Machine Learning of natural language

Veronique Hoste

LT3 Language and Translation Technology Team
Ghent University

Machine Learning

Over the past decade ML techniques have become an essential tool for Natural Language Processing

Goals of this lecture:

- Cover the basics of ML
- Present a selection of widely used algorithms
- Illustrate ML in NLP tasks (WEKA)

Part I

MACHINE LEARNING

Background

Ever since computers were invented, we have wondered whether they might be made to learn.

Imagine

- computers learning from medical records which treatments are most effective for new diseases
- houses learning from experience to optimize energy costs based on the particular usage patterns of their occupants
- personal software assistants learning the evolving interests of their users in order to highlight especially relevant articles from the online morning newspaper

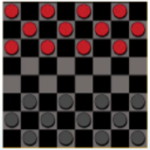
Machine learning

> The field of machine learning is concerned with the question of how to construct computer programs that automatically improve with experience.

Machine learning

A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*.

A checkers learning problem:



» Task *T*: learning checkers
» Performance measure *P*: percent of games won against opponents
» Training experience *E*: playing practice games against itself

What can we expect from machine learning?

We do not yet know how to make computers learn nearly as well as people learn.
But:

- ASR: algorithms based on machine learning outperform all other approaches
- Data mining: successful application of ML to discover knowledge from databases to detect credit card fraud detection, purchase patterns, etc.
- https://www.youtube.com/watch?v=WFR3lOm_xhE

Machine learning in NLP

text

To train modules

Lexical/morphological analysis

Tagging
Chunking
Parsing
Word sense disambiguation
Named entity recognition

Semantic role labeling

Coreference resolution

Discourse analysis

To construct LT applications

Spelling correction
Grammar checking

Information retrieval
Document classification
Information extraction
Summarization
Question answering

Sentiment detection
Authorship recognition

Machine translation

etc.

meaning

The move to machine learning

Acquisition

OLD: Construct a rule-based model about the domain
vs.
NEW: Induce a stochastic model from a corpus of examples

Processing

OLD: Use rule-based reasoning, deduction, on these models to solve new problems in the domain
vs.
NEW: Use statistical inference (generalization) from the stochastic model to solve new problems in the domain

Advantages

Deductive

- Linguistic knowledge and intuition can be used
- Precision

Inductive

- Fast development of model
- Good coverage
- Knowledge-poor
- Scalable / Applicable

Problems

Deductive

- Representation of sub/irregularity
- Cost and time of model development

Inductive

- Sparse data
- Estimation of relevance statistical events
- Understandability

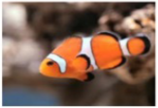
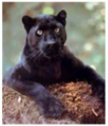
Induction in Machine Learning

= the inference from observations to given general rules.
In supervised machine learning, we have a set of data points or observations for which we know the desired output, class, target variable or outcome. In unsupervised learning, we are trying to identify the patterns inherent in the data that separate like observations in one way or another.

Supervised Machine Learning

- Very popular in NLP applications
- A supervised learner has access to a teacher which describes the function to be learned over a number of training examples, in practice an annotated data set (corpus, etc)
- Supervised learning methods are usually employed in learning of classification tasks
- Some notation:
  $D = d1, ..., d|D|$: a set of data instances.
  $C = c1, ..., c|C|$: a set of categories with respect to which instances will be classified.

Supervised Machine Learning

Data	Class label	Features
	Fish	Animal, jaws, black, orange, white
	panther	Animal, paws, black
	bird	Animal, feathers, black, orange, white

Supervised Machine Learning

Features

Animal, feathers, black

Class label

bird

Supervised Machine Learning

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Feature vector

Supervised Machine Learning

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy ›	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

New test instance: rainy mild normal false
Play tennis??

Learners

Once the data is converted into feature vector format, any supervised learning algorithm can be applied, e.g.

- Support Vector Machines
- Nearest Neighbor Classifiers
- Decision Trees
- Decision Lists
- Naïve Bayesian Classifiers
- Neural Networks
- Log Linear Models

The importance of algorithm bias

A learner that makes no a priori assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instance.(Mitchell)

Prior assumptions = inductive bias

the policy by which the learner generalizes beyond the observed training data, to infer the classification of new instances

E.g. decision tree learners favor compact decision trees

Learners

- Lazy leaners:
  Keep all training instances in memory during training;
  At classification time, there is the extrapolation of a class from the most similar items in memory to the new test item
  No abstraction is made from the data
- Eager leaners:
  The training material is compressed by extracting a limited number of rules;
  At classification time, these rules are applied to the test instances

No free lunch

No free lunch theorem (Wolpert and Macready 95)
= no inductive algorithm is universally better than any other

