CS544: Information Extraction,

Named Entity Recognition and Classification

March 23, 2010

Zornitsa Kozareva
USC/ISI
Marina del Rey, CA
kozareva@isi.edu
www.isi.edu/~kozareva

Named Entity Recognition and Classification

<PER>**Prof. Jerry Hobbs**</PER> taught CS544 during <DATE>**February 2010**</DATE>.
<PER>**Jerry Hobbs**</PER> killed his daughter in <LOC>**Ohio**</LOC>.
<ORG>**Hobbs corporation**</ORG> bought <ORG>**FbK**</ORG>.

- Identify mentions in text and classify them into a predefined set of categories of interest:
  - Person Names: **Prof. Jerry Hobbs**, **Jerry Hobbs**
  - Organizations: **Hobbs corporation**, **FbK**
  - Locations: **Ohio**
  - Date and time expressions: **February 2010**
  - E-mail: **mkg@gmail.com**
  - Web address: **www.usc.edu**
  - Names of drugs: **paracetamol**
  - Names of ships: **Queen Marry**
  - Bibliographic references:
  - ...

1

Why simple things would not work?

- Capitalization is a strong indicator for capturing proper names, but it can be tricky because:
  - nouns in German are capitalized
  - first word of a sentence is capitalized
  - in nested named entity

    University of Southern California is Organization
  - sometimes titles in web pages are all capitalized
- Currently, no gazetteer contains all existing proper names.
- New proper names constantly emerge

  movie titles, books, singers etc.

2

Why simple things would not work?

- The same entity can have multiple variants of the same proper name

  Beyonce

  Beyonce Knowles

  B

- Proper names are ambiguous

  Jordan the *person* vs. Jordan the *location*

  JFK the *person* vs. JFK the *airport*

  May the *person* vs. May the *month*
- Proper names have abbreviations and acronyms

  ***I**nformation **S**ciences **I**nstitute* and *ISI*

3

Knowledge NER vs. Learning NER

<u>Knowledge Engineering</u>

+ very precise (hand-coded rules)

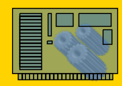
+ small amount of training data

- expensive development & test cycle

- domain dependent

-changes over time are hard

<u>Learning Systems</u>



+ higher recall

+ no need to develop grammars

+ developers do not need to be experts

+ annotations are cheap

-require lots of training data

4

Rule Based NER (1)

- **Create regular expressions:** a set of pattern matching rules encoded in a string according to certain syntax rules.

Suppose you are looking for a word that:

1. starts with a capital letter "P"
2. is the first word on a line
3. the second letter is a lower case letter
4. is exactly three letters long
5. the third letter is a vowel

the regular expression would be "^P[a-z][aeiou]" where

^ - indicates the beginning of the string
[a-z] – any letter in range a to z
[aeiou] – any vowel

5

Perl RegEx

- \w (word char) any alpha-numeric
- \d (digit char) any digit
- \s (space char) any whitespace
- . (wildcard) anything
- \b word bounday
- ^ beginning of string
- $ end of string
- ? For 0 or 1 occurrences
- + for 1 or more occurrences
- specific range of number of occurrences: {min,max}.
  - A{1,5} One to five A's.
  - A{5,} Five or more A's
  - A{5} Exactly five A's

6

Rule Based NER (1)

- **Create regular expressions**
  - E-mail
  - Capitalized names
  - Telephone number

blocks of digits separated by hyphens

RegEx = (\d+\-)+\d+

- matches valid phone numbers like 900-865-1125 and 725-1234
- incorrectly extracts social security numbers 123-45-6789
- fails to identify numbers like 800.865.1125 and (800)865-CARE

Improved RegEx = (\d{3}[-.\ ()]){1,2}[\dA-Z]{4}

7

Rule Based NER (2)

- **Create rules like**
  - Capitalized word + {city, center, river} indicates location
    Ex. *New York city*
    Hudson river
  - Capitalized word + {street, boulevard, avenue} indicates location
    Ex. *Fifth avenue*

8

Rule Based NER (3)

- **Use context patterns**
  - [*PERSON*] earned [*MONEY*]
    Ex. *Frank earned $20*
  - [*PERSON*] joined [*ORGANIZATION*]
    Ex. *Sam joined IBM*
  - [*PERSON*],[*JOBTITLE*]
    Ex. *Mary, the teacher*

  still not so simple:
  - [*PERSON*|*ORGANIZATION*] fly to [*LOCATION*|*PERSON*|*EVENT*]
    Ex. *Jerry flew to Japan*
    Sarah flies to the party
    Delta flies to Europe

