaurus category to which that sense belongs

Step 2: Calculate score for each sense by using the context words

→ context words will add +1 to score of the sense **if** thesaurus category of the word matches that of the sense...

• Example

• "The *money* in this *bank* fetches an *interest* of 8% per *annum*"

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

Clue words from the context = { *money*, *interest*, *annum*, *fetch* }

[MMKWSD06]

Walker's Algorithm

• Problem

- A general categorization of words into topics is often unsuitable for a particular domain
 - Mouse → mammal, electronic device
 - When in a computer manual... ☹️
- A general topic categorization may also have a problem of coverage
 - Martina Navrátilová → sports
 - When entry is not found in the thesaurus... ☹️

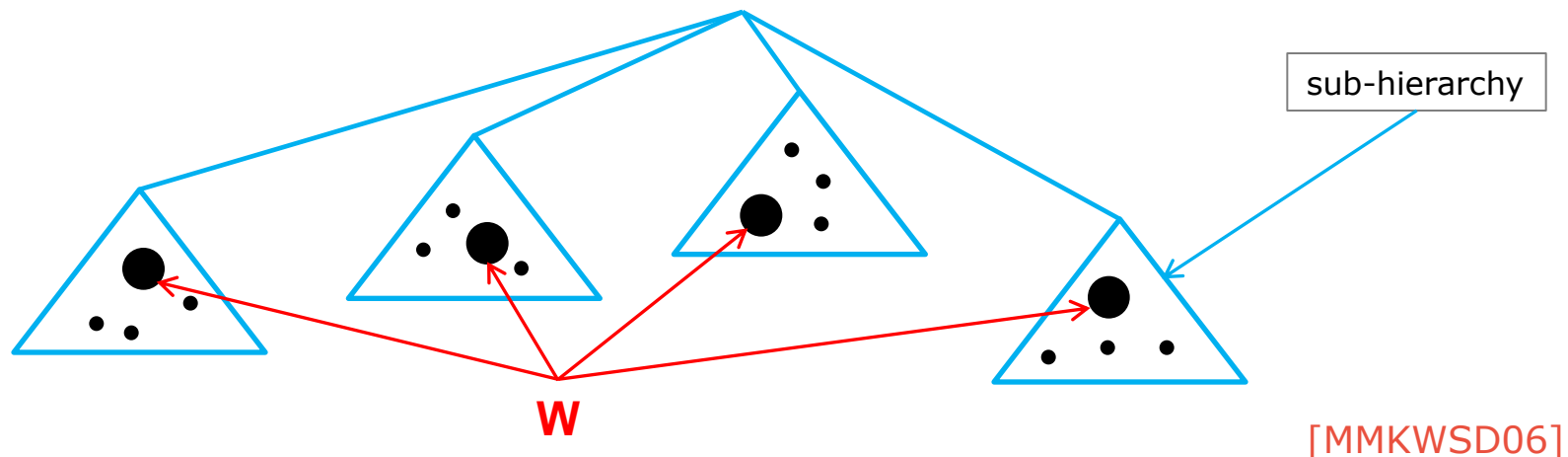
Conceptual Density

- 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)
- Approach uses also a **lexical database** (WordNet) for finding the conceptual distance
- Smaller conceptual distance leads to higher conceptual density!

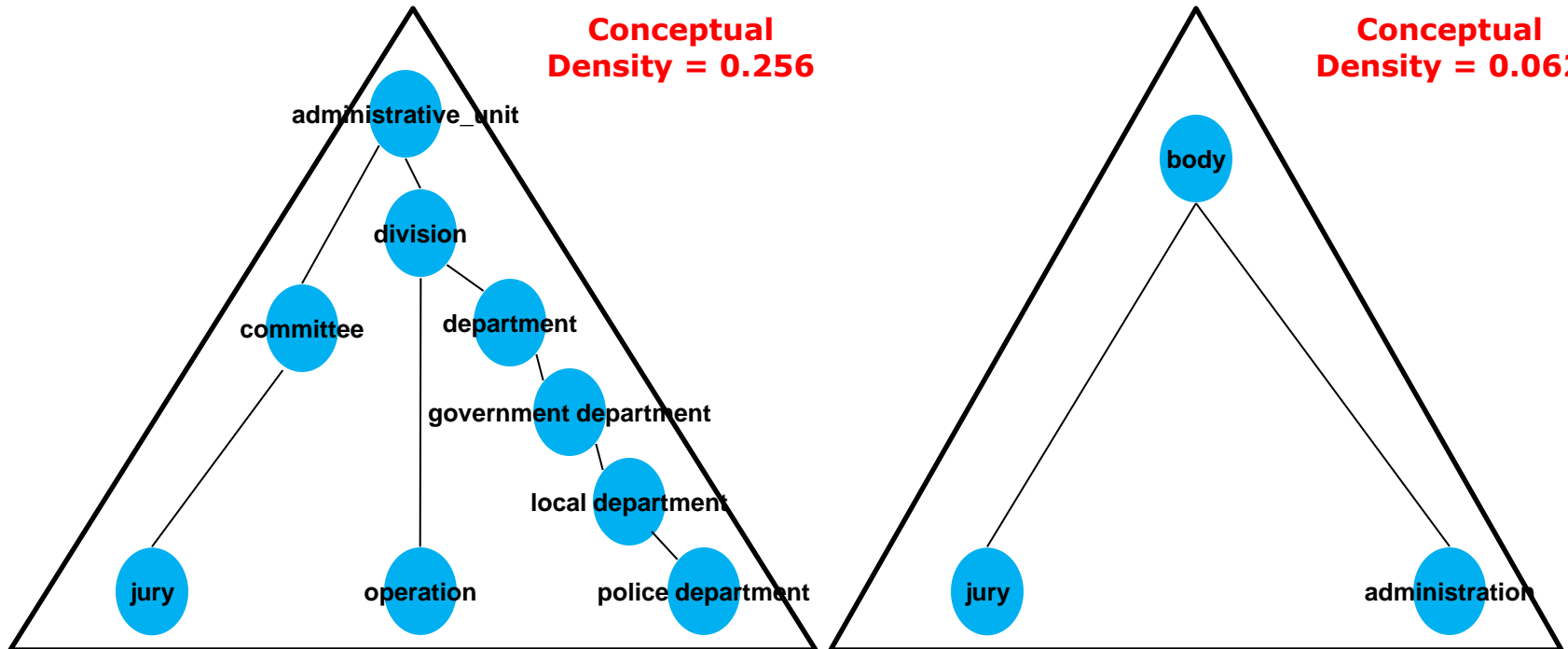
Conceptual Density

• Example

- The dots in the figure represent the senses of the word **W** 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
- Choose sense of **W** (contained in the sub-hierarchy) with the highest CD



