A Gentle Introduction to Machine Learning in Natural Language Processing using R

ESSLLI '2013
Düsseldorf, Germany

http://ufal.mff.cuni.cz/mlnlpr13

Barbora Hladká
hladka@ufal.mff.cuni.cz

Martin Holub
holub@ufal.mff.cuni.cz

Charles University in Prague,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics

Course organization

- **Course webpage:** `http://ufal.mff.cuni.cz/mlnlpr13`
- All materials will be available at the web page
- We will post every day after the lesson
    - slides
    - data and R scripts needed for homeworks
- Course is organized in blocks. Please, ask questions between blocks.

The course goal

We do not assume any strong backgroung (mathematical or programming) and our presentation tries to be as intuitive as possible.

- Don't be affraid of anything!
- You cannot expect that you become experts immediately after the course, BUT
- you will be provided with useful instructions, typical examples, helpful hints SO THAT
- you will be able to start learning machine learning and working with R seriously.

******* This 5-days course is a sort of teaser *******

Day 1

- 1.1 Relation between NLP and ML
- 1.2 Course outline
- 1.3 Non-technical view on ML
- 1.4 Dealing with data
- 1.5 Intro to R
- Summary

Block 1.1
Relation between NLP and ML and R

- Research areas
  - Natural Language Processing (**NLP**)
  - Machine Learning (**ML**)
- Software environment
  - **R**

Data

What are we working on computers with?

Texts in word editors. Images in graphic editors. Numbers in spreadsheets. Audios and videos in players. Etc.

In brief, we work with **data**.

Natural Language Processing (NLP)

NLP deals with computer and human interaction in both written and spoken natural language.

Data are texts or speeches, sometimes enriched with linguistic information.

Example of an NLP task

Word-sense disambiguation (WSD)

Assign the correct sense of a word in a sentence.

Fill out the quiz for the word *line* with the following senses:

- `CORD`
- `DIVISION`
- `FORMATION`
- `PHONE`
- `PRODUCT`
- `TEXT`

Example of an NLP task

Word-sense disambiguation

- What knowledge did you use to assign the senses?
- What were the keys for the correct decision?
- Which sentences were easy to recognize the correct sense and which were the most difficult for you?

Example of an NLP task

Word-sense disambiguation

- I've got Inspector Jackson on the **line** for you. PHONE
- Outside, a **line** of customers waited to get in. FORMATION
- He quoted a few **lines** from Shakespeare. TEXT
- He didn't catch many fish, but it hardly mattered.
  With his **line** out, he sat for hours staring at the Atlantic. CORD
- The company has just launched a new **line** of small,
  low-priced computers. PRODUCT
- Draw a **line** that passes through the points P and Q. DIVISION
- This has been a very popular new **line**. PRODUCT? FORMATION?

Machine learning (ML)

ML deals with teaching computers to learn from data presented as **examples**.

"The computer is claiming its intelligence is real, and ours is artificial."

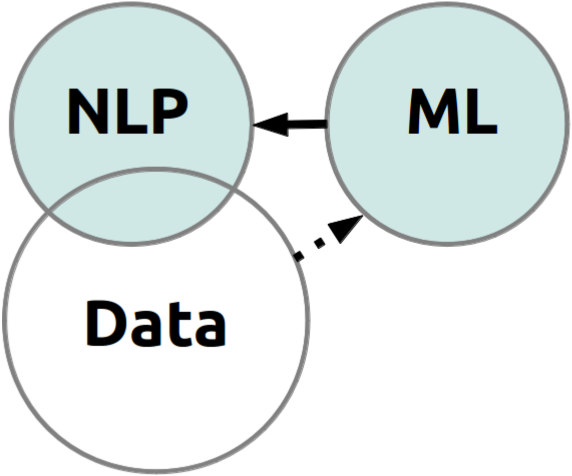
NLP and ML

- We – human beings – do word sense disambiguation easily using the **context in the sentence** and our **knowledge of the world**.
- We want computers to master it as well.
  - Could you implement the procedure how you disambiguate the senses in the quiz?
  - How well would your code perform the task? Does it make errors?

Let's prepare examples and guide computers to learn from them.

That is Machine learning!