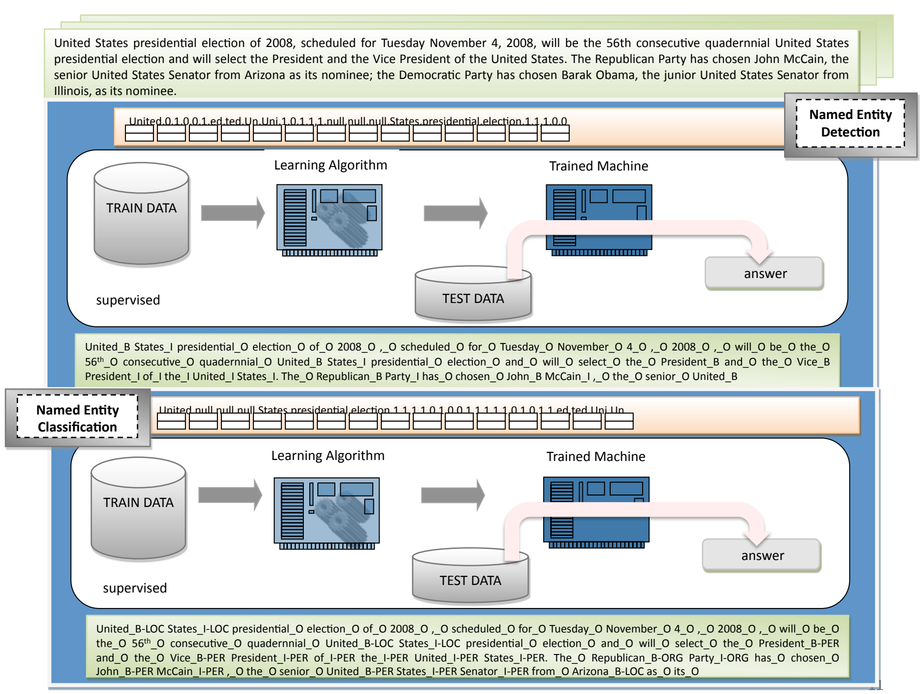
9

Machine Learning NER

Adam_B-PER Smith_I-PER works_O for_O **IBM_B-ORG** ,_O **London_B-LOC** ._O

- **NED**: Identify named entities using BIO scheme
  - B beginning of an entity
  - I continues the entity
  - O word outside the entity
- **NEC**: Classify into a predefined set of categories
  - Person names
  - Organizations (companies, governmental organizations, etc.)
  - Locations (cities, countries, etc.)
  - Miscellaneous (movie titles, sport events, etc.)

10



Adapted from Raymond Mooney

Learning for Categorization

- A training example is an instance $x \in X$, paired with its correct category $c(x)$: $<x, c(x)>$ for an unknown categorization function, c.
- Given:
  - A set of training examples, T.
  - A hypothesis space, H, of possible categorization functions, $h(x)$.
- Find a consistent hypothesis, $h(x) \in H$, such that:

$$\forall < x, c(x) > \in T : h(x) = c(x)$$

12

k Nearest Neighbor

- Learning is just storing the representations of the training examples.
- Testing instance x_p:
  - compute similarity between x_p and all training examples
  - take vote among x_p k nearest neighbours
  - assign x_p with the category of the most similar example in T

13

Distance measures

- Nearest neighbor method uses similarity (or distance) metric.

- Given two objects *x* and *y* both with *n* values

$$x = (x_1, x_2, \ldots, x_n)$$
$$y = (y_1, y_2, \ldots, y_n)$$

calculate the Euclidean distance as

$$d(x,y) = \sqrt[2]{\sum_{i=1}^{p} |x_i - y_i|^2}$$

14

An Example

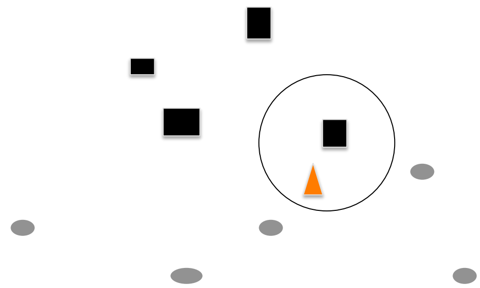
	isPersonName	**isCapitalized**	**isLiving**
Jerry Hobbs	1	1	1
USC	0	1	0

Euclidean distance:

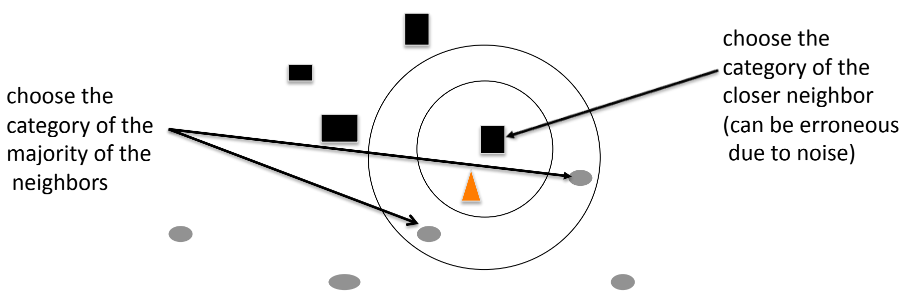
$$d(JerryHobbs, USC) = \sqrt[2]{(1^2 + 0 + 1^2)} = 1.41$$

