# Taal- en spraaktechnologie

Sophia Katrenko, also thanks to Navigli&Ponzetto

Utrecht University, the Netherlands

June 20, 2012

## Outline

1. **Today**
   - WSD: supervised
   - WSD: dictionary-based
   - WSD: minimally supervised
   - WSD: unsupervised
   - Concrete noun categorization task

# Focus

This part of the course focuses on

- meaning representation
- lexical semantics
- distributional similarity
- intro to machine learning
- word sense disambiguation
- information extraction

Today we discuss Chapter 19 (sections 1-5, 8 and 9; the rest of the chapter has already been covered!), and more precisely

1. Word sense disambiguation

Word sense disambiguation (WSD)

## WSD references

*WSD is the task of finding out which sense of a word is activated by its use in a particular context in an automatic way.*

- Navigli R. Word Sense Disambiguation: a Survey. ACM Computing Surveys, 41(2), ACM Press, 2009, pp. 1-69.
- Agirre E. and Edmonds P. Word Sense Disambiguation: Algorithms and Applications, New York, USA, Springer, 2006.
- Ide N. and Vronis J. Word Sense Disambiguation: The State of The Art. Computational Linguistics, 24(1), 1998, pp. 1-40.

# WSD approaches

- WSD has been typically seen as a supervised problem: classification given a fixed number of senses
- grouping words having the same sense together (in an unsupervised way, clustering) is called *word sense discrimination*
- is important for many NLP applications (e.g., machine translation)
- has been a popular topic for decades: have a look at Senseval-1 (1998) up to SemEval (2010)!

`http://www.senseval.org/`

## Senseval

Senseval has introduced the following tasks:

- *lexical sample*: only a selected number of words are tagged according to their senses. E.g., in Senseval-1, these were 35 words of different PoS, such as *accident, bother, bitter*.
- *all-words*: all content (open-class) words in text have to be annotated ⇒ more realistic, but also more difficult.
- *lexical substitution*: find an alternative substitute word or phrase for a target word in context (McCarthy and Navigli, 2007), whereby both synonyms need to be found and the context needs to be disambiguated.
- *cross-lingual disambiguation*: disambiguate a target word by labeling it with the appropriate translation in other languages (Lefever and Hoste, 2009)

## Senseval

Even though primarily for English, Senseval has expanded its language list to Basque, Chinese, Czech, Danish, Dutch, English, Estonian, Italian, Japanese, Korean, Spanish, Swedish.

### Example of lexical substitution

**Input:** "The packed screening of about 100 high-level press people loved the **film** as well"
**Output:** synonyms for the target movie (5); picture (3)

### Example of cross-lingual disambiguation

**Input:** "I'll buy a train or **coach** ticket"
**Output:** translations in other languages
NL: autobus (3); bus (3); busvervoer (1); toerbus (1);
IT: autobus (3); corriera (2); pullman (2); pulmino (1);
DE: Bus (3); Linienbus (2); Omnibus (2); Reisebus (2);

# WSD

So what is a word sense?

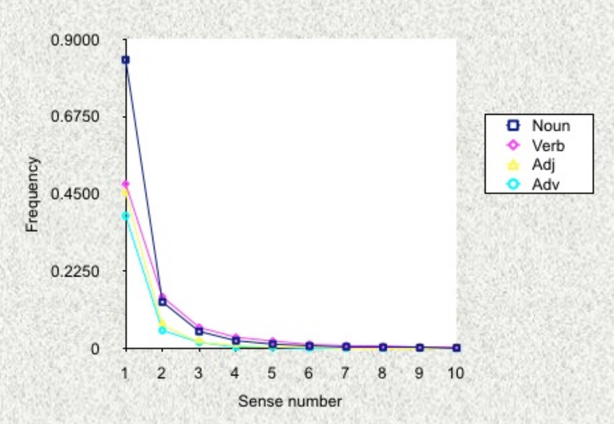
**R. Navigli:** A word sense is a commonly-accepted meaning of a word:

- We are fond of fruit such as *kiwi*$_{fruit}$ and banana.
- The *kiwi*$_{bird}$ is the national bird of New Zealand.

1. But is the number of senses per word really fixed?
2. What about the boundaries between senses - are they rigid?

# WSD

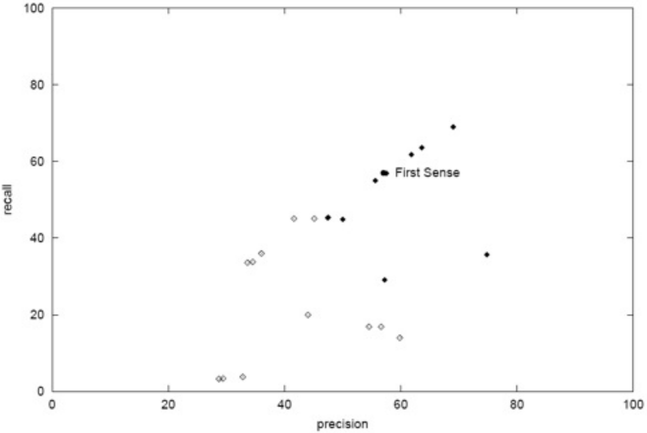
So why is it difficult? Consider the distribution of senses (*source* MacCartney; Navigli):



## WSD: Baselines

- take the most frequent sense (MFS) in the corpus (or the first WordNet sense)
- yields around 50-60% accuracy on lexical sample task w/ WordNet senses
- is a strong baseline (why?)