- In order to know which algoritmm fits a certain NLP task the best: experiment
- Usefulness of NLP (e.g. SemEval) competitions: comparison of different methodologies on the same data sets

Memory-based learning

- Background: performance in real-world tasks is based on remembering past events rather than creating rules or generalizations
- Lazy (vs. eager) : MBL keeps all training data in memory and only abstracts at classification time by extrapolating a class from the most similar items in memory to the new test item

Memory-based learning

1. memory-based learning component
   During learning, the learning component adds new training instances to the memory without any abstraction or restructuring
2. similarity-based performance component
   The classification of the most similar instance in memory is taken as classification for the new test instance

Memory-based learning

Given

(x_1, y_1) (x_2, y_2) (x_3, y_3) ... (x_n, y_n)

Example

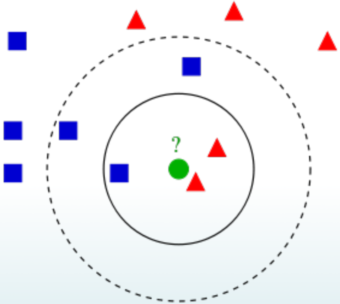
P-2	P-1	P+1	P+2	fish	check	river	interest	SENSE
S1	det	prep	det	Y	N	Y	N	SHORE
S2	det	verb	det	N	Y	N	Y	FINANCE

Task at classification time is to find the closest x_i for a new data point x_q.

Memory-based learning

Crucial components

- A distance metric
- The number of nearest neighbours to look at
- A strategy of how to extrapolate from the nearest neighbours



MBL: distance metric

- When presenting a new instance for classification to the MBL learner, it looks in memory to find all instances whose input attributes are similar to the newly presented test instance.
- Need for a *distance metric* that defines how far x_q and x_i are
- e.g. Overlap metric

$$\Delta(x_q, x_i) = \sum_{i=1}^{n} \delta(x_{qi}, x_{ii})$$

where:

$$\begin{aligned} \delta(x_{qi}, x_{ii}) &= 0 \; if \; x_{qi} = x_{ii} \\ \delta(x_{qi}, x_{ii}) &= 1 \; if \; x_{qi} \neq x_{ii} \end{aligned}$$

$=>$ number of matching and mismatching feature values in 2 instances (all feats. equally important)

MBL: distance metric (ctd.)

- Some features will be more informative for the prediction of the class label than others
- Some type of feature selection or feature weighting is required.
- e.g. Weighing each feature by **information gain**: a number expressing the average entropy reduction a feature represents when its value is known (Quinlan 93).

  Calculate the database information entropy:

  $$H(C) = -\sum_{c \in C} P(c) log_2 P(c)$$

  Calculate the information gain of feature *i*:

  $$w_i = H(C) - \sum_{v \in V_i} P(v) \times H(C|v)$$

MBL: the nearest neighbours

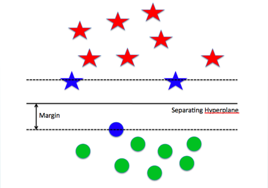
- Nearest neighbours: instances in memory which are near to the test item to be classified
- The classification of these nearest neighbours is used as classification for the new test instance
- Number of nearest neighbours is expressed by k
- In case of symbolic features: often nearest neighbours that have the same distance
  => k = number of nearest distances

MBL: model of how to extrapolate from the nearest neighbours

- **majority voting**: all nearest neighbours receive equal weight; most frequent class in the nearest neighbour set is taken as classification for the new test item
- **distance weighted voting:** link the choice of classification to the distance between the nearest neighbours and the new test item

Support vector machines

- Support Vector Machines (SVMs) learn a (linear) hyperplane separating 2 categories of training instances, in which the margin (distance between the hyperplane and the closest data point) is maximised.
- The category of new data points is predicted on the basis of the side of the hyperplane where the data points are located.
- Example: SVMlight (Joachims 1998)



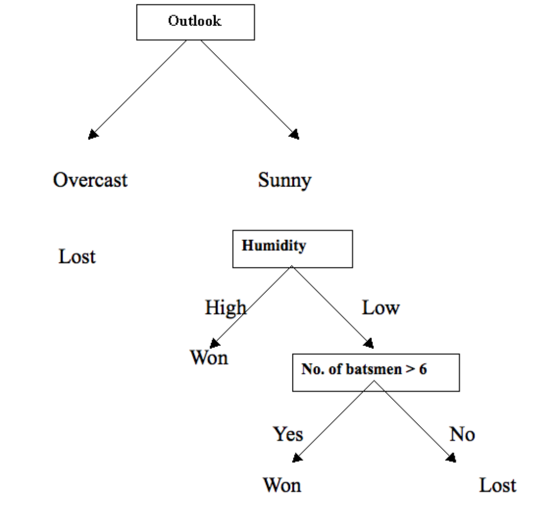
Decision Tree

- A decision tree is induced from a set of exampels. It is a special kind of tree structure which represents the alternatives and choices in the decision process.
- Important decision tree learners: ID3- and C4.5-(C5.0) algorithms
- Example: “Game won or lost?”