ML in NLP

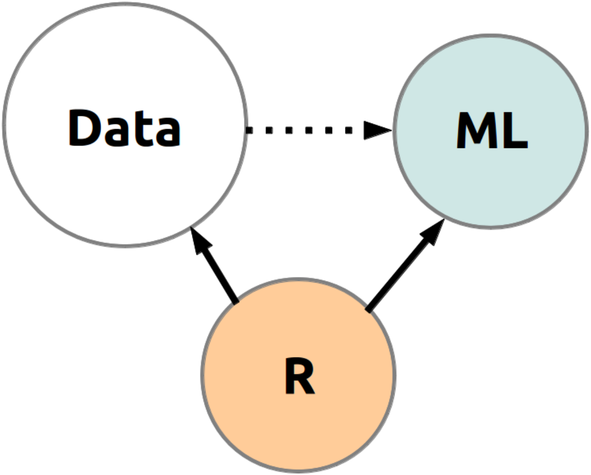


R

R is a software system

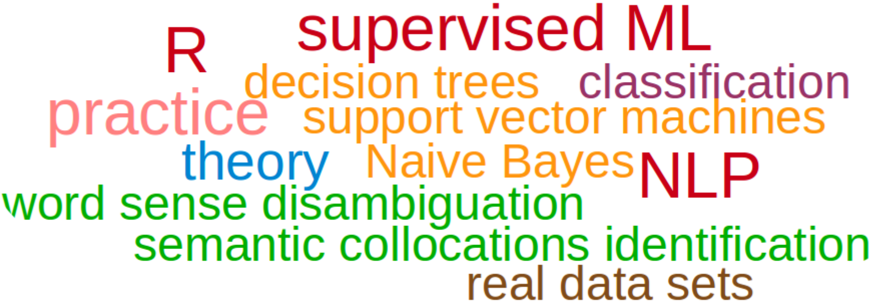
- for data analysis, and
- an environment of ML methods as well.

ML using R



Block 1.2
Course outline

Course word cloud



Goals & Ambitions

- **Goal 1**
  To introduce some basic principles and algorithms of ML **theoretically**.
- **Ambition 1**
  The students do know what is ML on, what are the fundamental concepts and why it's useful to know it.

Goals & Ambitions

- **Goal 2**
  To introduce these principles and algorithms of ML **practically**.

- **Ambition 2**
  The students do know that they don't have to implement ML algorithms themselves because there already exists such a system (out of many). The students do know how to use this system.

Goals & Ambitions

- **Goal 3**
  To **demonstrate ML** techniques for the **natural language processing** tasks.

- **Ambition 3**
  The students do know how to run a ML experiment from the very beginning to the end.

Block 1.3
A non-technical view on ML

Example: Teach kids to do word sense disambiguation of *line*.

1. You are a teacher.
2. Lend English dictionary in a library.
3. Discuss with kids the meanings of *line* listed in the dictionary.
    - Focus on the context of *line* in example sentences.
4. Prepare a quiz to test the kids.
   Select new sentences that kids have NOT seen in the dictionary.
5. Evaluate kids' answers.

Will all the kids get "A" grade?

May be, and may be not. Why?

- They have not seen all possible sentences with *line* during their learning.

How to get better grade?

1. Get more example sentences, e.g. lend more English dictionaries.

2. Focus on other specific characteristics, e.g. part of speech classes that occur in the context of *line*.

Another quiz

- Test kids on sentences that they already SAW in the dictionary.
- Will all the kids get "A" grade?
- May be, and may be not. Why?

There can appear ambiguous sentences in the quiz.

Machine learning process

Teach computers to learn from examples in five essential steps

1. **Formulating the task**
   ("Assign the correct sense of *line* in a sentence.")
2. **Getting examples**
   Splitting them into training and test examples.
   (Getting the dictionary)
3. **Learning from training data**
   (Understanding the examples in the dictionary)
4. **Testing the learned knowledge on test data**
   (Taking the quiz)
5. **Evaluation**
   (Evaluation of kids' answers)

Machine learning process

Learning from training data

Application of a ML method results in a **trained model** (= learned knowledge).

Example of an NLP task

Semantic collocations identification (COL)

Decide whether the given word pair forms a semantic collocation.

Example: *green card, grey market*.