Conceptual Density



*"The jury praised the administration and **operation** of Atlanta Police Department"*

- Step 1:** Make a lattice of the nouns in the context, their senses and hypernyms
- Step 2:** Compute conceptual density of resultant concepts (sub-hierarchies → triangles ☺)
- Step 3:** Select concept with highest "Conceptual Density"
- Step 4:** Select senses below the selected concept as the correct sense for the respective words

[MMKWSD06]

Conceptual Density

- How does the computation looks like?
- **Given**
 - concept **c** (at the top of a subhierarchy)
 - **nhyp** (average number of hyponyms per node)
 - **h** (height of the subhierarchy, respectively)
- The Conceptual Density CD for **c** is given by the formula:

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

- **Note**
 - Subhierarchy of **c** contains a number **m** (marks) of senses of the words to disambiguate
- The **0.20** tries to smooth the exponential i , as **m** ranges between 1 and the total number of senses in WordNet. It was found that the best performance was attained consistently when the parameter was near **0.20**

[ENGRWSD96]

Random Walk Algorithm

The church bells no longer rung on Sundays.

church

- 1: one of the groups of Christians who have their own beliefs and forms of worship
- 2: a place for public (especially Christian) worship
- 3: a service conducted in a church

bell

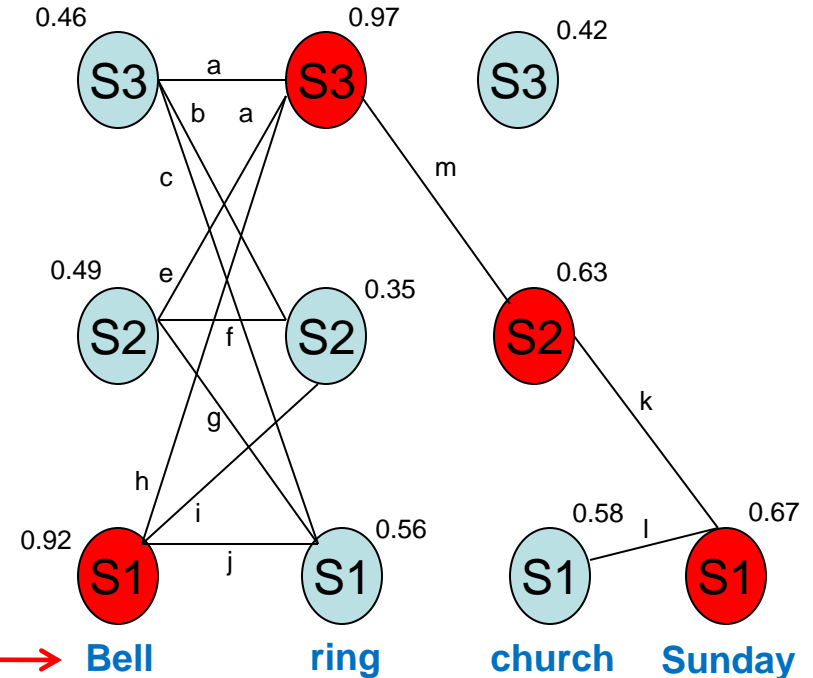
- 1: a hollow device made of metal that makes a ringing sound when struck
- 2: a push button at an outer door that gives a ringing or buzzing signal when pushed
- 3: the sound of a bell

ring

- 1: make a ringing sound
- 2: ring or echo with sound
- 3: make (bells) ring, often for the purposes of musical edification

Sunday

- 1: first day of the week; observed as a day of rest and worship by most Christians



Step 1: Add a vertex for each possible sense of each word in the Context-Window

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

[MMKWSD06]

KBD Approaches: Comparisons

Algorithm	Accuracy
Lesk's algorithm	50-60% on short samples of: <i>"Pride and Prejudice"</i> and some <i>"news stories"</i>
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

[MMKWSD06]

KBD Approaches: Conclusions



TECHNISCHE
UNIVERSITÄT
DARMSTADT

• Many drawbacks

- Dictionary definitions are generally **very small**
 - Manual tagging of word senses which is expensive
- Hard to obtain non-contentious definitions for words
 - In general, it's difficult for humans to agree on the division of senses of a word
- Proper nouns in context of an ambiguous word can act as strong disambiguators, **BUT** → Proper nouns are not present in the thesaurus!
- **Coverage:** "Michael Jordan" will not likely be in a thesaurus, **BUT** → is an excellent indicator for topic "sports"...
- **Domain-dependence:** In computer manuals: "mouse" will not be evidence for topic "mammal"...

[MMKWSD06]

Table of contents

✓ Motivation

✓ Introduction

✓ Variants of WSD

➤ **Approaches to WSD**

- ✓ Knowledge-Based Disambiguation
- Supervised Disambiguation
- Unsupervised Disambiguation

➤ Appendix

➤ References

Supervised Disambiguation: Task Definition

- Approach based on a labeled training set
- Supervised Disambiguation (SD) is known as a classification task
- The SD learning system has a training set of feature-encoded inputs and their appropriate sense label (category)
- **Resources**
 - Training corpora (hand-labeled with correct word senses)
- **Scope**
 - One target word per context (typically)

Bayesian Classification

- In 1992 Gale presented his approach for WSD:
 - The approach treats the context of occurrence as a “bag of words” without structure, but it integrates information from many words in the Context-Window
 - Recall the “Bayes Decision rule”:
 - Let $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \dots, \mathbf{s}_n$ be senses of an ambiguous word \mathbf{w}
 - Decide \mathbf{s}' if $P(\mathbf{s}' | c) > P(\mathbf{s}_k | c)$ for $\mathbf{s}_k \neq \mathbf{s}'$
 - Bayes decision rule is optimal because it minimizes the probability of error
 - Choose the class (or sense) with the highest conditional probability and hence the smallest error rate

Bayesian Classification



- In 1992 Gale presented his approach for WSD:
 - The approach treats the context of occurrence as a “bag of words” without structure, but it integrates information from many words in the Context-Window → **Note:** the Context-Window has a sequential order
- Recall the “Bayes Decision rule”:
 - Let $s_1, s_2, s_3, \dots, s_n$ be senses of an ambiguous word w
 - Decide s' if $P(s' | c) > P(s_k | c)$ for $s_k \neq s'$
- Bayes decision rule is optimal because it minimizes the probability of error
- Choose the class (or sense) with the highest conditional probability and hence the smallest error rate

Bayesian Classification

- Task: Assign an ambiguous word **w** to its sense **s'**, given a Context-Window **c**
- Select best sense **s'** from among the different senses:

$$\begin{aligned} s' &= \arg \max P(s_k | c) \\ &= \arg \max \frac{P(c | s_k)}{P(c)} P(s_k) \\ &= \arg \max P(c | s_k) P(s_k) \\ &= \arg \max [\log P(c | s_k) + \log P(s_k)] \end{aligned}$$

Baye's Rule

log

The "arg" stands for probability of argument **s_k**

Computationally it's pretty simpler to calculate logarithms...