# WSD: Baselines



# WSD approaches

WSD approaches we consider today/next time:

- Supervised (Gale et al., 1992)
- Dictionary-based (Lesk, simplified Lesk, 1986)
- Minimally supervised (Yarowsky, 1995)
- Unsupervised (Mihalcea, 2009; Ponzetto and Navigli, 2010)
- Our own work on using qualia structures for WS induction (2008)

# WSD approaches: data

Supervised WSD needs training data! Then the steps are as follows

- extract features given training/test set
- train a ML method on the training set
- apply a model to test data

Sense-annotated corpora for *all-words task*

- **SemCor:** 200K words from Brown corpus w/ WordNet senses
- **SENSEVAL 3:** 2081 tagged content words

# WSD approaches: data

SemCor 3.0

```
<wf cmd=ignore pos=DT>The< /wf>
<wf cmd=done rdf=group pos=NNP lemma=group wnsn=1
lexsn=1:03:00:: pn=group>Fulton_County_Grand_Jury< /wf>
<wf cmd=done pos=VB lemma=say wnsn=1 lexsn=2:32:00::>
said< /wf>
<wf cmd=done pos=NN lemma=friday wnsn=1 lexsn=1:28:00::>
Friday< /wf>
<wf cmd=ignore pos=DT>an< /wf>
<wf cmd=done pos=NN lemma=investigation wnsn=1
lexsn=1:09:00::>investigation< /wf>
<wf cmd=ignore pos=IN>of< /wf>
```

## Supervised WSD

We have already talked about Naive Bayesian approach. How to use it for the WSD?

- we aim at selecting the most *probable* sense $\hat{s}$ of a given word $w$, described by a set of features $f_1 \dots f_n$, $\arg\max_{s \in S} P(s|f)$

$$\begin{aligned}\hat{s} = \arg\max_{s_i \in S} P(s|w) &= \arg\max_{s_i \in S} \frac{P(w|s_i)P(s_i)}{P(w)} \\ &= \arg\max_{s \in S} P(w|s_i)P(s_i)\end{aligned}$$

- we also naively assume all features to be independent:

$$\hat{s} = \arg\max_{s_i \in S} P(s_i) \prod_{j=1}^{n} P(f_j|s_i)$$

## Supervised WSD

- we have to calculate $P(s_i)$

$$P(s_i) = \frac{freq(s_i, w)}{freq(w)}$$

- we have to calculate $P(f_j|s_i)$

$$P(f_j|s_i) = \frac{freq(f_j, s_i)}{freq(s_i)}$$

- don't forget smoothing!

## Supervised WSD

Naive Bayes has been used in (Gale, Church,  Yarowsky, 92)

- to disambiguate 6 words (*duty, drug, land, language, position, sentence*) with 2 senses each
- using context of varying size
- achieved around 90%
- concluded that wide contexts are useful, as well as non-immediately surrounding words

# Dictionary-based WSD

Introduced in 1986 by Lesk and uses the following steps

- Retrieve all sense definitions of target word from a machine readable dictionary
- Compare with sense definitions of words in context
- Choose the sense with the most overlap

# Dictionary-based WSD

## Example (MacCartney)

pine
(a) a kind of **evergreen tree** with needle-shaped leaves
(b) to waste away through sorrow or illness

cone
(a) a solid body which narrows to a point
(b) something of this shape, whether solid or hollow
(c) fruit of certain **evergreen trees**

A simplified version of Lesk's method (Kilgarriff and Rosenzweig, 2000) works on the overlap of words, and not senses from the definitions

# Minimally supervised WSD

Introduced in 1995 by Yarowsky, based on two assumptions:

- **one sense per discourse**
  the sense of a target word can be determined by looking at the words nearby (e.g. *river* and *finance* for $\mathbf{bank}_n^1$ and $\mathbf{bank}_n^2$, respectively)
- **one sense per collocation**
  sense of a target word tends to be preserved consistently within a single discourse (e.g. a document in finance)

Paper: `http://acl.ldc.upenn.edu/P/P95/P95-1026.pdf`

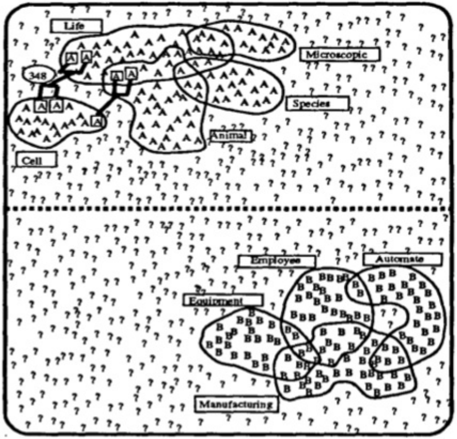
# Minimally supervised WSD

It's a bootstrapping method:

1. start from small seed set of manually annotated data $D_l$
2. learn decision-list classifier from $D_l$
3. use learned classifier to label unlabeled data $D_u$
4. move high-confidence examples to $D_l$
5. repeat from step 2

# Minimally supervised WSD

Source: Yarowsky (1995).



