# Computational Semantics and Pragmatics

## Autumn 2011


Raquel Fernández

Institute for Logic, Language & Computation
University of Amsterdam

# Plan for Today

- Follow-up from next week:
  - ∗ Recap of supervised learning – Bos & Markert (2005) as example
  - ∗ Unsupervised learning of word senses
  - ∗ Brief discussion of HW#2
- Towards linguistic interaction:
  - ∗ Intro to Gricean pragmatics and conversational implicature
    [*we didn't have time to cover this part*]
- Planning for final projects

# Recap of Supervised Learning

Computational linguists often use machine learning (ML) to model the ability to classify linguistic objects (in a very broad sense) into classes or categories – the ability to categorise.

- In supervised learning, the learning algorithms are trained on data annotated with the classes we want to be able to predict.
- Inter-annotator agreement: an annotation is considered reliable if several annotators agree sufficiently on their classifications.
- Each item in the data set is characterised as a feature vector – we choose features that we think are predictive
- The data is split into training/development/test sets
- A learning algorithm is trained and tested on disjoint data sets
- Evaluation results are reported using standard measures (comparison to baseline and possibly upper bound)
- We analyse the contribution of each feature to understand underlying processes.

# Unsupervised Learning

- In unsupervised learning the training data is not annotated with the properties that the algorithm is intended to produce as output.
- The algorithm is provided with the data alone and must learn some interesting structure through identifying patterns
- Choice of supervised vs. unsupervised learning:
  - From a practical or engineering perspective, we are interested in balancing the degree of accuracy achieved in proportion to the cost of resources it requires.
  - From a theoretical perspective, we may consider these issues:
    - do these methods tell us anything about the learning mechanisms humans employ in acquiring knowledge of their language?
    - can they be models of human language acquisition?

# Unsupervised WSD

Why use unsupervised learning for WSD?

- It is expensive and difficult to build hand-labelled corpora.
- Hand-labelled senses may not be theoretically sound.
  Recall Kilgarriff's arguments:
  * defining a fix set of word senses may be impossible, and would at any rate be a domain-dependent task.
  * word senses should be reduced to abstractions over clusters of word usages.

In unsupervised WSD we do not start with a set of human-defined senses – the "senses" are created automatically from the instances of each word in the training set.

⇒ we can use a version of a DSM where we compute context vectors for each token of interest, i.e. for each usage, instead of computing vectors for *types* of target terms.

# Unsupervised WSD

Training: creating “senses” from usages

- For each token $t_w$ of word $w$ in a corpus, compute a context vector $\mathbf{c}_{t_w}$
- Use a clustering algorithm to cluster the vectors into groups or clusters; each cluster defines a sense of $w$
- Compute the vector centroid (the average or arithmetic mean) of each cluster; each centroid is a vector $\mathbf{s}_{w_i}$ representing that sense of $w$

Prediction: disambiguating a token $t_w$ of $w$ by assigning it a sense

- Compute a context vector $\mathbf{v}_{t_w}$ for $t_w$
- Retrieve all sense vectors for $w$
- Assign to $t_w$ the sense represented by the sense vector $\mathbf{s}_{w_i}$ that is closest to $\mathbf{v}_{t_w}$

This procedures requires a clustering algorithm and a distance metric to compare vectors.

# Clustering

Clustering is a general term referring to the task of classifying a set of objects into groups (clusters) so that the objects in the same cluster are more similar to each other than to those in other clusters.

Several clustering algorithms exist. Two common techniques are:

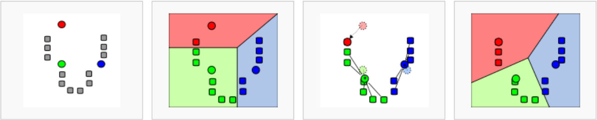
- $k$-means clustering
- Agglomerative hierarchical clustering

We will briefly review the basic steps involved in these two types of algorithms. For further details, you can consult these reference:

Manning & Schütze (1999) Foundations of Statistical Natural Language Processing, ch. 14: *Clustering*, MIT Press.
Jain, Murty & Flynn (1999) Data Clustering: A Review, *ACM Computing Surveys*, 31:264-323.

# $k$-means Clustering: Basics

1. assume a certain number $k$ of clusters;
2. select $k$ objects that are as distant as possible from each other; these are the starting centroids of the clusters;
3. assign each remaining object to the cluster whose centroid is the closest;
4. when all objects have been assigned, recalculate the positions of the $k$ centroids.
5. Repeat Steps 3 and 4 until the centroids are stable.