Decision Tree: “Game won or lost?”

Independent Variables			Dependent Variable
Outlook	**Humidity**	**Number of batsmen in team > 6**	**Final Outcome**
Sunny	High	Yes	Won
Overcast	High	No	Lost
Sunny	Low	No	Lost
Sunny	High	No	Won
Overcast	Low	Yes	Lost
Sunny	Low	Yes	Won
Sunny	Low	No	Lost
Sunny	High	No	Won
Sunny	Low	Yes	Won
Sunny	Low	Yes	Won

Decision Tree: “Game won or lost?”



Classifier ensembles

Intuition

- Combine the predictions of the individual classifiers by using a “voting” mechanism.
- An ideal ensemble consists of highly correct classifiers that disagree as much as possible.

Unsupervised Learning

- algorithm discovers on its own some kind of structure in the training data
- no (manually) labeled examples

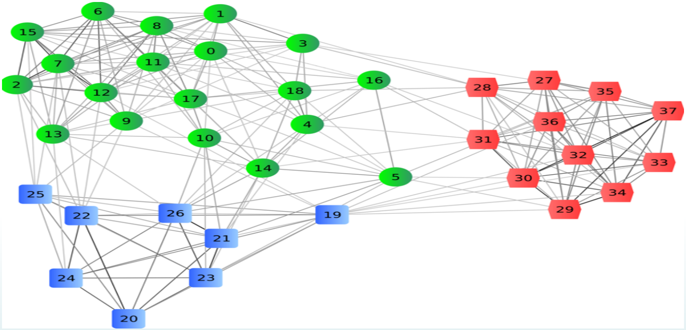
Clustering

Definition

Clustering is the organisation of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster (Jain et al. 99)

Clustering ctd.

- try to find a structure in labeled data
- group objects in homogeneous clusters or groups of which the members are similar to each other and dissimilar to the members of other clusters.

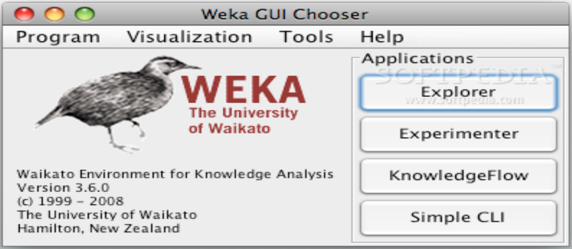


Part II

WEKA

Weka

- set of machine learning algorithms for data mining tasks
- tools for data preprocessing, classification, regression, clustering and visualisation



Weka

Reading material:

- http://www.cs.waikato.ac.nz/ml/weka/
- Manual:
  http://transact.dl.sourceforge.net/sourceforge/weka/WekaManual-3.6.0.pdf

Weka: arff files

- Weka file format: arff
- consists of a header which contains the list of features + a data section (feature values separated by a comma)
- Features:
  - Nominal: predefined list of values (e.g red, green, blue)
  - Numeric: number
  - String (between quotation marks)
  - Date
  - Relational

Weka: arff files

ARFF File Example

```
% This is a toy example, the UCI weather dataset.
% Any relation to real weather is purely coincidental

@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
```