# Minimally supervised WSD

Decision list: a sequence of “if/else if/else” rules

- If $f_1$, then class 1
- Else if $f_2$, then class 2
- ...
- Else class n

Collocational features are identified from tagged data

- Word immediately to the left or right of target :
  The *window* **bars**$_n^3$ were broken.
- Pair of words to immediate left or right of target:
  The *world's largest* **bar**$_n^1$ is here in New York.

# Minimally supervised WSD

For all collocational features the log-likelihood ratio is computed, and they are ordered according to it:

$$\log \frac{P(sense_i | f_j)}{P(sense_k | f_j)} \tag{1}$$

What does the log-likelihood ratio really mean?

# Minimally supervised WSD

## Quote from Yarowsky (1995)

“New data are classified by using the single most predictive piece of disambiguating evidence that appears in the target context. By not combining probabilities, this decision-list approach avoids the problematic complex modeling of statistical dependencies encountered in other frameworks.”

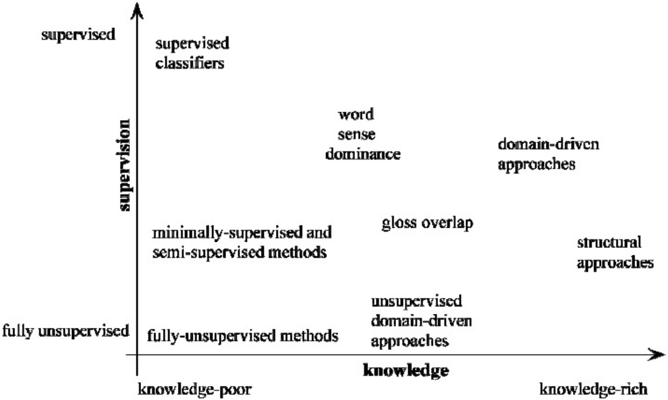
# Minimally supervised WSD

Initial decision list for plant (abbreviated), source: Yarowsky (1995)

| LogL | Collocation | Sense |
|---|---|---|
| 8.10 | plant life | A |
| 7.58 | manufacturing plant | B |
| 7.39 | life (within ±2-10 words) | A |
| 7.20 | manufacturing (in ±2-10 words) | B |
| 6.27 | animal (within ±2-10 words) | A |
| 4.70 | equipment (within ±2-10 words) | B |
| 4.39 | employee (within ±2-10 words) | B |
| 4.30 | assembly plant | B |
| 4.10 | plant closure | B |
| 3.52 | plant species | A |
| 3.48 | automate (within ±2-10 words) | B |
| 3.45 | microscopic plant | A |

# WSD methods: an overview

Source: Navigli and Ponzetto, 2010.



## Unsupervised WSD

- Most methods we have discussed so far focused on the classification, where the number of senses is fixed.
- Noun categorization has already shifted the focus to the unsupervised learning, whereby the learning itself was *unsupervised*, while the evaluation was done as for the *supervised* systems.
- We will move now more to the unsupervised learning, and discuss clustering (as a mechanism) in more detail.

# Unsupervised WSD

- The sense of a word can never be taken in isolation.
- The same sense of a word will have similar neighboring words.
- "You shall know a word by the company it keeps" (Firth, 1957).
- "For a large class of cases - though not for all - in which we employ the word meaning it can be defined thus: the meaning of a word is its use in the language." (Witgenschtein, "*Philosophical Investigations (1953)*").

## Unsupervised WSD

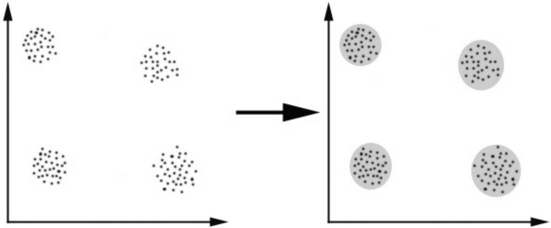
The unsupervised WSD relies on the observations above:

- take word occurrences in some (possibly predefined) contexts
- cluster them
- assign new words to one of the clusters

The noun categorization task followed only the first 2 steps (no assignment for new words).