15

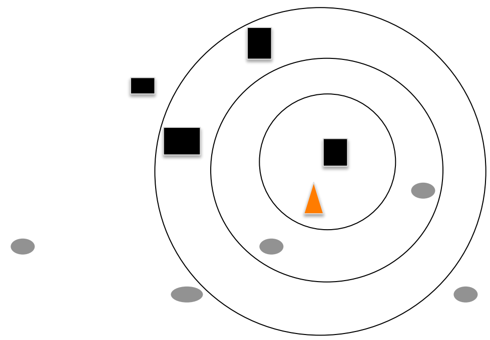
1-Nearest Neighbor



3-Nearest Neighbor



5-Nearest Neighbor

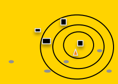


the value of *k* is typically odd to avoid ties

18

k Nearest Neighbours

Pros

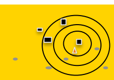


+ robust

+ simple

+ training is very fast (storing examples)

Cons



- depends on similarity measure & k-NNs

- easily fooled by irrelevant attributes

- computationally expensive

19

Decision Trees

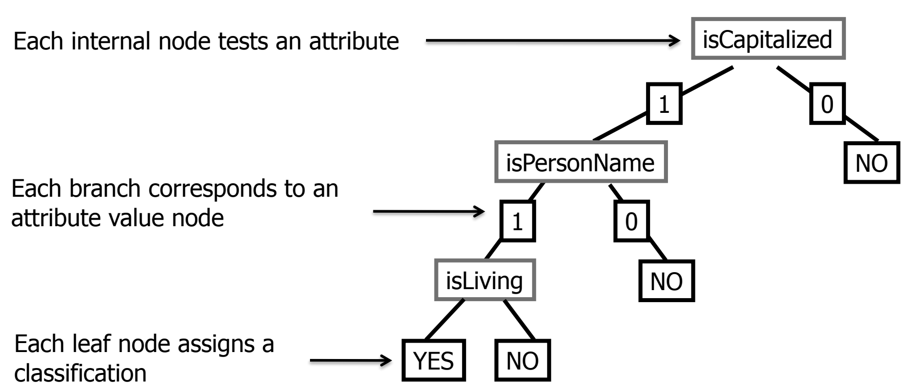
- The classifier has a tree structure, where each node is either:
  - a leaf node which indicates the value of the target attribute (class) of examples
  - a decision node which specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test

- An instance x_p is classified by starting at the root of the tree and moving through it until a leaf node is reached, which provides the classification of the instance

20

An Example

	isPersonName	isCapitalized	isLiving	X is PersonName?
profession	0	0	0	NO
Jerry Hobbs	1	1	1	YES
USC	0	1	0	NO
Jordan	1	1	0	NO



21

Building Decision Trees

- Select which attribute to test at each node in the tree.
- The goal is to select the attribute that is most useful for classifying examples.
- Top-down, greedy search through the space of possible decision trees. It picks the best attribute and never looks back to reconsider earlier choices.

22

Decision Trees

Pros

+ generate understandable rules

+ provide a clear indication of which features are most important for classification

Cons

-error prone in multi-class classification and small number of training examples

- expensive to train due to pruning

23

Carreras et al. 2002

- Learning algorithm: AdaBoost
- Binary classification
- Binary features

- $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$ (Schapire & Singer, 99)
- Weak rules (h_t): Decision Trees of fixed depth.

24

Features for NE Detection

- **Contextual**
  - current word W_0
  - words around W_0 in [-3,…,+3] window
- **Part-of-speech tag** (when available)
- **Orthographic (binary and not mutually exclusive)**

initial-caps	*all-caps*	*all-digits*
roman-number	*contains-dots*	*contains-hyphen*
acronym	*lonely-initial*	*punctuation-mark*
single-char	*functional-word**	*URL*

- **Word-Type Patterns:**

functional	*lowercased*	*quote*
capitalized	*punctuation mark*	*other*

- **Left Predictions**
  - the tag predicted in the current classification for W_{-3}, W_{-2}, W_{-1}

**functional-word is preposition, conjunction, article*

25

Results for NE Detection

CoNLL-2002 Spanish Evaluation Data		
Data sets	#tokens	#NEs
Train	264,715	18,794
Development	52,923	4,351
Test	51,533	3,558