Comment

Dataset name

Attributes

Target / Class variable

Data Values

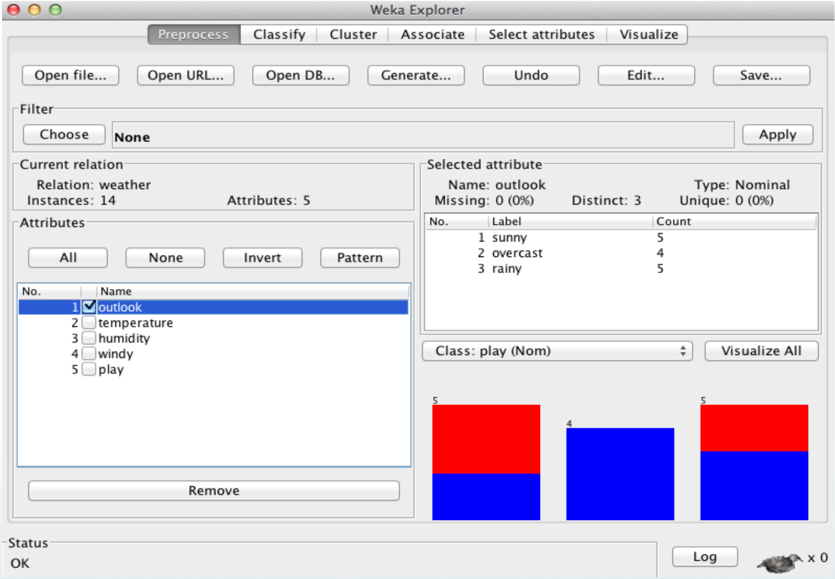
Weka Explorer

- Preprocess
- Classify
- Cluster
- Associate
- Select attributes
- Visualize

Explorer: Preprocess

- Load Data
- Preprocess Data
- Analyse Attributes

Weka Explorer: Preprocess



Classify

- Select test options (use training set, cross validation, etc.)
- Choose classifier
- Run classifier
- View results

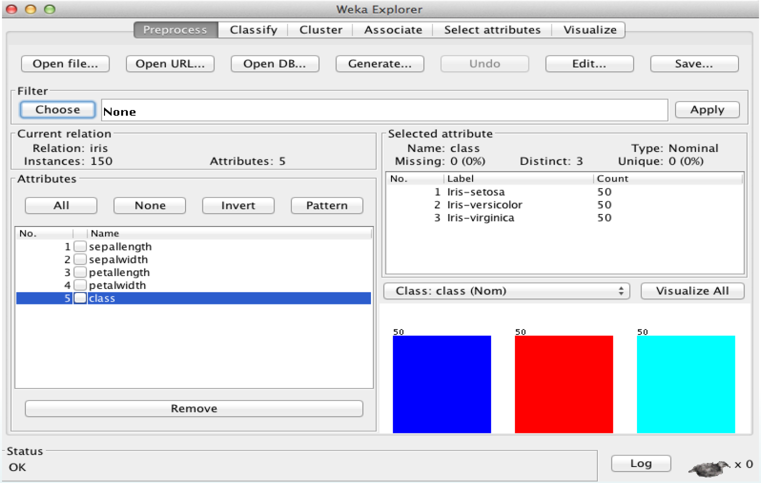
Clustering

- load “iris.arff” data set
- visualize attributes + classes
- cluster algorithm: simpleKMeans
- change the number of output clusters to 3 (click the clustering command)
- how many instances are incorrectly clustered?
- now try out 2 supervised learners: ZeroR and J48 and comment on the output

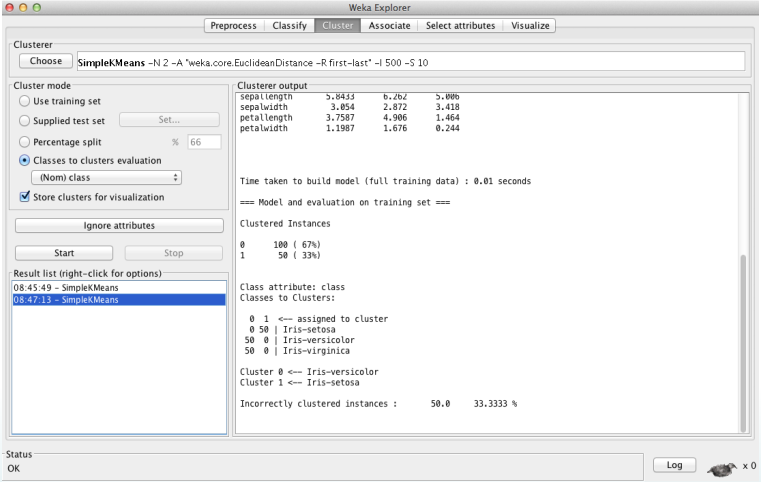
iris.arff

Features = length and width of sepal and petal
Classes = three northern American species of iris

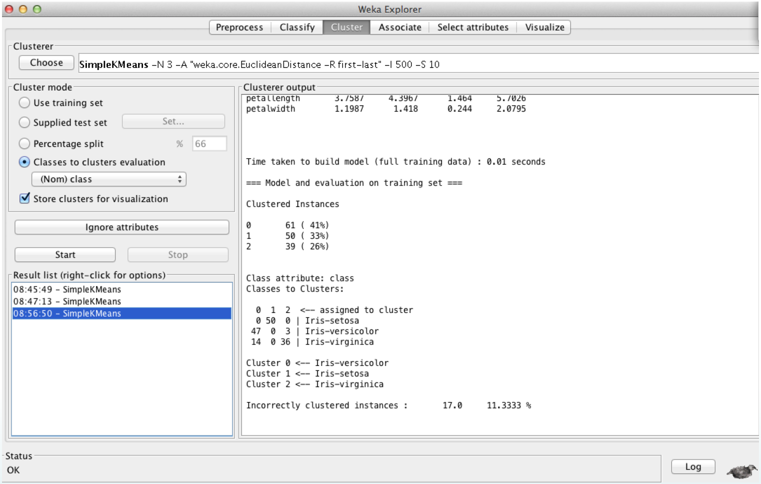
iris.arff


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Attributes
All
None
Invert
Pattern
No. Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Remove
Selected attribute
Name: class
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)
No. Label Count
1 Iris-setosa 50
2 Iris-versicolor 50
3 Iris-virginica 50
Class: class (Nom)
Visualize All
50
50
50
Status
OK
Log
x 0