# Clustering

- clustering is a type of unsupervised machine learning which aims at grouping similar objects into groups
- no apriori output(i.e., no labels)
- a cluster is a collection of objects which are similar (in some way)



# Clustering

Types of clustering

- Exclusive clustering (= a certain datum belongs to a definite cluster, no overlapping clusters)
- Overlapping clustering (= uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership)
- Hierarchical clustering (= explores the union between the two nearest clusters)
- Probabilistic clustering

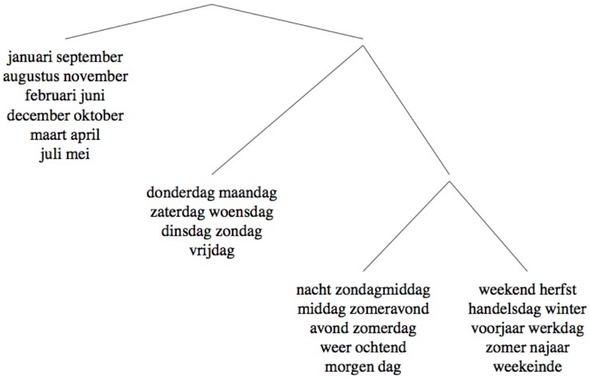
## Clustering

Hierarchical clustering is in turn of two types

- Bottom-up (agglomerative)
- Top-down (divisive)

# Clustering

Hierarchical clustering for Dutch text
Source: van de Cruys (2006)



januari september
augustus november
februari juni
december oktober
maart april
juli mei

donderdag maandag
zaterdag woensdag
dinsdag zondag
vrijdag

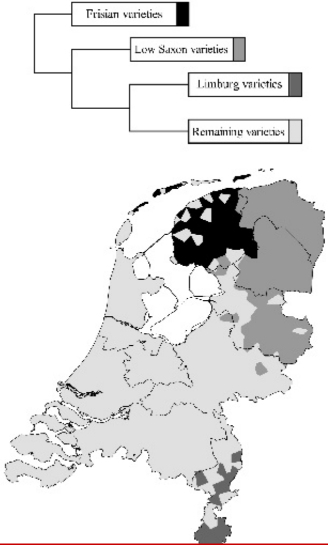
nacht zondagmiddag
middag zomeravond
avond zomerdag
weer ochtend
morgen dag

weekend herfst
handelsdag winter
voorjaar werkdag
zomer najaar
weekeinde

# Clustering

Hierarchical clustering for Dutch dialects
Source: Wieling and Nerbonne (2010)
The Goeman-Taeldeman-Van Reenen-project data

- 1876 phonetically transcribed items for 613 dialect varieties in the Netherlands and Flanders



## Clustering

Now ...

- clustering problems are hard (it is impossible to try all possible clustering solutions).
- clustering algorithms look at a small fraction of all possible partitions of the data.
- the portions of the search space that are considered depend on the kind of algorithm used.

# Clustering

What is a good clustering solution?

- the **intra-cluster** similarity is high, and the **inter-cluster** similarity is low.
- the quality of clusters depends on the definition and the representation of clusters.
- the quality of clustering depends on the similarity measure.

## Clustering

Agglomerative clustering works as follows:

1. Assign each object to a separate cluster.
2. Evaluate all pair-wise distances between clusters.
3. Construct a distance matrix using the distance values.
4. Look for the pair of clusters with the shortest distance.
5. Remove the pair from the matrix and merge them.
6. Evaluate all distances from this new cluster to all other clusters, and update the matrix.
7. Repeat until the distance matrix is reduced to a single element.

# Clustering

K-means algorithm

- Partitions $n$ samples (objects) into $k$ clusters.
- Each cluster $c$ is represented by its centroid:

$$\mu(c) = \frac{1}{|c|} \sum_{x \in c} x$$

- The algorithm converges to stable centroids of clusters (= minimizes the sum of the squared distances to the cluster centers)

$$E = \sum_{i=1}^{k} \sum_{x \in c_i} ||x - \mu_i||^2$$

# Clustering

K-means algorithm

1. Initialization: select k points into the space represented by the objects that are being clustered (seed points)
2. Assignment: assign each object to the cluster that has the closest centroid (mean)
3. Update: after all objects have been assigned, recalculate the positions of the k centroids (means)
4. Termination: go back to (2) until the centroids no longer move - i.e., there are no more new assignments

# Clustering

K-means algorithm

1. Initialization: select k points into the space represented by the objects that are being clustered (seed points)
2. Assignment: assign each object to the cluster that has the closest centroid (mean)
3. Update: after all objects have been assigned, recalculate the positions of the k centroids (means)
4. Termination: go back to (2) until the centroids no longer move - i.e., there are no more new assignments

# Clustering

K-means algorithm

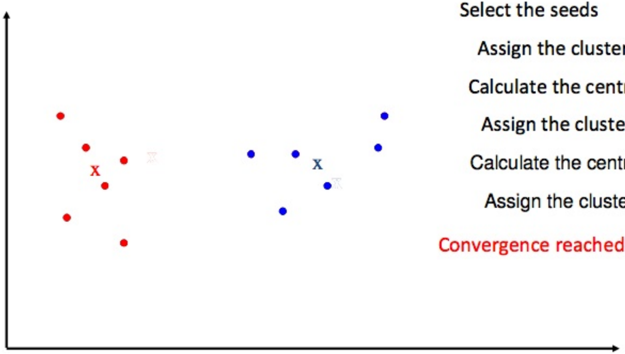
1. Initialization: select k points into the space represented by the objects that are being clustered (seed points)
2. Assignment: assign each object to the cluster that has the closest centroid (mean)
3. Update: after all objects have been assigned, recalculate the positions of the k centroids (means)
4. Termination: go back to (2) until the centroids no longer move - i.e., there are no more new assignments

# Clustering

K-means algorithm

1. Initialization: select k points into the space represented by the objects that are being clustered (seed points)
2. Assignment: assign each object to the cluster that has the closest centroid (mean)
3. Update: after all objects have been assigned, recalculate the positions of the k centroids (means)
4. Termination: go back to (2) until the centroids no longer move - i.e., there are no more new assignments

# Clustering



Select the seeds

Assign the clusters

Calculate the centroids

Assign the clusters

Calculate the centroids

Assign the clusters

Convergence reached!