Evaluation Measures

$$\text{Precision} = \frac{\#\,\text{correct identified NEs}}{\#\,\text{identified NEs}}$$

$$\text{Recall} = \frac{\#\,\text{correct identified NEs}}{\#\,\text{gold standard data}}$$

Carreras et al.,2002	Precision	Recall	F-score
BIO dev.	92.45	90.88	91.66

26

Features for NE Classificaton (1)

- **Contextual**
  - current word W_0
  - words around W_0 in [-3,…,+3] window
- **Part-of-speech tag** (when available)
- **Bag-of-Words**
  - words in [-5,…,+5] window
- **Trigger words**
  - for person (*Mr, Miss, Dr, PhD*)
  - for location (*city, street*)
  - for organization (*Ltd., Co.*)
- **Gazetteers**
  - geographical
  - first name
  - surname

27

Features for NE Classificaton (2)

- Length in words of the entity being classified
- Pattern of the entity with regard to the type of constitutent words
- **For each classs**
  - whole NE is in gazetteer
  - any component of the NE appears in gazetteer
- **Suffixes** (length 1 to 4)
  - each component of the NE
  - whole NE

28

Results for NE Classification*

Spanish Dev.	Precision	Recall	F-score
LOC	79.04	80.00	79.52
MISC	55.48	54.61	55.04
ORG	79.57	76.06	77.77
PER	87.19	86.91	87.05
overall	79.15	77.80	78.47

Spanish Test.	Precision	Recall	F-score
LOC	85.76	79.43	82.47
MISC	60.19	57.35	58.73
ORG	81.21	82.43	81.81
PER	84.71	93.47	88.87
overall	81.38	81.40	81.39

System of Carreras et al.,2002

29

Homework
Named Entity Challenge

30

- Given: a train and development set of English sentences tagged with four named entity classes:
  - PER (people)
  - ORG (organization)
  - LOC (location)
  - MISC (miscellaneous)
- Your objective is: to develop a machine learning NE system, which when given a new previously unseen text (i.e. test set) will identify and classify the named entities correctly

31

Data Description

- The data consists of two columns separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence.

U.N. NNP I-ORG
official NN O
Ekeus NNP I-PER
heads VBZ O
for IN O
Baghdad NNP I-LOC
. . O

word
part-of-speech-tag
named entity tag

I-TYPE means the word is inside a phrase of type TYPE
O means the word is not part of a phrase

Timeline

	Release
Train and Development data	March 24th 2010
Test data	April 9th 2010
Result submission deadline	April 10th 2010 (11:59 pm) later submissions will not be accepted
Presentation submission deadline	April 13th 2010

Submit (1)

- The source code for the feature generation (**make sure it will run under Linux**)
- The official train and test feature files used in the final run, together with the final output of your system for the test data
- The additionally generated resources (if any)
- Write 1-2 page brief description of your approach explaining:
  - the used NLP tools
  - the designed features
  - the employed machine learning algorithm

34

Submit (2)

- Make a short power point presentation which you will present in 3 minutes to the class on April 15th.
- Please, be prompt so I can include your slides in the set to be presented
- Note you will have maximum 3 minutes to present your work in class, make sure your presentation is to the point

35

Evaluation is based on

- the ranking of your system against the rest
- the designed features
  - novel, previously unknown features will be favored
  - system's pre or post processing
  - a study on the groups of features used
- the generated resources
  - size, methods and sources for gazetteer extraction
  - trigger lists

36

Generate Your Own Resources

- Extract gazetteers from Wikipedia
  - People (singers, teachers, mathematicians etc.)
  - Locations (cities, countries)
  - Organizations (universities, IT companies etc.)
- Extract trigger words from WordNet
  - look for hyponyms of person, location, organization
- Extract and rank the patterns in which the NEs occurred in the train and development data. Show what percentages of these were found in the final test data.
- Extract lists of verbs found next to the NEs. Do you find any similarity/regularity of the verbs associated with each one of the NE categories?