iris.arff


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
08:45:49 - SimpleKMeans
08:47:13 - SimpleKMeans
Clusterer output
sepallength 5.8433 6.262 5.006
sepalwidth 3.054 2.872 3.418
petallength 3.7587 4.906 1.464
petalwidth 1.1987 1.676 0.244
Time taken to build model (full training data) : 0.01 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 100 (67%)
1 50 (33%)
Class attribute: class
Classes to Clusters:
0 1 <-- assigned to cluster
0 50 | Iris-setosa
50 0 | Iris-versicolor
50 0 | Iris-virginica
Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Incorrectly clustered instances : 50.0 33.3333 %
Status
OK
Log
x 0

iris.arff



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
08:45:49 - SimpleKMeans
08:47:13 - SimpleKMeans
08:56:50 - SimpleKMeans
Clusterer output
petallength 3.7587 4.3967 1.464 5.7026
petalwidth 1.1987 1.418 0.244 2.0795
Time taken to build model (full training data) : 0.01 seconds
=== Model and evaluation on training set ===
Clustered Instances
0 61 (41%)
1 50 (33%)
2 39 (26%)
Class attribute: class
Classes to Clusters:
0 1 2 <-- assigned to cluster
0 50 0 | Iris-setosa
47 0 3 | Iris-versicolor
14 0 36 | Iris-virginica
Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica
Incorrectly clustered instances : 17.0 11.3333 %
Status
OK
Log

iris.arff

```
J48 pruned tree
------------------

petalwidth <= 0.6: Iris-setosa (50.0)
petalwidth > 0.6
|   petalwidth <= 1.7
|   |   petallength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   petallength > 4.9
|   |   |   petalwidth <= 1.5: Iris-virginica (3.0)
|   |   |   petalwidth > 1.5: Iris-versicolor (3.0/1.0)
|   petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves  :      5

Size of the tree :       9
```

Evaluation

		True Class	
		pos	neg
Predicted Class	pos	*TP* *True Positives*	*FP* *False Positives*
	neg	*FN* *False Negatives*	*TN* *True Negatives*

Evaluation

- Accuracy

$$\frac{TP + TN}{TP + FP + FN + TN}$$

- Precision

$$\frac{TP}{TP + FP}$$

- Recall
(True Positive Rate)

$$\frac{TP}{TP + FN}$$

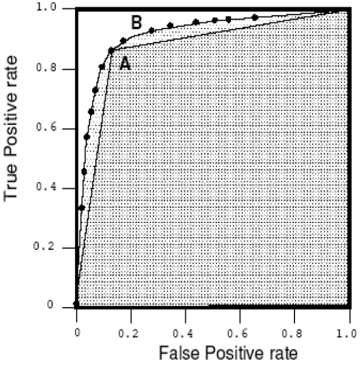
- F-score

$$\frac{2 \times precision \times recall}{precision + recall}$$

Evaluation

AUC (area under curve) in the ROC (receiver operator characteristics) space

- X-axis: False Positive Rate FP/(FP+TN)
- Y-axis: True Positive Rate TP/(TP+FN)
- Makes use of all cells in the matrix (unlike F-score)



Evaluation(ctd.): example

	Pred. as suicidal	Pred. as not suicidal
Suicidal	65 (TP)	12 (FN)
Not suicidal	42 (FP)	137 (TN)

Part III

Our machine learning problem: Word Sense Disambiguation

How to build a NLP system??

1. Step 1: collect data for your NLP problem to be solved
2. Step 2: annotate
3. Step 3: build feature vectors
4. Step 4: choose appropriate ML algorithm

How to build a NLP system??

1. Step 1: collect data for your NLP problem to be solved
2. Step 2: annotate
3. Step 3: build feature vectors
4. Step 4: choose appropriate ML algorithm

Collect data for your NLP problem to be solved

- depends on your research question
- how large?
- genre-balanced?

Collect data for your NLP problem to be solved

(Banko and Brill, 2001)

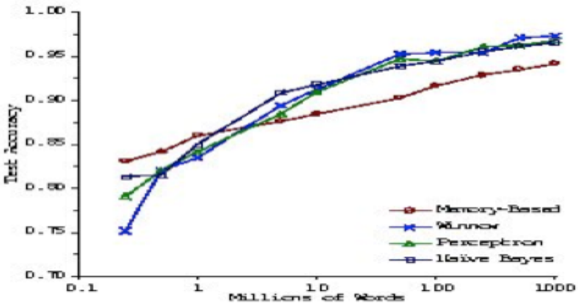


Figure 1. Learning Curves for Confusion Set Disambiguation

How to build a NLP system??