K-means: limitations

- sensitive to initial seed points (it does not specify how to initialize the mean values - randomly)
- need to specify $k$, the number of clusters, in advance (how do we chose the value of $k$?)
- unable to handle noisy data and outliers
- unable to model the uncertainty in cluster assignment

K-means: limitations

- sensitive to initial seed points (it does not specify how to initialize the mean values - randomly)
- need to specify $k$, the number of clusters, in advance (how do we chose the value of $k$?)
- unable to handle noisy data and outliers
- unable to model the uncertainty in cluster assignment

K-means: limitations

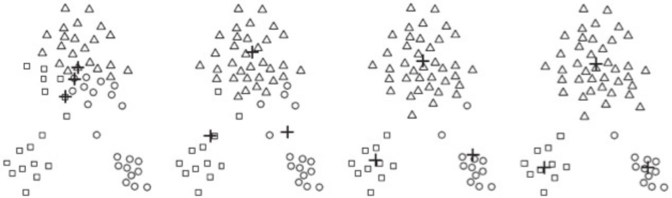
- sensitive to initial seed points (it does not specify how to initialize the mean values - randomly)
- need to specify $k$, the number of clusters, in advance (how do we chose the value of $k$?)
- unable to handle noisy data and outliers
- unable to model the uncertainty in cluster assignment

K-means: limitations

- sensitive to initial seed points (it does not specify how to initialize the mean values - randomly)
- need to specify $k$, the number of clusters, in advance (how do we chose the value of $k$?)
- unable to handle noisy data and outliers
- unable to model the uncertainty in cluster assignment

## Clustering

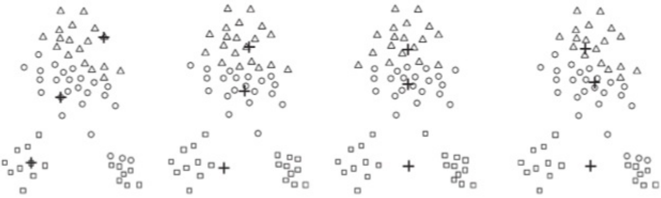
"Good" choice of seeds:



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

# Clustering

“Bad” choice of seeds:



(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

## Back to unsupervised WSD

1. Context clustering
   - Each occurrence of a target word in a corpus is represented as a context vector
   - Vectors are then clustered into groups, each identifying a sense of the target word
2. Word clustering
   - clustering words which are semantically similar and can thus convey a specific meaning
3. Co-occurrence graphs
   - apply graph algorithms to co-occurrence graph, i.e.graphs connect pairs of words which co-occur in a syntactic relation, in the same paragraph, or in a larger context

## Back to unsupervised WSD

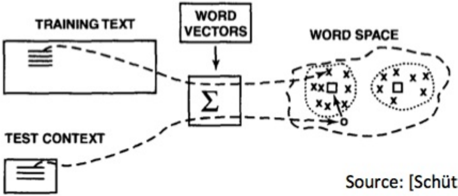
1. Context clustering
   - Each occurrence of a target word in a corpus is represented as a context vector
   - Vectors are then clustered into groups, each identifying a sense of the target word
2. Word clustering
   - clustering words which are semantically similar and can thus convey a specific meaning
3. Co-occurrence graphs
   - apply graph algorithms to co-occurrence graph, i.e.graphs connect pairs of words which co-occur in a syntactic relation, in the same paragraph, or in a larger context

## Back to unsupervised WSD

1. Context clustering
   - Each occurrence of a target word in a corpus is represented as a context vector
   - Vectors are then clustered into groups, each identifying a sense of the target word
2. Word clustering
   - clustering words which are semantically similar and can thus convey a specific meaning
3. Co-occurrence graphs
   - apply graph algorithms to co-occurrence graph, i.e.graphs connect pairs of words which co-occur in a syntactic relation, in the same paragraph, or in a larger context

# Context clustering

- A first proposal is based on the notion of word space (Schütze, 1992)
- A vector space whose dimensions are words
- The architecture proposed by Schütze (1998)



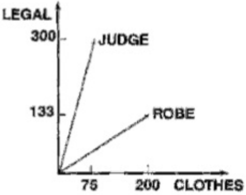
Source: [Schütze, 1998]

# Context clustering

So what is a word space?

- represent a word with a word vector
- a co-occurrence vector which counts the number of times a word co-occurs with other words

|  | **vector** |  |
|---|---|---|
| **dimension** | judge | robe |
| legal | 300 | 133 |
| clothes | 75 | 200 |



Source: [Schütze 1998]

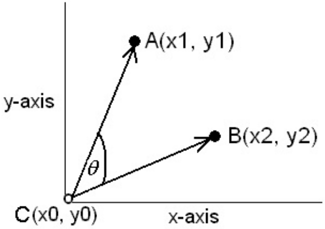
# Context clustering

Now, what can we do with all the vectors?