37

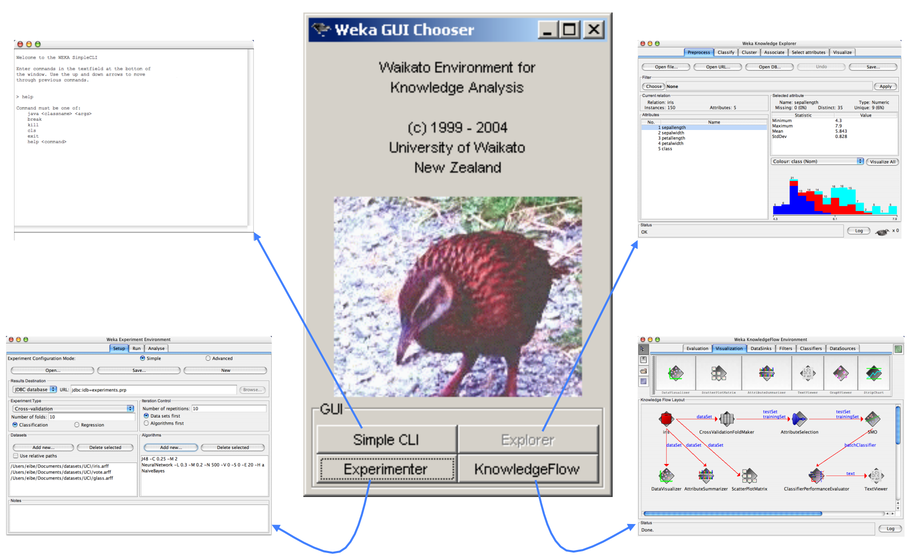
What must I do …

- Use the train and development data to design and tune your NE system
- Decide on the features you would like to incorporate in your NE system
- Choose a machine learning classifier from Weka
  - http://www.cs.waikato.ac.nz/ml/weka/
  - Intro by Marti Hearst
    http://courses.ischool.berkeley.edu/i256/f06/lectures/lecture16.ppt
- **This is a big assignment so start early!**