Naïve Bayes

- An instance of a particular kind of Bayes classifier (“Naïve Bayes Assumption”)
- Well known in Machine Learning community for good performance across a range of tasks...

$$P(c | s_k) = P(\{v_j | v_j \text{ in } c\} | s_k) = \prod_{v_j \text{ in } c} P(v_j | s_k)$$

The v_j stands for “contextual features”

- Obtain resulting sense s' exactly like in the “classic” Bayesian Classification:

$$s' = \arg \max P(s_k | c)$$

- Consequences of this assumption:
 - 1) Feature order doesn't matter (bag of words model → repetition counts !)
 - 2) Every surrounding word v_j is independent of the other ones

Naïve Bayes: Conclusions

- Very efficient and simple to implement
- **Training**
 - One pass over the corpus to count feature-class co-occurrences
- **Classification**
 - Linear in the number of “active” features in the example
- **Note**
 - **Not the best** model but sometimes not **much worse** than more complex models
- Often a **useful quick solution** → good baseline for advanced models

Bayesian Classification

Pseudo-Algorithm



```
for all senses  $s_k$  of  $w$  do
  for all words  $v_j$  in the vocabulary do
     $P(v_j|s_k) = \frac{C(v_j, s_k)}{C(v_j)}$ 
  end
end
```

```
for all senses  $s_k$  of  $w$  do
   $P(s_k) = \frac{C(s_k)}{C(w)}$ 
end
```

```
for all senses  $s_k$  of  $w$  do
   $\text{score}(s_k) = \log P(s_k)$ 
  for all words  $v_j$  in the context window  $c$  do
     $\text{score}(s_k) = \text{score}(s_k) + \log P(v_j|s_k)$ 
  end
end
```

```
choose  $s' = \arg \max_{s_k} \text{score}(s_k)$ 
```