1. Step 1: collect data for your NLP problem to be solved
2. Step 2: annotate
3. Step 3: build feature vectors
4. Step 4: choose appropriate ML algorithm

Step 2: annotate

Background

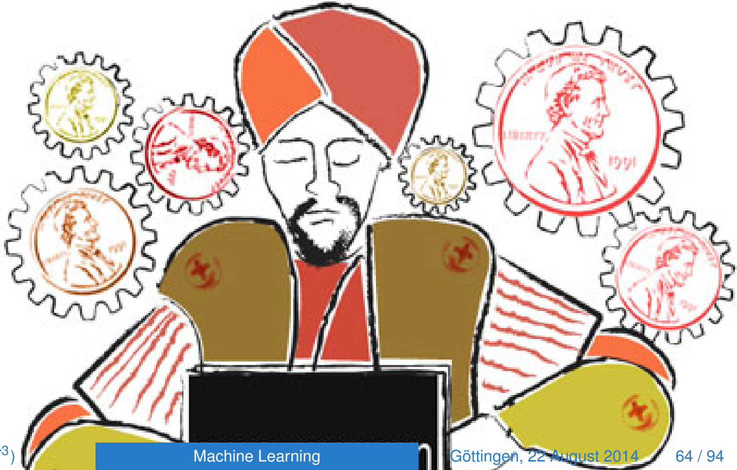
Modern (i.e., statistical) computational linguistics suffers from the need for more annotated data.
Creating 1M $>$ annotated corpora a major undertaking

2 possible ways to proceed:

- expert annotation: guidelines, expensive, slow, high quality
- crowdsourcing: no guidelines, cheap, fast, noise

Step 2: annotate

Efforts such as Wikipedia indicate that many Web surfers may be willing to participate in collective resource-producing efforts.
E.g.



Step 2: annotate

GAMES WITH A PURPOSE: THE ESP GAME (von AHN, 2006, 2008)

Step 2: annotate

www.phrasedetectives.org

Step 2: e.g. Sense tagged text

- SemCor [Miller et al. 1993]: 352 texts tagged with approximately 234,000 senses
- DSO corpus [Ng and Lee 1996]: 192,800 sense-tagged tokens of 191 words from the Brown and WSJ corpora
- Open Mind Word Expert corpus [Chklovski and Mihalcea 2002], 288 nouns semantically annotated by crowdsourcers
- de Senseval / Semeval data sets
  $\Rightarrow$ annotated with different version of WordNet
- others: MultiSemCor [Pianta et al. 2002], Interest corpus [Bruce and Wiebe 1994]

Step 2: e.g. Sense tagged text

Example

Bonnie and Clyde are two really famous criminals, I think they were bank/1 robbers.
My bank/1 charges too much for an overdraft.
I went to the bank/1 to deposit my check and get a new ATM card.
The University of Minnesota has an East and a West Bank/2 campus right on the Mississippi River.
My grandfather planted his pole in the bank/2 and got a great big catfish!
The bank/2 is pretty muddy, I can't walk there.

Step 2: e.g. Sense tagged text (ctd.)

<instance id="art.40001" docsrc="bnc_ACN_245">
<answer instance="art.40001" senseid="art%1:06:00::">
<context>
From their residency at the Fridge during the first summer of love, Halo used slide and film projectors to throw up a collage of op-art patterns, film loops of dancers like E-Boy and Wumni, and unique fractals derived from video feedback. "We're not aware of creating a visual identify for the house scene, because we're right in there. We see a dancer at a rave, film him later that week, and project him at the next rave."
Halo can be contacted on 071 738 3248.
<head>Art<head>you can dance to from the creative group called Halo
<context>
<instance>

Step 2: e.g. Sense tagged text (ctd.)

COACH

Step 2: e.g. Sense tagged text (ctd.)