38

WEKA GUI Chooser

java **-Xmx1000M** -jar weka.jar



39

WEKA File Format: ARFF

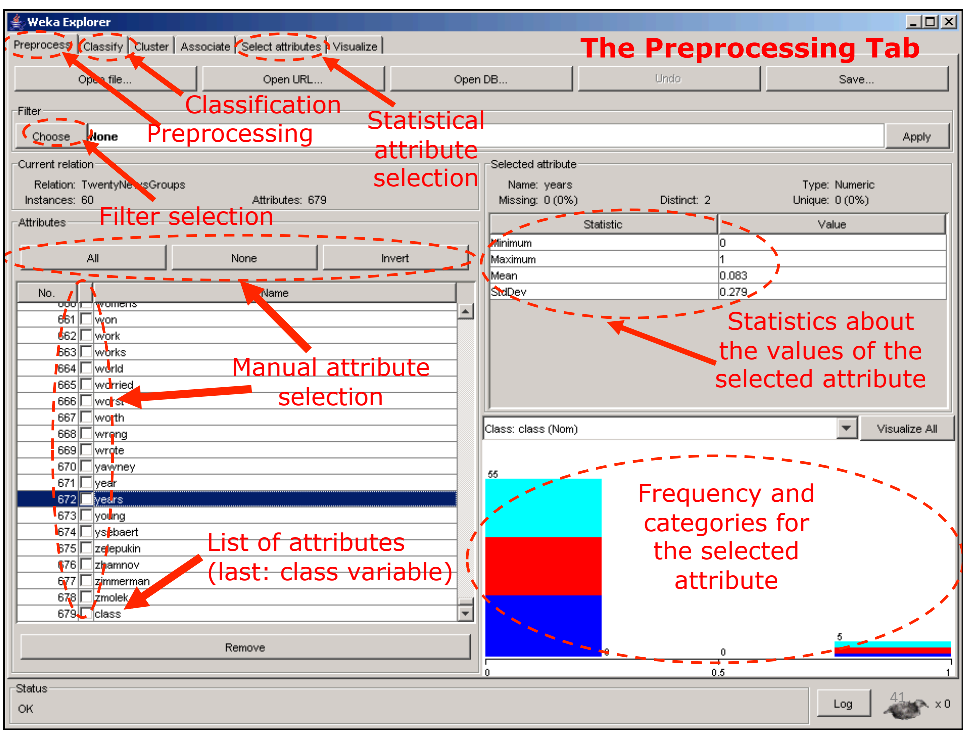
@relation english_named_entity

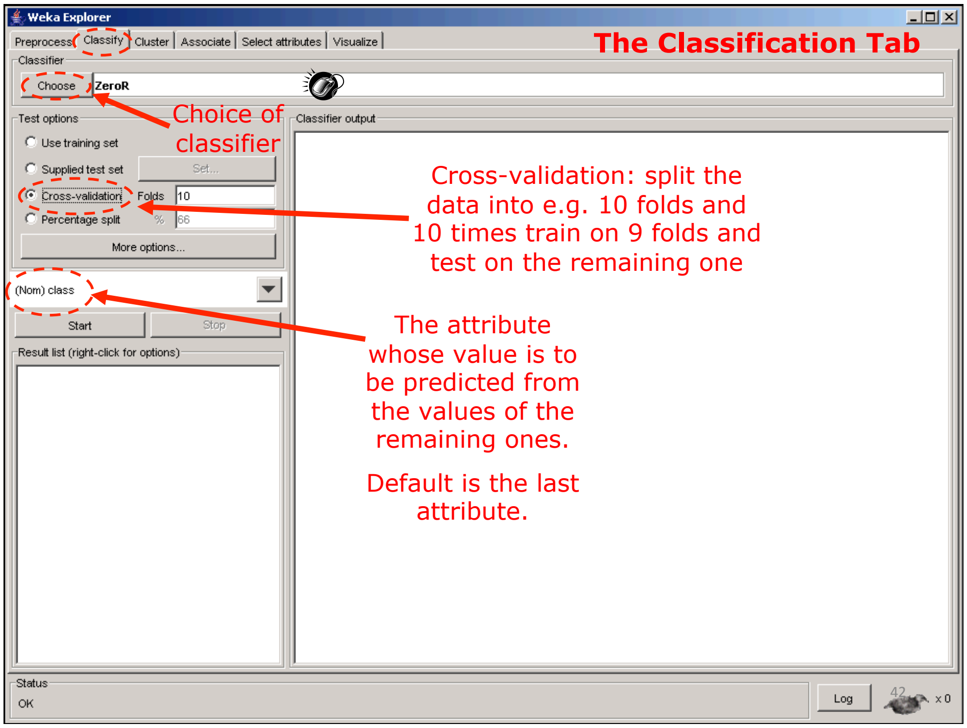
@attribute position **numeric**
@attribute pos_tag { NN, NP, VB, DT}
@attribute word_length numeric
@attribute in_gazetteer { no, yes}
@attribute class { PER, LOC, ORG, MISC}

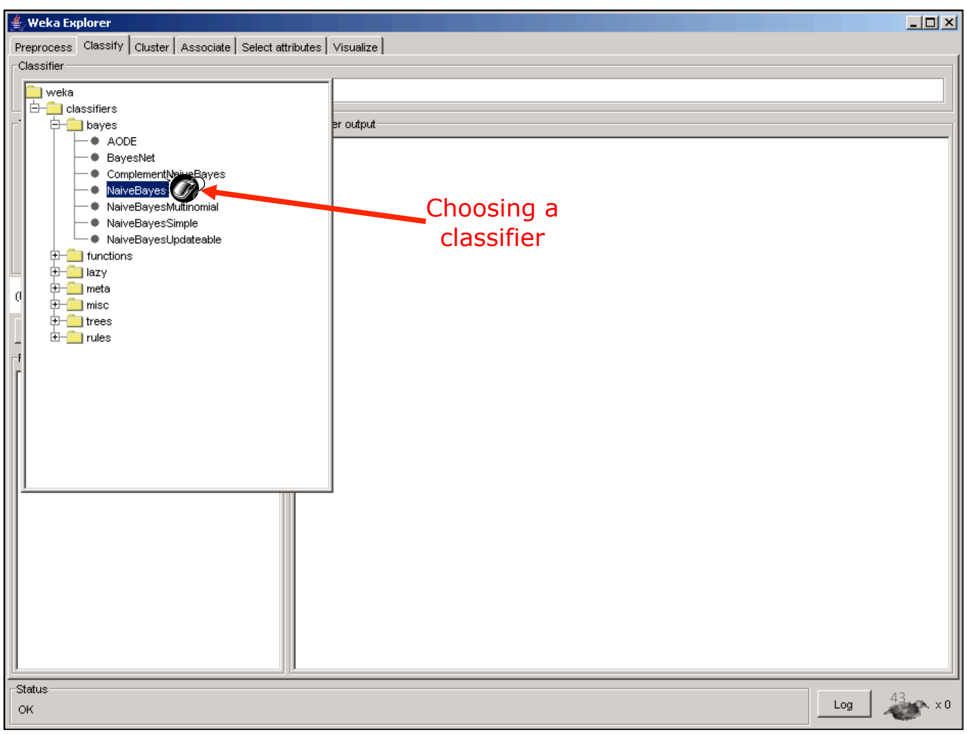
@data
3,DT,3,no,ORG
4,NP,10,yes,ORG
15,NP,6,yes,PER
7, NN,12,**?**,MISC
...