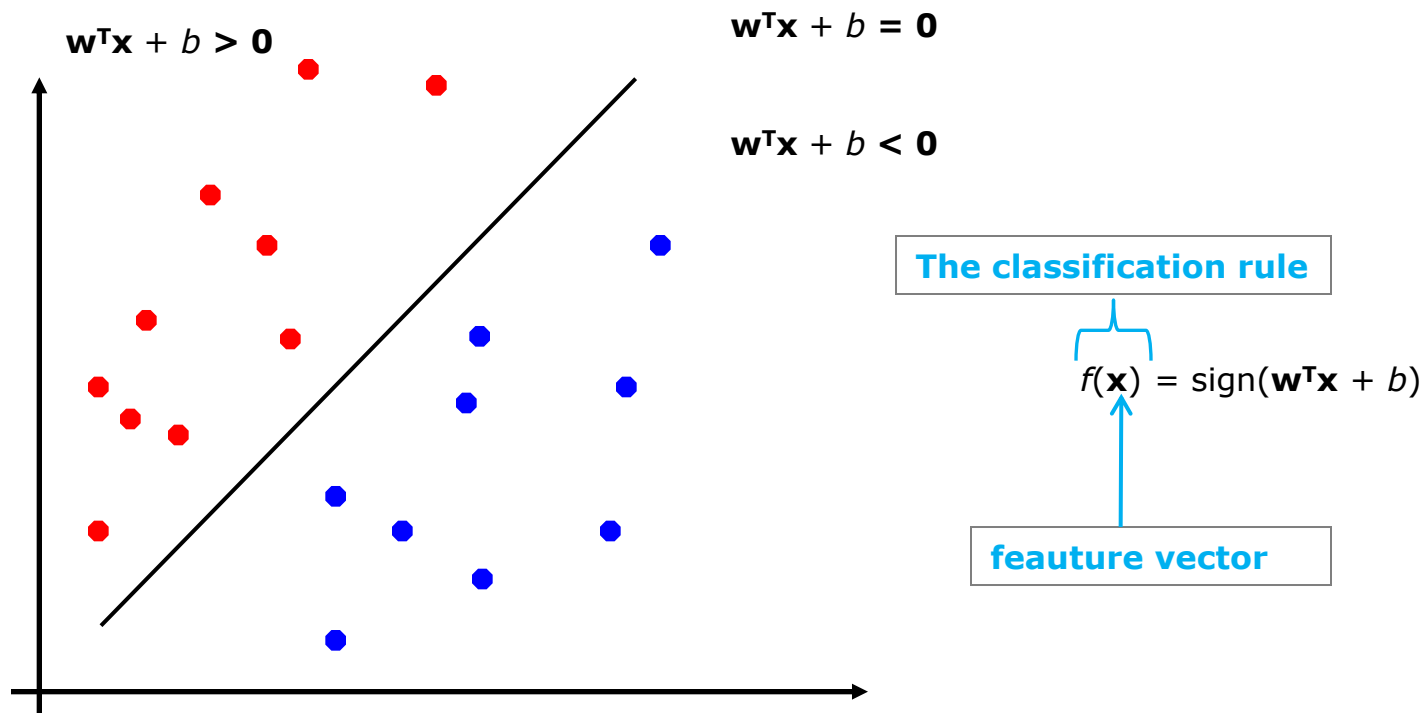
The training

The disambiguation process

The result

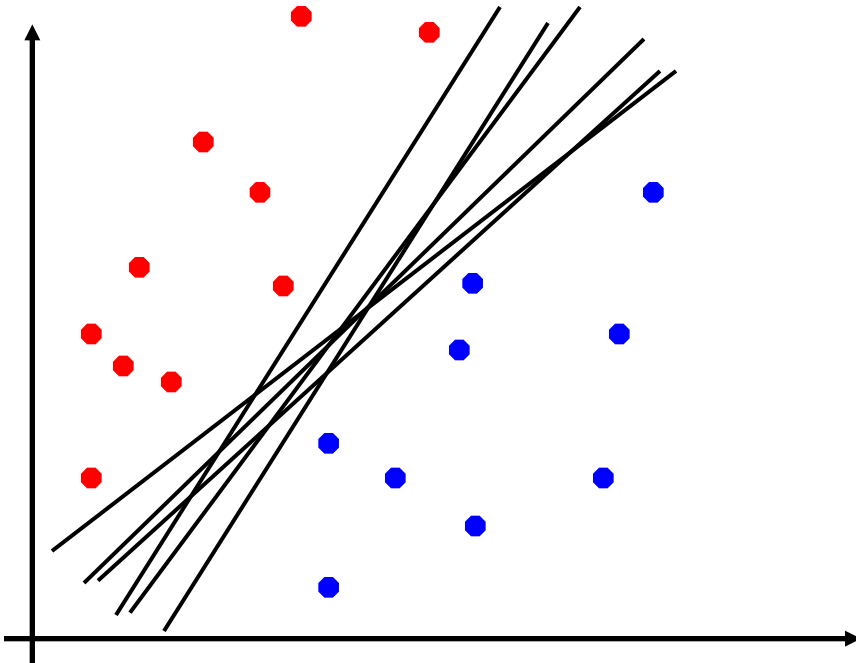
SVM: short intro...

- What are Support Vector Machines? → a very short introduction...
- Binary classification can be viewed as the task of separating classes in feature space:



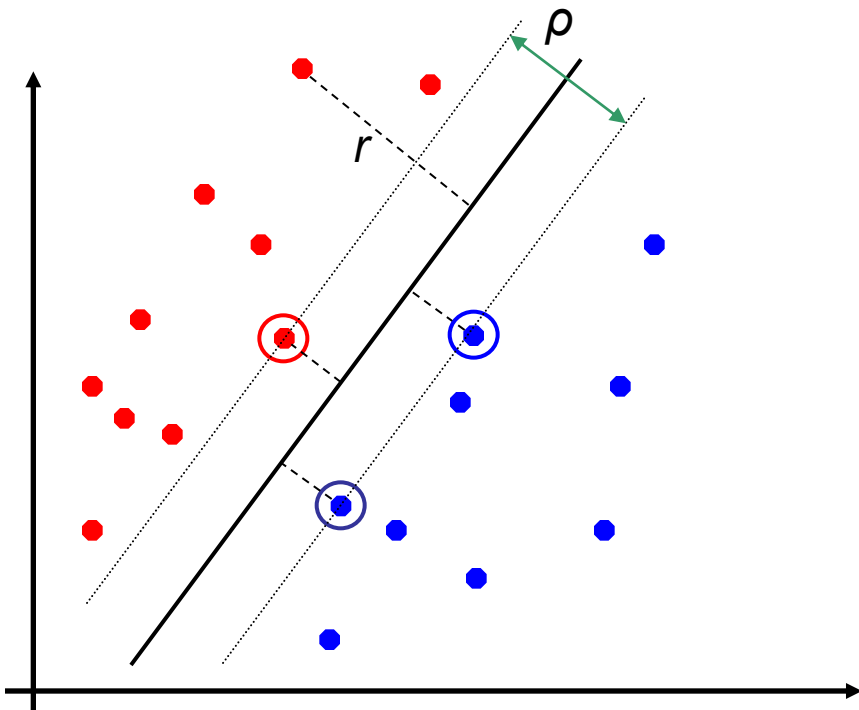
SVM: short intro...

- Which of the linear separators is optimal?



SVM: short intro...

- Distance from example \mathbf{x}_i to the separator is: $\longrightarrow r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are the so-called: **support vectors**
- Margin ρ of the separator is the distance between support vectors



SVM-based WSD

- So again, a **SVM** is a **binary classifier** which finds a hyperplane with the largest margin that separates training examples into 2 classes
- As SVM's are binary classifiers, a separate classifier is built for each sense of the word
- **Training Phase:** Using a tagged corpus, for every sense of the word a SVM is trained using the following features:
 - POS of w as well as POS of neighboring words
 - Local collocations
 - Co-occurrence vector
 - Features based on syntactic relations (e.g. headword, POS of headword, voice of head word etc.)
- **Testing Phase:** Given a test sentence, a test example is constructed using the above features and fed as input to each binary classifier
- The correct sense is selected based on the label returned by each classifier

k-nearest neighbor / Exemplar Based WSD

- A word-specific classifier, doesn't work for unknown words which do not appear in the corpus
- Uses several features (including morphological and noun-subject-verb pairs)

- **Step 1:** From each sense marked sentence containing the ambiguous word, a training example is constructed using:
 - POS of given word w as well as POS of neighboring words
 - Local collocations
 - Co-occurrence vector
 - Morphological features
 - Subject-verb syntactic dependencies
- **Step 2:** Given a test sentence containing the ambiguous word, a test example is similarly constructed
- **Step 3:** Compare test example to all training examples, select the k-closest training examples
- **Step 4:** Select sense which is most prevalent amongst these "k" examples is then selected as the correct sense

Supervised Disambiguation: Conclusions

- Supervised methods for WSD based on machine learning techniques are **undeniably effective** and they have obtained the **best results** to date

Approach	Average Precision	Average Recall	Corpus	Average Baseline Accuracy
Naïve Bayes	64.13%	Not reported	Senseval3 – All Words Task	60.90%
Exemplar Based disambiguation (k-NN)	68.6%	Not reported	WSJ6 containing 191 content words	63.7%
SVM	72.4%	72.4%	Senseval 3 – Lexical sample task (Used for disambiguation of 57 words)	55.2%

- However, **some questions** exists that **should be resolved** before stating that the supervised approach is a realistic way to construct accurate WSD-system

Table of contents



✓ Motivation

✓ Introduction

✓ Variants of WSD

➤ **Approaches to WSD**

- ✓ Knowledge-Based Disambiguation
- ✓ Supervised Disambiguation
- Unsupervised Disambiguation

➤ Appendix

➤ References

Unsupervised Disambiguation: Task Definition

- Approach identifies patterns in a large corpus (**not manually labeled**)
 - Besides this corpus **no other** external knowledge-base **sources** are allowed
 - Patterns are used to **divide data into clusters**
 - Each member of a cluster has more in common with other members of its own cluster than any other
- **Resources**
 - Large (raw) corpora, lexical database (without sense tags)
- **Scope**
 - Same as DS → One target word per context (typically)

Parallel Corpora Approach

- A word having multiple senses in one language will have distinct translations in another language
 - Based on the context in which it is used...
- Translations can thus be considered as contextual indicators of the sense of the word
- Pro's:
 - Many parallel corpora are available **for free** on the Web (see Appendix)
 - Manual **annotation** of sense tags is **not required!**

Parallel Corpora Approach

- However → text must be word aligned
(translations identified between the two languages)
- Given word aligned parallel text, sense distinctions can be discovered...