- **Semantic collocations** are multiword expressions that are lexically, syntactically, pragmatically and/or statistically **idiosyncratic**.
- In other words, semantic collocations have semantic and/or syntactic properties that cannot be fully predicted from their components, and therefore they **have to be listed in a dictionary**.

Example of an NLP task

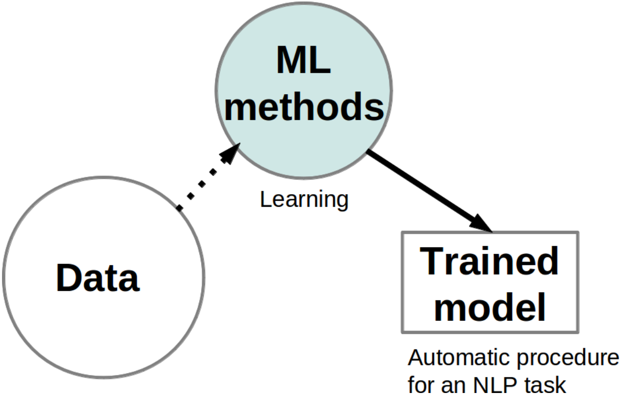
Semantic collocations (COL)

example	translation	description
Masarykův okruh	Masaryk circuit	motor sport race track named after the first president of Czechoslovakia, Tomáš Garrigue Masaryk
šedá ekonomika	gray market	legal trade of a commodity through unofficial distribution channels
Antonín Dvořák	Antonin Dvorak	Czech composer
trest smrti	death penalty	person is put to death by the state as a punishment for a crime
zelená karta	green card	ID card attesting to the permanent resident status of an immigrant in the USA
rovnoramenný trojúhelník	isosceles triangle	triangle with two sides equal in length

Block 1.4
Dealing with data for machine learning

The very basics of data manipulation and analysis

Data processing is one of crucial parts of the ML technology!



WSD as a classification task

Word Sense Disambiguation (WSD) is a typical example of a classification task.

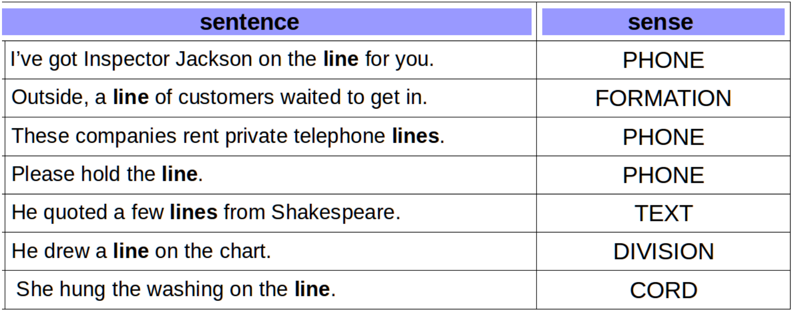
- WSD input = a sentence containing the target word “line”
- WSD output = one of the possible “output values” {`CORD`, `DIVISION`, `FORMATION`, `PRODUCT`, `PHONE`, `TEXT`}
- Input sentences are objects of classification, the possible senses are classes.

Classification task means that the **output value is discrete and represents possible classes**. In classification tasks we generally want to classify objects into an a priori given set of classes.

In this course we focus only on classification tasks.

Classification task as a supervised learning process

Intuitively we need a large set of classified **examples** to learn the essential knowledge necessary to recognize correct senses. Examples used for learning are called **training data**.



sentence	sense
I've got Inspector Jackson on the **line** for you.	PHONE
Outside, a **line** of customers waited to get in.	FORMATION
These companies rent private telephone **lines**.	PHONE
Please hold the **line**.	PHONE
He quoted a few **lines** from Shakespeare.	TEXT
He drew a **line** on the chart.	DIVISION
She hung the washing on the **line**.	CORD

In the process of supervised machine learning training examples are data that should be carefully processed and analyzed.