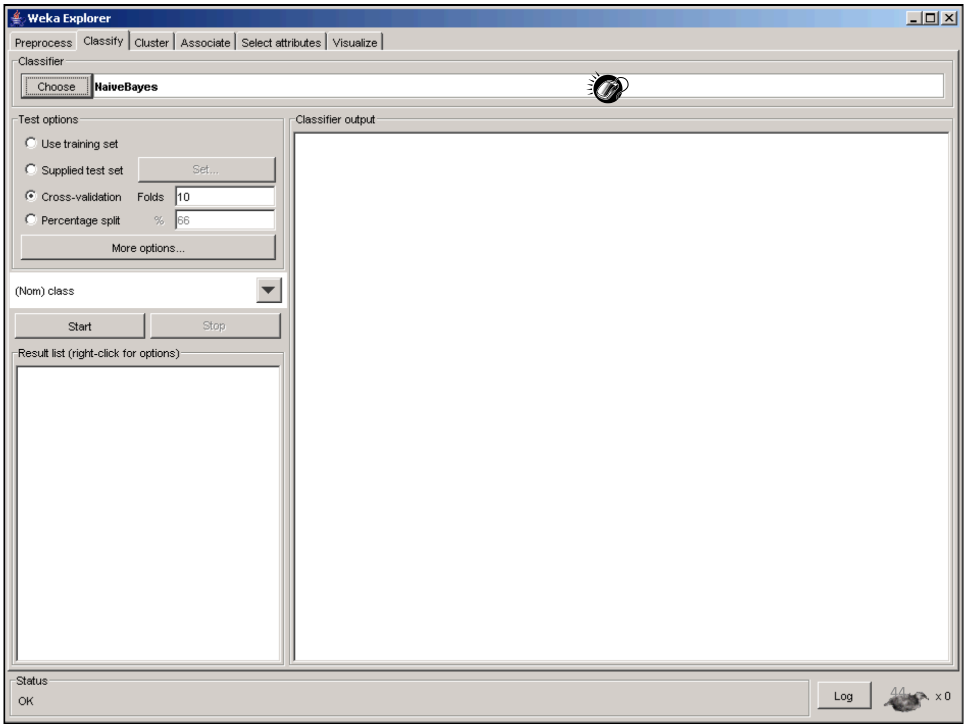
Missing value

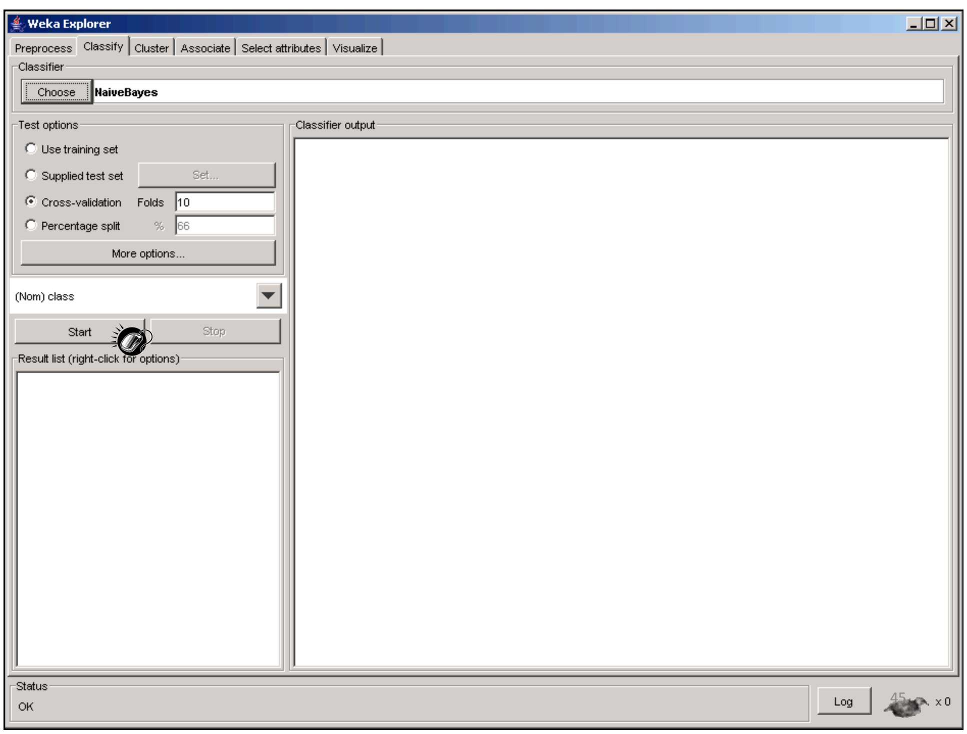
Other attribute types:
- String
- Date

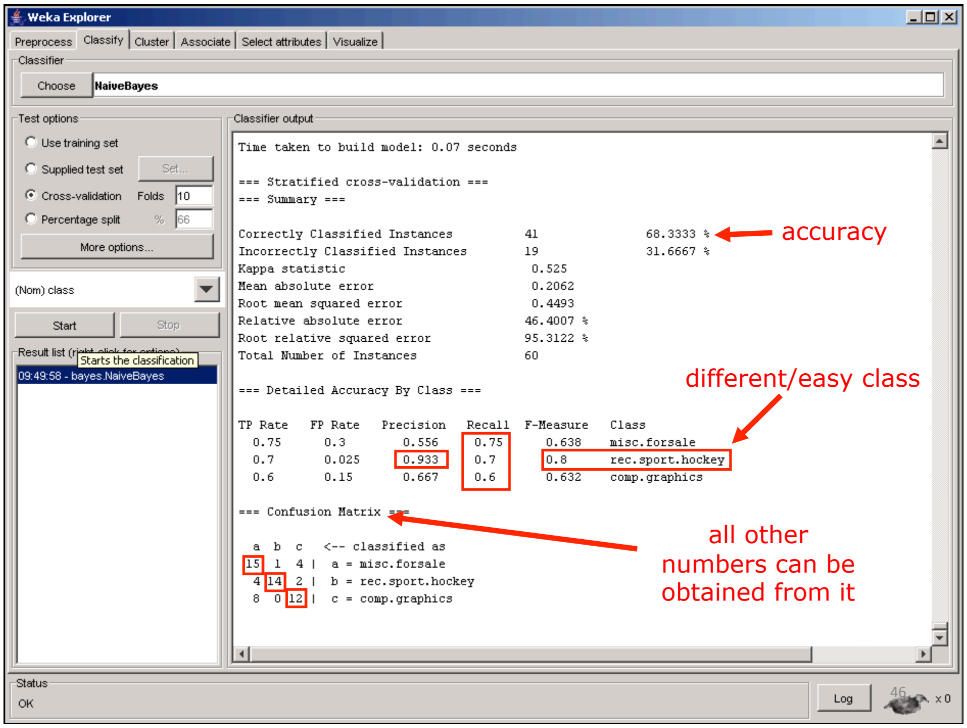



Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
The Classification Tab
Classifier
Choose
ZeroR
Choice of classifier
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
Cross-validation: split the data into e.g. 10 folds and 10 times train on 9 folds and test on the remaining one
(Nom) class
Start
Stop
Result list (right-click for options)
The attribute whose value is to be predicted from the values of the remaining ones.
Default is the last attribute.
Status
OK
Log
x 0

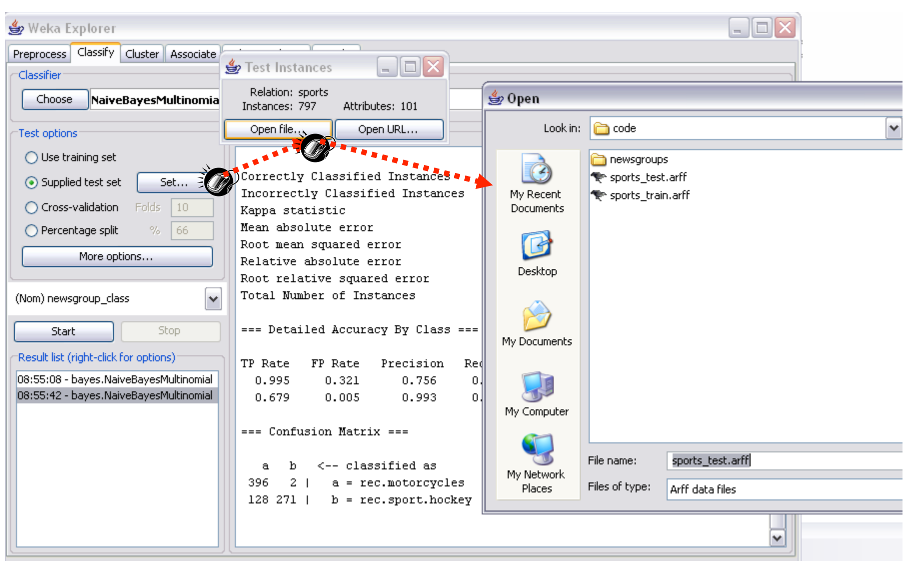

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
AODE
BayesNet
ComplementNaiveBayes
NaiveBayes
NaiveBayesMultinomial
NaiveBayesSimple
NaiveBayesUpdateable
functions
lazy
meta
misc
trees
rules
Choosing a classifier
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier output
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
(Nom) class
Start
Stop
Result list (right-click for options)
Classifier output
Status
OK
Log
x 0



Running on Test Set



47

Available Resources

- WordNet http://wordnet.princeton.edu/
- Part-of-speech taggers
  - TreeTagger http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
  - Stanford PoS Tagger http://nlp.stanford.edu/software/tagger.shtml
- NP chunker
  - http://www.dcs.shef.ac.uk/~mark/index.html?http://www.dcs.shef.ac.uk/~mark/phd/software/chunker.html
- Parser
  - Stanford Parser http://nlp.stanford.edu/software/lex-parser.shtml
- Named Entity Recognizer
  - Stanford NER http://nlp.stanford.edu/software/CRF-NER.shtml
  - LingPipe http://alias-i.com/lingpipe/
  - ANNIE http://www.aktors.org/technologies/annie/
- Other

  http://nlp.stanford.edu/links/statnlp.html

48

Good Luck!

49