- **Example**

- Let the word be “interest”

In English:	'legal share' (acquire an interest) 'attention' (show interest)
In German:	Beteiligung erwerben Interesse zeigen

- Depending on where the translations of related words occur,
determine which sense applies

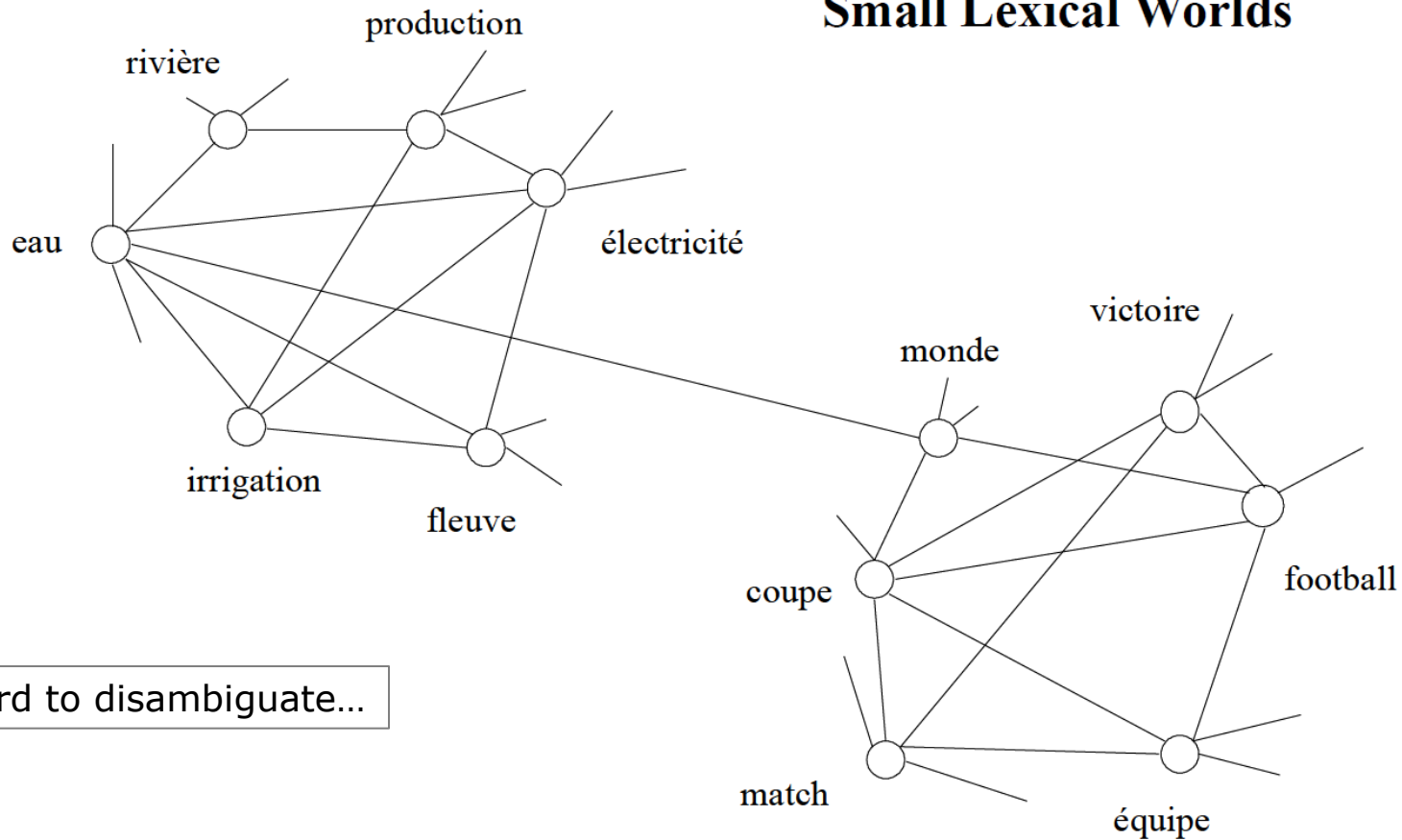
Parallel Corpora Approach

- Given a context \mathbf{c} in which a syntactic relation $R(\mathbf{w}, \mathbf{v})$ holds between \mathbf{w} and a context word \mathbf{v} :
- Score of sense \mathbf{s}_k is the number of contexts \mathbf{c}' in the second language such that:
 - $R(\mathbf{w}', \mathbf{v}') \in \mathbf{c}'$ where \mathbf{w}' is a translation of \mathbf{s}_k and \mathbf{v}' is a translation of \mathbf{v}
- Choose highest-scoring sense

- **Main idea**

- Instead of using "*dictionary defined senses*" extract: "*senses from the corpus*" itself
- These "*corpus senses*" correspond to **clusters** of similar contexts for a word...
- Build a co-occurrence graph **G**
 - "Small world" properties
 - Most nodes have few connections
 - Few are highly connected
 - Look for densely populated regions
 - Known as **High-Density Components**
- Map ambiguous instances to one of these regions