What computer extracts from examples

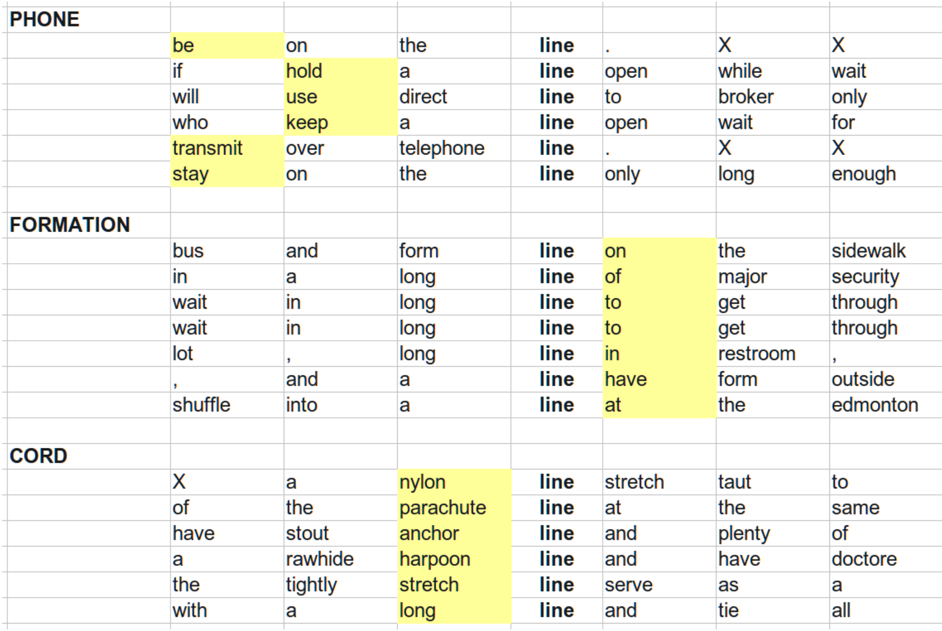
Both humans and computers need to know the **context of the target word** (“line”) to recognize correct senses.

Humans use their reason, intuition, and their real world knowledge.

Computers need to extract a limited set of useful **context clues** that are then used for automatic decision about the correct sense.

- Formally, the context clues are called **attributes or features** and should be exactly and explicitly defined.
- Then each object (a sentence) is characterized by a list of attributes, which is also called **feature vector**.

Computer makes feature vectors from examples.



PHONE							
	be	on	the	**line**	.	X	X
	if	hold	a	**line**	open	while	wait
	will	use	direct	**line**	to	broker	only
	who	keep	a	**line**	open	wait	for
	transmit	over	telephone	**line**	.	X	X
	stay	on	the	**line**	only	long	enough
FORMATION							
	bus	and	form	**line**	on	the	sidewalk
	in	a	long	**line**	of	major	security
	wait	in	long	**line**	to	get	through
	wait	in	long	**line**	to	get	through
	lot	,	long	**line**	in	restroom	,
	,	and	a	**line**	have	form	outside
	shuffle	into	a	**line**	at	the	edmonton
CORD							
	X	a	nylon	**line**	stretch	taut	to
	of	the	parachute	**line**	at	the	same
	have	stout	anchor	**line**	and	plenty	of
	a	rawhide	harpoon	**line**	and	have	doctore
	the	tightly	stretch	**line**	serve	as	a
	with	a	long	**line**	and	tie	all

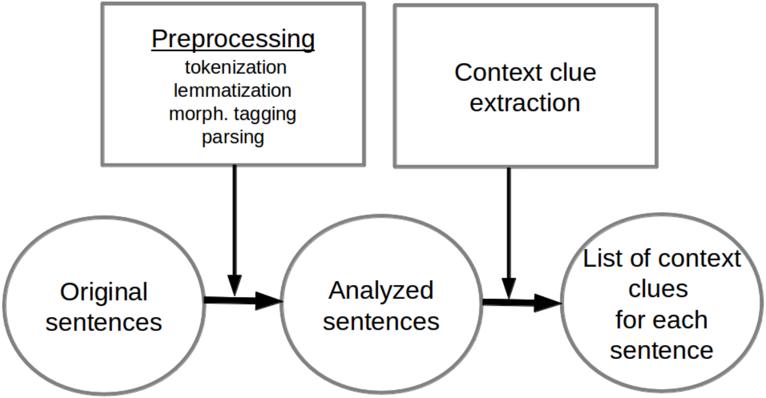
Intuitive feature extraction – examples

To choose an effective set of features we always need our intuition
Only then experiments with data can start