- compute the so-called dot-product (or inner product) $A \cdot B$
- measure their magnitudes $|A|$ and $|B|$ (Euclidean distance)

$$A \cdot B = x_1 * x_2 + y_1 * y_2 \tag{2}$$

$$|A| = d_{AC} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \tag{3}$$



# Context clustering

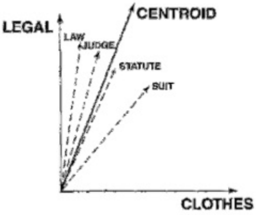
- Word vectors capture the “topical dimensions” of a word
- Given the word vector space, the similarity between two words **v** and **w** can be measured geometrically, e.g. by cosine similarity:

$$sim(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{vw}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{m} v_i w_i}{\sqrt{\sum_i^m v_i^2}\sqrt{\sum_i^m w_i^2}} \tag{4}$$

# Context clustering

Problem: the word vectors conflate senses of the word

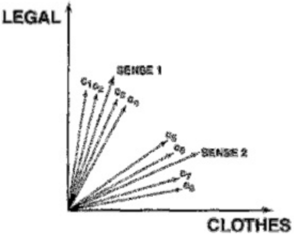
- we need to include information from the context
- context vector: the centroid (or sum) of the word vectors occurring in the context weighted according to their discriminating potential



Source: [Schütze 1998]

# Context clustering

- Finally: sense vectors are derived by clustering context vectors into a predefined number of clusters
- A sense is a group of similar contexts



Source: [Schütze 1998]

# Noun categorization task

1. Lexical representation/categorization in cognitive science
   - a lexical concept is represented by a set of features (Rapp & Caramazza, 1991; Gonnerman et. al., 1997)
   - lexical concepts are atomic representations and *"conceptual relations . . . can be captured by the sets of inferential relations drawn from elementary and complex concepts"* (Almeida, 1999), the thesis of conceptual atomism (Fodor, 1990)
2. Categorization in computational lingustics
   - word-space models (Sahlgren, 2006; Lenci, Baroni, and others)

## Data

- 44 concrete nouns have to be categorized in
  - 2 categories (natural kind and artifact)
  - 3 categories (vegetable, animal and artifact)
  - 6 categories (green, fruitTree, bird, groundAnimal, vehicle and tool) the entity derived from the origin.

## Data

- 44 concrete nouns have to be categorized in
  - 2 categories (natural kind and artifact)
  - 3 categories (vegetable, animal and artifact)
  - 6 categories (green, fruitTree, bird, groundAnimal, vehicle and tool) the entity derived from the origin.

## Data

- 44 concrete nouns have to be categorized in
  - 2 categories (natural kind and artifact)
  - 3 categories (vegetable, animal and artifact)
  - 6 categories (green, fruitTree, bird, groundAnimal, vehicle and tool) the entity derived from the origin.

## Data

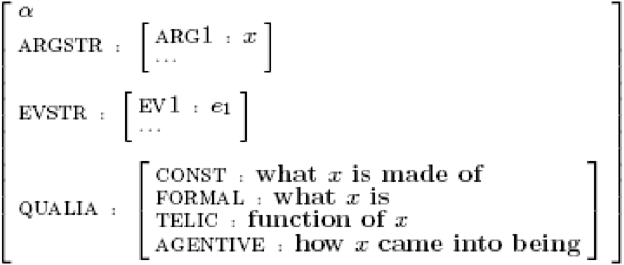
- 44 concrete nouns have to be categorized in
  - 2 categories (natural kind and artifact)
  - 3 categories (vegetable, animal and artifact)
  - 6 categories (green, fruitTree, bird, groundAnimal, vehicle and tool) the entity derived from the origin.

## Generative Lexicon Theory

Pustejovsky (1998) proposed a linguistically motivated approach to modelling categories. Semantic descriptions use 4 levels of linguistic representations such as

- argument structure ("specification of number and a type of logic arguments")
- event structure ("definition of the event type of an expression")
- qualia structure ("a structural differentiation of the predicative force for a lexical item")
- lexical inheritance structure ("identification of how a lexical structure is related to other structures in the type lattice")

## Generative Lexicon Theory (cont'd)



$$\begin{bmatrix} \alpha & & \\ \text{ARGSTR} & : & \begin{bmatrix} \text{ARG1} : x \\ \ldots \end{bmatrix} \\ \text{EVSTR} & : & \begin{bmatrix} \text{EV1} : e_1 \\ \ldots \end{bmatrix} \\ \text{QUALIA} & : & \begin{bmatrix} \text{CONST} : \text{what } x \text{ is made of} \\ \text{FORMAL} : \text{what } x \text{ is} \\ \text{TELIC} : \text{function of } x \\ \text{AGENTIVE} : \text{how } x \text{ came into being} \end{bmatrix} \end{bmatrix}$$

## Approach (cont'd)

How can we acquire qualia information? Some of the methods proposed in the past:

- Hearst, 1992 (hypernymy)
- Girju, 2007 (part-whole relations)
- Cimiano and Wenderoth, 2007
  - predefined patterns for all 4 roles
  - ranking results according to some measures
- Yamada et al., 2007
  - fully supervised
  - focuses on acquisition of telic information

# Approach (cont'd)

We make use of the patterns defined by Cimiano and Wenderoth, 2007

| role | pattern |
|---|---|
| formal | x_NN is_VBZ (a_DT\|the_DT) kind_NN of_IN |
| | x_NN is_VBZ |
| | x_NN and_CC other_JJ |
| | x_NN or_CC other_JJ |
| telic | purpose_NN of_IN (a_DT)* x_NN is_VBZ |
| | purpose_NN of_IN p_NNP is_VBZ |
| | (a_DT\|the_DT)* x_NN is_VBZ used_VVN to_TO |
| | p_NNP are_VBP used_VVN to_TO |

**Table:** Patterns: some examples

# Approach (cont'd)

| role | pattern |
|---|---|
| constitutive | (a_DT\|the_DT)* x_NN is_VBZ made_VVN (up_RP )*of_IN |
| | (a_DT\|the_DT)* x_NN comprises_VVZ |
| | (a_DT\|the_DT)* x_NN consists_VVZ of_IN |
| | p_NNP are_VBP made_VVN (up_RP )*of_IN |
| | p_NNP comprise_VVP |
| agentive | to_TO * a_DT new_JJ x_NN |
| | to_TO * a_DT complete_JJ x_NN |
| | to_TO * new_JJ p_NNP |
| | to_TO * complete_JJ p_NNP |
| | a_DT new_JJ x_NN has_VHZ been_VBN |
| | a_DT complete_JJ x_NN has_VHZ been_VBN |

**Table:** Patterns: some examples

# Approach (cont'd)

- Categorization procedure consists of the following steps
  - extraction of the passages containing candidates for the role fillers using patterns (Google, 50 snippets per pattern)
  - PoS tagging of all passages
  - actual extraction of the candidates for the role fillers using patterns
  - building a word-space model where rows correspond to the words provided by the organizers of the challenge and columns are the qualia elements for a selected role (clustering using CLUTO toolkit)

## Approach (cont'd)

- Categorization procedure consists of the following steps
  - extraction of the passages containing candidates for the role fillers using patterns (Google, 50 snippets per pattern)
  - PoS tagging of all passages
  - actual extraction of the candidates for the role fillers using patterns
  - building a word-space model where rows correspond to the words provided by the organizers of the challenge and columns are the qualia elements for a selected role (clustering using CLUTO toolkit)

# Approach (cont'd)

- Categorization procedure consists of the following steps
  - extraction of the passages containing candidates for the role fillers using patterns (Google, 50 snippets per pattern)
  - PoS tagging of all passages
  - actual extraction of the candidates for the role fillers using patterns
  - building a word-space model where rows correspond to the words provided by the organizers of the challenge and columns are the qualia elements for a selected role (clustering using CLUTO toolkit)

# Approach (cont'd)

- Categorization procedure consists of the following steps
  - extraction of the passages containing candidates for the role fillers using patterns (Google, 50 snippets per pattern)
  - PoS tagging of all passages
  - actual extraction of the candidates for the role fillers using patterns
  - building a word-space model where rows correspond to the words provided by the organizers of the challenge and columns are the qualia elements for a selected role (clustering using CLUTO toolkit)

# Evaluation

We use two evaluation measures (Zhao and Karypis, 2004):

- **Entropy**

$$p_{ij} = \frac{m_{ij}}{m_j}, H(c_j) = -\sum_{i=1}^{L} p_{ij} \log p_{ij} \tag{5}$$

$$H(C) = \sum_{j=1}^{K} \frac{m_j}{m} H(c_j) \tag{6}$$

- **Purity**

$$Pu(c_j) = \max_{i=1,\ldots,L} p_{ij}, Pu(C) = \sum_{j=1}^{K} \frac{m_j}{m} Pu(c_j) \tag{7}$$

where: $C = c_1, ..., c_K$ is the output clustering
$L$ is the set of classes ("gold" senses)
$m_{ij}$ is the number of words in cluster $j$ of class $i$ (a class is a gold cluster)
$m_j$ is the number of words in cluster $j$
$m$ is the overall number of words to cluster

# Evaluation

| clustering | entropy | purity |
|---|---|---|
| 2-way | 0.59 | 0.80 |
| 3-way | **0.00** | **1.00** |
| 6-way | 0.13 | 0.89 |
| 2-way$_{>1}$ | 0.70 | 0.77 |
| 3-way$_{>1}$ | 0.14 | 0.96 |
| 6-way$_{>1}$ | 0.23 | 0.82 |

**Table:** Performance using *formal* role only

# What are the most representative elements in the clusters?

The similarity between elements in a cluster is measured as follows:

$$z_l = \frac{s_j^l - \mu_l^l}{\delta_l^l} \tag{8}$$

$s_j^l$ stands for the average similarity between the object $j$ and the rest objects in the same cluster, $\mu_l^l$ is the average of $s_j^l$ values over all objects in the $l$th cluster, and $\delta_l^l$ is the standard deviation of the similarities.