Small Lexical Worlds



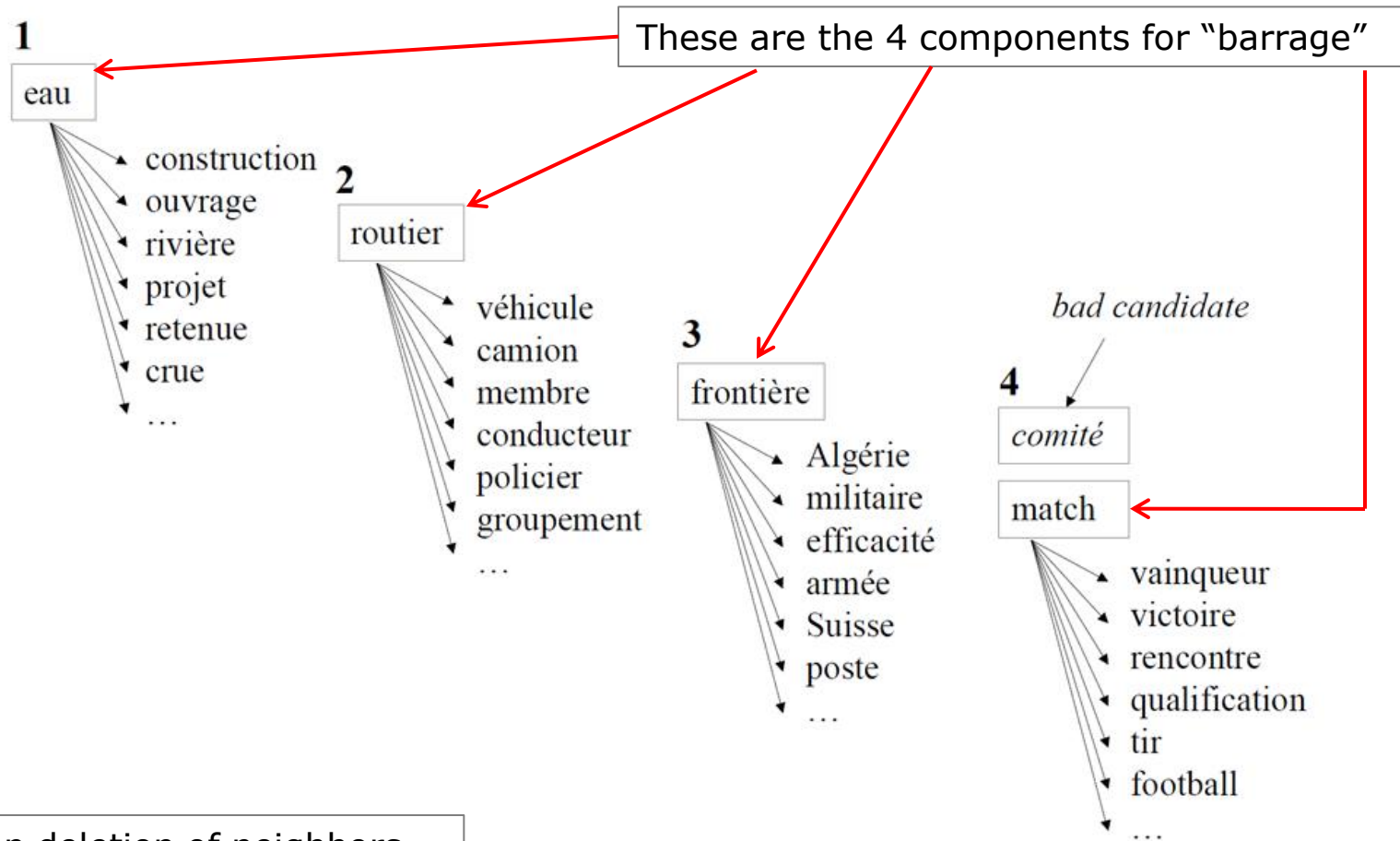
The word to disambiguate...

barrage = { dam, barrier, roadblock, play-off, police cordon, barricade }

- Nodes correspond to words
- Edges reflect the degree of semantic association between words
 - Model with conditional probabilities...
 - *Weight edges with: $w_{A,B} = 1 - \max[p(A|B), p(B|A)]$*

Note: the nodes A and B are already scored with a pagerank algo...

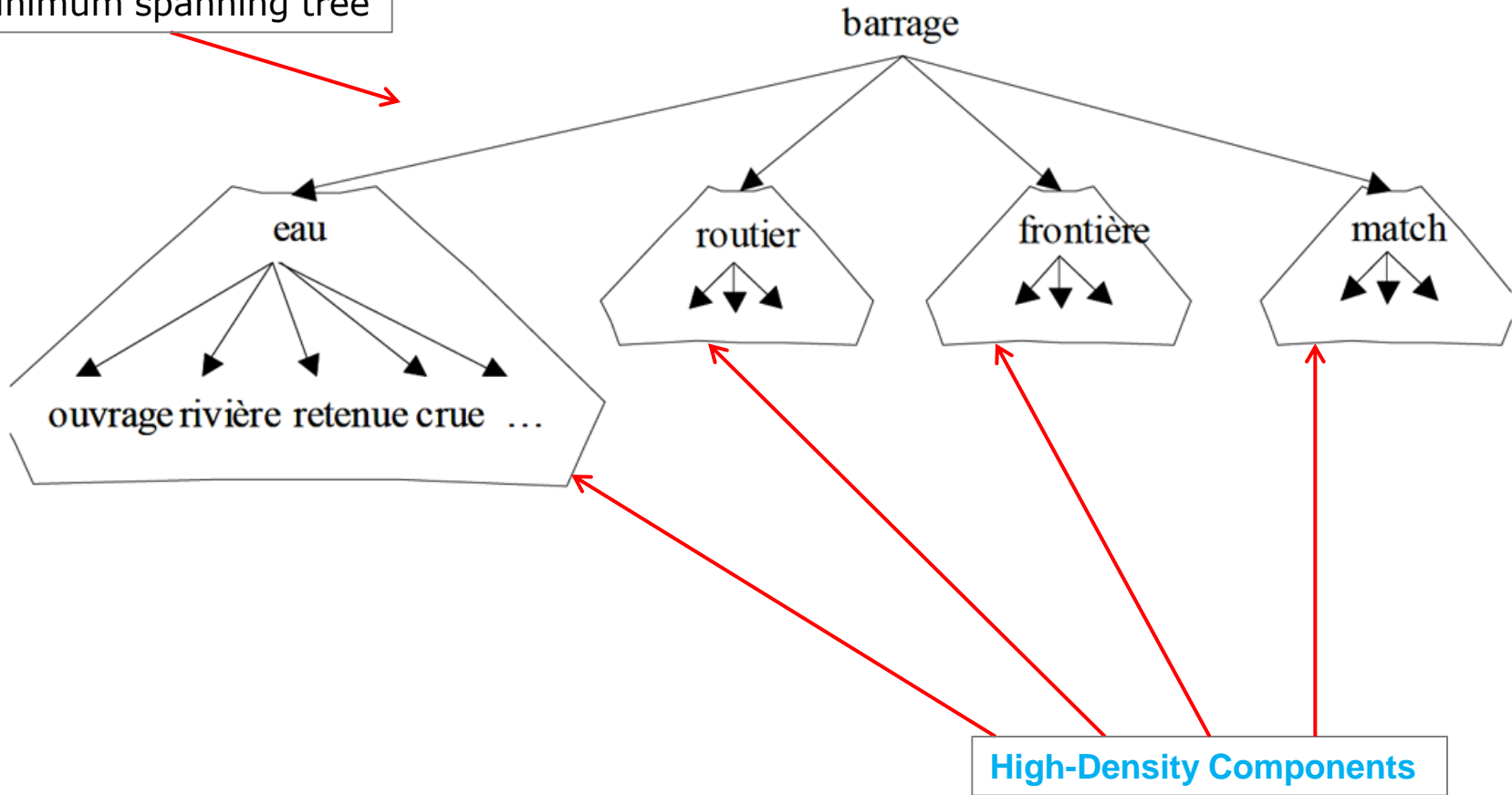
- Detect **High-Density Components**
 - Sort nodes by their degree
 - Take the top one (the so called “**root hub**”) and remove along with all its neighbors (hoping to eliminate the entire component)
 - Iterate until all the **High-Density Components** are found