Noun

- S: (n) **coach (coach%1:18:01::)**, manager (manager%1:18:01::), handler (handler%1:18:00::) ((sports) someone in charge of training an athlete or a team)
- S: (n) **coach (coach%1:18:00::)**, private instructor (private_instructor%1:18:00::), tutor (tutor%1:18:00::) (a person who gives private instruction (as in singing, acting, etc.))
- S: (n) passenger car (passenger_car%1:06:00::), **coach (coach%1:06:01::)**, carriage (carriage%1:06:01::) (a railcar where passengers ride)
- S: (n) **coach (coach%1:06:00::)**, four-in-hand (four-in-hand%1:06:00::), coach-and-four (coach-and-four%1:06:00::) (a carriage pulled by four horses with one driver)
- S: (n) bus (bus%1:06:00::), autobus (autobus%1:06:00::), **coach (coach%1:06:02::)**, charabanc (charabanc%1:06:00::), double-decker (double-decker%1:06:00::), jitney (jitney%1:06:00::), motorbus (motorbus%1:06:00::), motorcoach (motorcoach%1:06:00::), omnibus (omnibus%1:06:00::), passenger vehicle (passenger_vehicle%1:06:00::) (a vehicle carrying many passengers; used for public transport) *"he always rode the bus to work"*

How to build a NLP system??

1. Step 1: collect data for your NLP problem to be solved
2. Step 2: annotate
3. Step 3: build feature vectors
4. Step 4: choose appropriate ML algorithm

Step 3: Bag-of-words

Example

Bonnie and Clyde are two really famous criminals, I think they were bank/1 robbers.
My bank/1 charges too much for an overdraft.
I went to the bank/1 to deposit my check and get a new ATM card.
The University of Minnesota has an East and a West Bank/2 campus right on the Mississippi River.
My grandfather planted his pole in the bank/2 and got a great big catfish!
The bank/2 is pretty muddy, I can't walk there.

Step 3: Bag-of-words

FINANCIAL_BANK_BAG

a an and are ATM Bonnie card charges check Clyde criminals deposit famous for get I much My new overdraft really robbers the they think to too two went were

RIVER_BANK_BAG

a an and big campus cant catfish East got grandfather great has his I in is Minnesota Mississippi muddy My of on planted pole pretty right River The the there University walk West

Or: filter by using PoS tagging, terminology extraction, NER, etc.

Step 3: Bag-of-words

FINANCIAL_BANK_BAG

ATM Bonnie card charges check Clyde criminals deposit famous get new overdraft really robbers think went were

RIVER_BANK_BAG

big campus cant catfish East got grandfather great has is Minnesota Mississippi muddy planted pole pretty right River University walk West

Step 3: A simple supervised system

Given a sentence S containing the word “bank”:

For each word Wi in S
If Wi is in FINANCIAL_BANK_BAG then
Sense_1 = Sense_1 + 1;
If Wi is in RIVER_BANK_BAG then
Sense_2 = Sense_2 + 1;

If Sense_1 > Sense_2 then print “Financial”
else if Sense_2 > Sense_1 then print
“River”
else print “Can’t Decide”;

Step 3: More features

Preprocessing of the input tekst:

Word	Part-of-Speech	lemma	Chunk info
It	PRP	it	I-NP
is	VBZ	be	I-VP
no	RB	no	I-ADVP
longer	RBR	long	I-ADVP
the	DT	the	I-NP
locomotive	NN	locomotive	I-NP
it	PRP	it	B-NP
once	RB	once	I-ADVP
was	VBD	be	I-VP
,	,	,	O
it	PRP	it	I-NP
is	VBZ	be	I-VP
now	RB	now	I-ADVP
the	DT	the	I-NP
last	JJ	last	I-NP
coach	NN	coach	I-NP
in	IN	in	I-PP
the	DT	the	I-NP
train	NN	train	I-NP
.	.	.	O

Step 3: More features

The result of the preprocessing is converted into features (pieces of encoded information), e.g. :

- local features, refer to the local context of the target word (e.g. POS, lemma, etc.)
- topical features, refer to the general topic of a text (= broader context, e.g. sentence, paragraph, etc.), usually represented as a BoW
- syntactic features, syntactic information on the target word and other words in the sentence
- semantic features: e.g. domain information, etc.

Learner ingredients: a simple example

It is no longer the locomotive it once was, it is now the last coach in the train.

(a) Features focus word: coach coach NN I-NP
(b) Features context word -3: now now RB I-ADVP
(c) Features context word -2: the the DT I-NP
(d) Features context word -1: last last JJ I-NP
(e) Features context word +1: in in IN I-PP
(f) Features context word +2: the the DT I-NP
(g) Features context word +3: train train NN I-NP

Learner ingredients: another simple example (ctd.)

It is no longer the locomotive it once was, it is now the last coach in the train.
In our training corpus:

He always rode the coach to work.
A coach was used to transport children to or from school.
It was a passenger coach with an electric motor that draws power from overhead wires.
The coach was pulled by four horses.
Two coaches were in charge of training the athletes.

Learner ingredients: another simple example (ctd.)

It is no longer the locomotive it once was, it is now the last coach in the train.