## What are the most representative elements in the clusters?

- the core of the cluster respresenting tools is formed by *chisel* followed by *knife* and *scissors* as they have the largest internal *z*-score (the same cluster wrongly contains *rocket* but according to the internal *z*-score, it is an outlier (with the lowest *z*-score in the cluster))
- *bowl*, *cup*, *bottle* and *kettle* all have the lowest internal *z*-scores in the cluster of vehicles. The core of the cluster is formed by a *truck* and *motorcycle*

# Descriptive and discriminative features: 3-way clustering

| Cl | Features |
|---|---|
| VEG | fruit (41.3%), vegetables (28.3%), crop (14.6%), food (3.4%), plant (2.5%) |
| ANI | animal (43.3%), bird (23.0%), story (6.6%), pet (3.5%), waterfowl (2.4%) |
| ART | tool (31.0%), vehicle (15.3%), weapon (5.4%), instrument (4.4%), container (3.9%) |
| VEG | fruit (21.0%), vegetables (14.3%), animal (11.6%), crop (7.4%), tool (2.5%) |
| ANI | animal (22.1%), bird (11.7%), tool (10.1%), fruit (7.4%), vegetables (5.1%) |
| ART | tool (15.8%), animal (14.8%), bird (7.9%), vehicle (7.8%), fruit (6.8%) |

# Results: telic role

| seed | extractions |
|---|---|
| helicopter | to rescue |
| rocket | to propel |
| chisel | to cut, to chop, to clean |
| hammer | to hit |
| kettle | to boil, to prepare |
| bowl | to serve |
| pencil | to draw, to create |
| spoon | to serve |
| bottle | to store, to pack |

**Table:** Some extractions for the *telic* role

# Results: constitutive role

| **seed** | **extractions** |
|---|---|
| helicopter | a section, a body |
| rocket | a section, a part, a body |
| motorcycle | a frame, a part, a structure |
| truck | a frame, a segment, a program, a compartment |
| telephone | a tranceiver, a handset, a station |
| kettle | a pool, a cylinder |
| bowl | a corpus, a piece |
| pen | an ink, a component |
| spoon | a surface, a part |
| chisel | a blade |
| hammer | a handle, a head |
| bottle | a container, a component, a wall, a segment, a piece |

**Table:** Some extractions for the *constitutive* role

## Results per role

| role | clustering | entropy | purity | comments |
|---|---|---|---|---|
| formal | 6-way | 0.13 | 0.89 | all 44 words |
| agentive | 6-way | 0.54 | 0.61 | 43 words |
| constitutive | 6-way | 0.51 | 0.61 | 28 words |

**Table:** Performance using one role only

# Results: formal and agentive roles combined

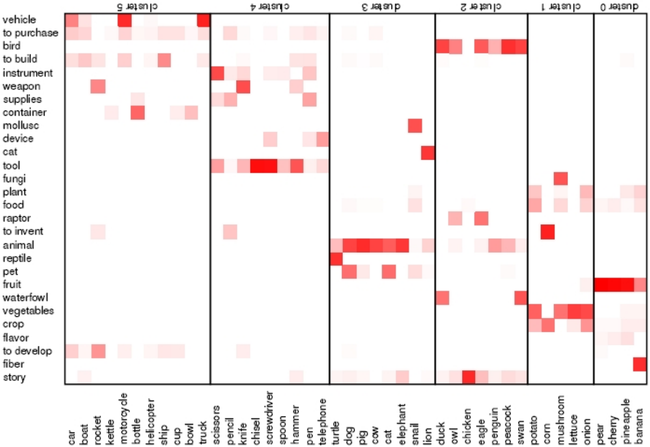


**Figure:** A combination of the formal and the agentive roles

# The best performance

The best results are obtained by combining formal role with the agentive one

| clustering | entropy | purity |
|---|---|---|
| 2-way | 0.59 | 0.80 |
| 3-way | 0.00 | 1.00 |
| 6-way | 0.09 | 0.91 |

**Table:** Performance using *formal* and *agentive* roles

Interestingly, the worst performance on 2-way clustering is achieved by combining formal and constitutive roles (entropy of 0.92, purity of 0.66)

# Error analysis

1. Errors due to the extraction procedure
   - incorrect PoS tagging/sentence boundary detection
   - patterns do not always provide correct extractions/features ("chicken and other stories")
2. Ambiguous words ("in fact, scottish gardens are starting to see many more butterflies including peacocks")
3. Features that do not suffice to discriminate among all categories

# Error analysis (cont'd)

1. 6-way clustering always fails to discriminate between tools and vehicles well. Containers (a bowl, a kettle, a cup, a bottle) are always placed in the cluster of vehicles (instead of tools). This is the only type of errors for the 6-way clustering.
2. In 2-way clustering, vegetables are usually not considered natural objects

## Conclusions

1. formal role is already sufficient for identification of vegetables, animals and artifacts (perfect clustering)
2. a combination of formal and agentive roles provides the best performance on 6-way clustering (in line with Pustejovsky, 2001)
3. no combination of roles accounts well for natural objects and artifacts