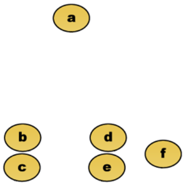


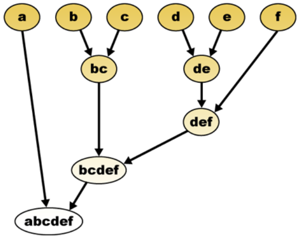
Picture from Wikipedia http://en.wikipedia.org/wiki/K-means_algorithm
There seems to be a mistake with red cluster, but good enough for illustration

# Agglomerative Clustering: Basics

1. assign each training instance to its own cluster
2. compute the distance between the clusters and merge the most similar pair of clusters
   * similarity between clusters can be computer by taking the shortest, the longest, or the average distance
3. repeat step 2 until either a specified number of clusters is reached or the clusters have some desired property.

By repeating step 2 until all items belong to the same cluster we end up with a tree that can be cut at the desired level of specificity.



# Evaluation of Unsupervised Predictions

In unsupervised learning we don't have a gold standard or ground truth against which we can compare the output of our system. Therefore evaluation can be tricky...

Some possibilities include:

- extrinsic evaluation: is the system's output positively evaluated by human judgements?
- end-to-end evaluation: does the output of the system improve the performance of a larger task? (e.g. does unsupervised WSD improve machine translation?)
- if an annotated corpus exists, we can also do an intrinsic evaluation (such as those in supervised learning). For instance, for WSD:
  - map each cluster (induced sense) to the predefined sense that in the training set has most word tokens overlapping with the cluster; or
  - for all pairs of usages of a word in the test set, test whether the system and the hand-labels consider the pairs to have the same sense or not.

# Recap

- Many linguistic tasks (incl. semantic/pragmatic ones) can be computationally approached as classification and learning tasks.
- Supervised learning can be used when the classifications to be made are well-motivated and we have annotated data
  - ∗ it is not only useful to get a task done but it can also shed light on the underlying processes that lead to particular categorisations
- Unsupervised learning may help us to avoid pre-conceptions that are not well motivated
  - ∗ it may be considered closer to human language learning
  - ∗ note that there is MUCH more to unsupervised learning than what we covered here!
- Evaluation is taken very seriously in Computational Linguistics
  - ∗ evaluation of human annotations
  - ∗ evaluation of systems' performance

## Towards Language Interaction

# Conversational Implicature

Background readings:

- Grice (1975) Logic and Conversation. In *Syntax and Semantics 3: Speech Acts*, 43-58. New York: Academic Press.
- Davis (2010) Implicature, The Stanford Encyclopaedia of Philosophy (Winter 2010 Edition), Edward N. Zalta (ed.).

# Computational Exploration of Implicature

Implicature is a very diverse and complex phenomenon. In the coming days we will look into some computational approaches that treat aspects related to implicature:

- Indirect answers to polar questions: can we use a data-driven approach to automatically predict whether the answer intended to convey "yes" or "no"? (dataset developed by Chris Potts)
- Generation of referring expressions: how can we automatically generate expressions that comply with the CP and the maxims and do not generate unintended implicatures?
- perhaps something else...

# What's Next

- After implicature, we will look into topics related to dialogue modelling. To get an overview, please read my draft chapter on "Dialogue" (see website) preferably *before* you meet me to discuss your project.
- On Thursday 3 Nov there is no class. Think about a topic for your final project and schedule an appointment with me to discuss it (preferred meeting times: Thursday 3/Friday 4 Nov 10-13h.)

# Final Projects: Timeline

- Next week (3/4 Nov): discuss with me possible topics
- 1 December: submit a 2-page abstract of your project
- 15 December: presentation of your work in progress
- 29 December: submit the final version of the paper

More details regarding the format and the length of the final paper as well as the grade breakdown will be announced shortly.

# Possible Types of Projects

The output of your project should be a paper related to the topics of course – computational semantics and pragmatics broadly.

Very important: Choose a topic you find interesting.

Here are a few ideas on possible types of projects (abstracting over particular topics):

- a quantitative corpus study of some interesting phenomenon
- a machine learning experiment using an existing corpus
- an analysis of data collected by youself
- an analysis and small extension of a paper from the literature
- an analysis of interesting connections between different approaches

Some options in the list above may seem unfeasible to you, but they may be perfectly possible — don't abandon an interesting idea before discussing it with me!