In our training corpus: select informative keywords based on PoS

He always rode the coach to work.
A coach was used to transport children to or from school.
The coach was a passenger bus with an electric motor that draws power from overhead wires.
The coach was pulled by four horses.
Two coaches were in charge of training the athletes.
The locomotive of the train broke down, but one of the coaches was used to replace it.

Learner ingredients: another simple example (ctd.)

He always rode the coach to work.
A coach was used to transport children to or from school.
The coach was a passenger bus with an electric motor that draws power from overhead wires.
The coach was pulled by four horses.
Two coaches were in charge of training the athletes.
The locomotive of the train broke down, but one of the coaches was used to replace it.

In a real-world setting, BoW vectors are huge. But maybe some words are less informative and should be filtered out?

Learner ingredients: another simple example (ctd.)

Filtering out uninformative words

- Different possible metrics: TF_IDF, Log likelihood, etc. For multiword terms: mutual expectation, etc.
- Termhood (Drouin 2006): degree to which a linguistic unit is related to domain-specific context
- Unithood: degree of strength or stability of syntagmatic combinations or collocations,

Learner ingredients: another simple example (ctd.)

Log likelihood

	First Corpus	Second Corpus	Total
Frequency of word	a	b	a+b
Frequency of other words	c-a	d-b	c+d-a-b
Total	c	d	c+d

Learner ingredients: another simple example (ctd.)

Log likelihood

$$E_1 = c * (a + b)/(c + d)$$
$$E_2 = d * (a + b)/(c + d)$$

$$LL = 2 * ((a * log(\frac{a}{E_1})) + (b * log(\frac{b}{E_2})))$$

Learner ingredients: adding semantic information

very sparse lexical feature vectors: only a small amount of the lexical features has a positive value per instance

only exact lexical overlap is taken into account, no overlap between synonyms

a possible solution: LSA

Latent Semantic Analysis (Landauer and Dumais 1997, Landauer, Foltz and Laham 1998) starts from the distributional hypothesis that words that are close in meaning will occur in similar contexts.

Learner ingredients: adding semantic information

- hypothesis: words that are close in meaning occur in similar contexts
- LSA: uses Singular Value Decomposition (SVD), a mathematical technique, to:
  - reduce the dimensionality of the feature vectors by keeping the most relevant information → non-informative features are removed
  - capture *latent* and higher order associations between terms → capable of finding hidden associations between synonyms of different instances

Learner ingredients: adding semantic information

- example:

1. *English*: I should also like to add that these two texts focus, in particular, on strengthening the framework of criminal law in order to fight organised **rings** of facilitators.

   Dutch: Ter verduidelijking wil ik er nog aan toevoegen dat het er in deze twee teksten voornamelijk om gaat het strafrechtelijk kader te versterken om te kunnen optreden tegen **netwerken** voor mensensmokkel.

2. *English*: That figure has now risen to 800000, and the well-organised criminal slave trading **rings** for that is what I call them do not shrink from trafficking in children as well.

   Dutch: Dit aantal is nu gestegen naar 800.000, en de goed georganiseerde criminele **organisaties** van slavenhandelaars, zoals ik deze lieden graag wil noemen, deinzen er niet voor terug om ook kinderen te verhandelen.

3. *English*: It is mainly due to the lack of information among sportsmen and women, and the report therefore proposes that there should be an indicator on the boxes of pharmaceutical products, consisting of five Olympic **rings** and a traffic light.

   Dutch: Deze is hoofdzakelijk het gevolg van een gebrekkige voorlichting aan de sportlieden. In het verslag wordt dan ook voorgesteld om de farmaceutische producten te voorzien van een duidelijk etiket met vijf Olympische **ringen** en een verkeerslicht.

Example LSA

- consider the two most important dimensions that result from the SVD reduction on the three example sentences
- the first two sentences are much more correlated than the third sentence, which is characterized by very different values → SVD is indeed capable of finding correlations between terms that are semantically close and collapses them into the same dimension in the new representation.

	Sentence 1	Sentence 2	Sentence 3
dim_1	1.321	1.233	3.243
dim_2	-0.507	-0.861	1.295

Gensim

gensim

topic modelling for humans

How to build a NLP system??

1. Step 1: collect data for your NLP problem to be solved
2. Step 2: annotate
3. Step 3: build feature vectors
4. Step 4: choose appropriate ML algorithm and evaluate

Evaluation

- Training data to learn and validate the learning algorithm
- Test data
- (sometimes) Development data

Evaluation (ctd.)

n-fold cross-validation

- separate the training data in *n* parts
- repeat *n* times: take every part once as test part and the other *n*-1 parts as training part

WSD data set

- load “coach.arff” data set from http://www.lt3.ugent.be/semeval/coach/
- run classification with memory based learning (lazy/IB1)
- run clustering with simpleKMeans
- inspect the errors; discuss.