Step-by-step deletion of neighbors...

HyperLex

Minimum spanning tree



- Finally, the disambiguation process...
- Each node inside the “MST-node” is assigned to a score vector with as many dimensions as there are components(!!)
- The score vector can be calculated as follows:

$$\mathbf{s}_i = \frac{1}{1 + d(h_i, v)} \text{ if } v \text{ belongs to component } i$$


$$\mathbf{s}_i = 0 \text{ otherwise}$$

$d(h_i, v)$ is the distance between root hub h_i and node v in the tree.

e.g. Pluei(rain) belongs to the component EAU(water) and $d(\text{eau}, \text{pluie}) = 0.82$, $\mathbf{s}_{\text{pluei}} = (0.55, 0, 0, 0)$

Step 1: For a given context, add the score vectors of all words in that context

Step 2: Select the component that receives the highest weight

- **Example**

“Le **barrage** recueille l’eau a la saison des plueis”

“The **dam** collects water during the rainy season”

	EAU	ROUTIER	FRONTIERE	MATCH
<i>S_{eau}</i>	1.00	0.00	0.00	0.00
<i>S_{saison}</i>	0.00	0.00	0.00	0.39
<i>S_{pluie}</i>	0.55	0.00	0.00	0.00
Total	1.55	0.00	0.00	0.39

 **EAU** is the winner in this case...

Unsupervised Disambiguation: Comparisons



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Approach	Precision	Average Recall	Corpus	Baseline
WSD using parallel corpora	SM: 62.4% CM: 67.2%	SM: 61.6% CM: 65.1%	Trained using a English Spanish parallel corpus Tested using Senseval 2 – All Words task (only nouns were considered)	Not reported
Hyperlex	97%	82% (words which were not tagged with Confidence > threshold were left untagged)	Tested on a set of 10 highly polysemous French words	73%

Unsupervised Disambiguation: Conclusions

- Combine advantages of supervised & knowledge based approaches
 - Just as supervised approaches: They **extract evidence from corpus**
 - Just as knowledge based approaches: They **do not need tagged corpus**
- Some drawbacks of Unsupervised Algorithms
 - Unsupervised methods **may not** discover clusters equivalent to the classes learned in supervised learning
 - The evaluation which is based on assuming that sense tags represent the 'true' cluster **is likely a bit harsh**

Questions

Still have any questions ?

Questions

Still have any questions ?

Sure ?

Questions

Still have any questions ?

Sure ?

Well then: Thanks for your attention 

Table of contents



✓ **Motivation**

✓ **Introduction**

✓ **Variants of WSD**

✓ **Approaches to WSD**

- ✓ Knowledge-Based Disambiguation
- ✓ Supervised Disambiguation
- ✓ Unsupervised Disambiguation

➤ **Appendix**

➤ **References**

Appendix

- On the following slides you can find a short list of valuable corpora, which have (and still) been used widely in the area of WSD for a long time
- Most of them are freely available (mind the restrictions: „fee“, „research activity“, etc.)

Appendix



Name / Title:	OED – Oxford English Dictionary & DIMAP
Description:	<p>The 2nd edition of the Oxford English Dictionary (OED) was released in October 2002 and contains 170,000 entries covering all varieties of English.</p> <p>Available in XML and in SGML, this dictionary includes phrases and idioms, semantic relations and subject tags corresponding to nearly 200 major domains.</p> <p>A computer-tractable version of the machine-readable OED was released by CL Research. This version comes along with the DIMAP software, which allows the user to develop computational</p> <p>lexicons by parsing and processing dictionary definitions.</p> <p>It has been used in a number of experiments.</p>
Availability:	DIMAP version: available for a fee
Web:	<p>http://www.oed.com</p> <p>http://www.clres.com</p>

Appendix



Name / Title:	Hector
Description:	<p>The Hector dictionary was the outcome of the Hector project (1992–1993) and was used as a sense inventory in Senseval-1. It was built by a joint team from Systems Research Centre of Digital Equipment Corporation, Palo Alto, and lexicographers from Oxford Univ. Press. The creation of this dictionary involved the analysis of a 17.3 million word corpus of 80-90s British English. Over 220,000 tokens and 1,400 dictionary entries were manually analyzed and semantically annotated. It was a pilot for the BNC (see below). Senseval-1 used it as the English sense inventory and testing corpus.</p>
Availability:	n/a
Web:	n/a

Appendix



Name / Title:	Roget's Thesaurus
Description:	<p>The older 1911 edition has been made freely available by Project Gutenberg. Although it lacks many new terms, it has been used to derive a number of knowledge bases, including Factotum. In a more recent edition, Roget's Thesaurus of English Words and Phrases contains over 250,000 word entries arranged in 6 classes and 990 categories. Jarmasz and Szpakowicz, at the University of Ottawa, developed a lexical knowledge base derived from this thesaurus. The conceptual structures extracted from the thesaurus are combined with some elements of WordNet.</p>
Availability:	1911 version and Factotum: freely available
Web:	<p>http://gutenberg.org/etext91/roget15a.txt http://www.cs.nmsu.edu/~tomohara/factotum-roles/node4.html</p>

Appendix



Name / Title:	WordNet
Description:	<p>The Princeton WordNet (WN), one of the lexical resources most used in NLP applications, is a large-scale lexical database for English developed by the Cognitive Science Laboratory at Princeton University. In its latest release (version 2.1), WN covers 155,327 words corresponding to 117,597 lexicalized concepts, including 4 syntactic categories: nouns, verbs, adjectives and adverbs. WN shares some characteristics with monolingual dictionaries. Its glosses and examples provided for word senses resemble dictionary definitions. However, WN is organized by semantic relations, providing a hierarchy and network of word relationships.</p> <p>WordNet has been used to construct or enrich a number of knowledge bases including Omega and the Multilingual Central Repository (see addresses below). The problems posed by the different sense numbering across versions can be overcome using sense mappings, which are freely available (see address below). It has been extensively used in WSD. WordNet was used as the sense inventory in English Senseval-2 and Senseval-3.</p>
Availability:	Free for research
Web:	http://wordnet.princeton.edu

Appendix



Name / Title:	EuroWordNet
Description:	<p>EuroWordNet (EWN) is a multilingual extension of the Princeton WN. The EWN database built in the original projects comprises WordNet-like databases for 8 European languages (English, Spanish, German, Dutch, Italian, French, Estonian and Czech) connected to each other at the concept level via the “Inter-Lingual Index”. It is available through ELDA (see below). Beyond the EWN projects, a number of WordNets have been developed following the same structural requirements, such as BalkaNet. The Global WordNet Association is currently endorsing the creation of WordNets in many other languages, and lists the availability information for each WordNet. EWN has been extensively used in WSD.</p>
Availability:	Depends on language
Web:	<p>http://www.globalwordnet.org</p> <p>http://www.ceid.upatras.gr/Balkanet</p>

Appendix



Name / Title:	FrameNet (and annotated examples)
Description:	<p>The FrameNet database contains information on lexical units and underlying conceptual structures. A description of a lexical item in FrameNet consists of a list of frames that underlie its meaning and syntactic realizations of the corresponding frame elements and their constellations in structures headed by the word. For each word sense a documented range of semantic and syntactic combinatory possibilities is provided. Hand-annotated examples are provided for each frame.</p> <p>At the time of printing FrameNet contained about 6,000 lexical units and 130,000 annotated sentences. The development of German, Japanese, and Spanish FrameNets has also been undertaken. Although widely used in semantic role disambiguation, it has had a very limited connection to WSD. Still, it has the potential in work to combine the disambiguation of semantic roles and senses.</p>
Availability:	Free for research
Web:	http://framenet.icsi.berkeley.edu/

Appendix

Name / Title:	The British National Corpus
Description:	<p>The British National Corpus (BNC) is the result of joint work of leading dictionary publishers (Oxford University Press, Longman, and Chambers-Larousse) and academic research centers (Oxford University, Lancaster University, and the British Library).</p> <p>The BNC has been built as a reasonably balanced corpus: for written sources, samples of 45,000 words have been taken from various parts of single-author texts. Shorter texts up to a maximum of 45,000 words, or multi-author texts such as magazines and newspapers, were included in full, avoiding over-representing idiosyncratic texts.</p>
Availability:	Available for a fee
Web:	http://www.natcorp.ox.ac.uk

Appendix



Name / Title:	The Wall Street Journal Corpus
Description:	<p>This corpus has been widely used in NLP. It is the base of the manually annotated DSO, Penn Treebank, and PropBank corpora.</p> <p>It is not directly available in raw form, but can be accessed through the Penn Treebank.</p>
Availability:	Available for a fee at LDC
Web:	http://www ldc upenn edu/Catalog/LDC2000T43.html

Appendix



Name / Title:	The Reuters News Corpus
Description:	<p>This corpus has been widely used in NLP, especially in document categorization. It is currently being used to develop a specialized hand-tagged corpus (see the domain specific Sussex corpus below).</p> <p>An earlier Reuters corpus (for information extraction research) is known as Reuters-21578.</p>
Availability:	Freely available
Web:	http://trec.nist.gov/data/reuters/reuters.html

Appendix

Name / Title:	Semcor
Description:	<p>Semcor, created at Princeton University by the same team who created WordNet, is the largest publicly available sense-tagged corpus. It is composed of documents extracted from the Brown Corpus that were tagged both syntactically and semantically.</p> <p>The POS tags were assigned by the Brill tagger, and the semantic tagging was done manually, using WordNet 1.6 senses. Semcor is composed of 352 texts. In 186 texts all of the open class words (192,639 nouns, verbs, adjectives, and adverbs) are annotated with POS, lemma, and WordNet synset, while in the remaining 166 texts only verbs (41,497 occurrences) are annotated with lemma and synset.</p> <p>Although the original Semcor was annotated with WordNet version 1.6, the annotations have been automatically mapped into newer versions (available from the same website below).</p>
Availability:	Freely available
Web:	http://www.cs.unt.edu/~rada/downloads.html

Table of contents



TECHNISCHE
UNIVERSITÄT
DARMSTADT

✓ **Motivation**

✓ **Introduction**

✓ **Variants of WSD**

✓ **Approaches to WSD**

- ✓ Knowledge-Based Disambiguation
- ✓ Supervised Disambiguation
- ✓ Unsupervised Disambiguation

✓ **Appendix**

➤ **References**

References



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- [Intro Logo]
<http://2jabaste.files.wordpress.com/2009/06/the-tower-of-babel-by-pieter-brueghel-the-elder.jpg>
- [AWS DHS98]
<http://www.aclweb.org/anthology-new/J/J98/J98-1004.pdf>
- [BMCWSD09]
<http://www.stanford.edu/class/cs224u/lec/224u.10.lec3.ppt>
- [SOTAWSD98]
<http://sites.univ-provence.fr/~veronis/pdf/1998wsd.pdf>
- [ULVWSD92]
<http://www.cse.unt.edu/~rada/papers/mihalcea.emnlp05a.pdf>
- [LDJJG92]
<http://acl.ldc.upenn.edu/C/C92/C92-1056.pdf>

References



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- [RDWSD99]
<http://www.aclweb.org/anthology-new/P/P99/P99-1020.pdf>
- [MMKWSD06]
<http://www.cse.iitb.ac.in/~nlp-ai/WSD.ppt>
- [MLESK04]
<http://www.lesk.com/mlesk>
- [ENGRWSD96]
<http://www.cs.helsinki.fi/u/huhmarni/opetus/tikik03/agirre96word.pdf>
- [SENSEVAL]
<http://www.senseval.org>