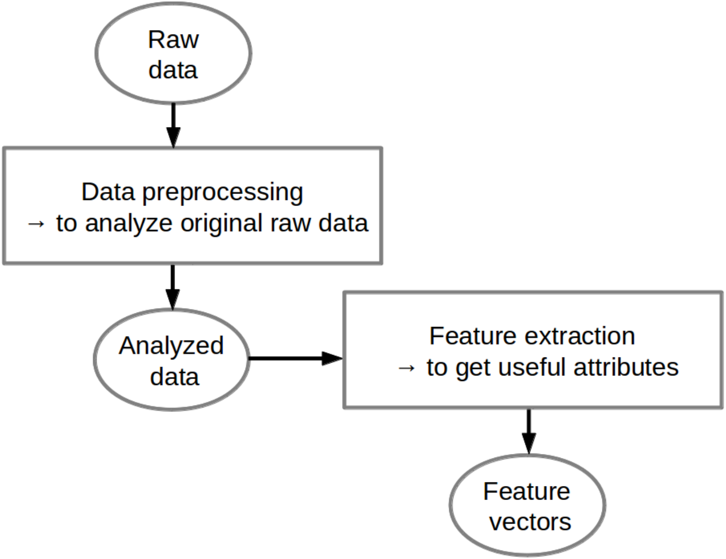
A few example hints:

class	**a feature to recognize the class – will be useful?**
`CORD`	immediately preceding word
`FORMATION`	immediately following word
`PHONE`	can be often recognized by characteristic verbs

Data preprocessing – the WSD task

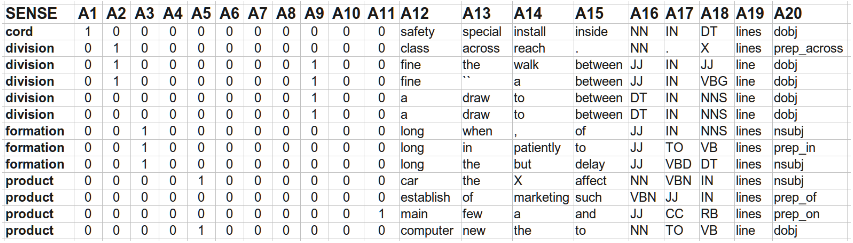


Data preprocessing – a general scheme



Example feature vectors – the WSD task

Each data instance is a list of feature values and target class value



SENSE	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
cord	1	0	0	0	0	0	0	0	0	0	0	safety	special	install	inside	NN	IN	DT	lines	dobj
division	0	1	0	0	0	0	0	0	0	0	0	class	across	reach	.	NN	.	X	lines	prep_across
division	0	1	0	0	0	0	0	0	1	0	0	fine	the	walk	between	JJ	IN	JJ	line	dobj
division	0	1	0	0	0	0	0	0	1	0	0	fine	``	a	between	JJ	IN	VBG	line	dobj
division	0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
division	0	0	0	0	0	0	0	0	1	0	0	a	draw	to	between	DT	IN	NNS	line	dobj
formation	0	0	1	0	0	0	0	0	0	0	0	long	when	,	of	JJ	IN	NNS	lines	nsubj
formation	0	0	1	0	0	0	0	0	0	0	0	long	in	patiently	to	JJ	TO	VB	lines	prep_in
formation	0	0	1	0	0	0	0	0	0	0	0	long	the	but	delay	JJ	VBD	DT	lines	nsubj
product	0	0	0	0	1	0	0	0	0	0	0	car	the	X	affect	NN	VBN	IN	lines	nsubj
product	0	0	0	0	0	0	0	0	0	0	0	establish	of	marketing	such	VBN	JJ	IN	lines	prep_of
product	0	0	0	0	0	0	0	0	0	0	1	main	few	a	and	JJ	CC	RB	lines	prep_on
product	0	0	0	0	1	0	0	0	0	0	0	computer	new	the	to	NN	TO	VB	line	dobj

Types of features

Features are properties of described objects that we can observe or measure.

Feature values can be of several types:

- **numerical**
  – either discrete or continuous
- **binary** (0/1, True/False, Yes/No)
  – can be viewed as a kind of categorical
- **categorical**
  – any list of discrete values, non-numerical

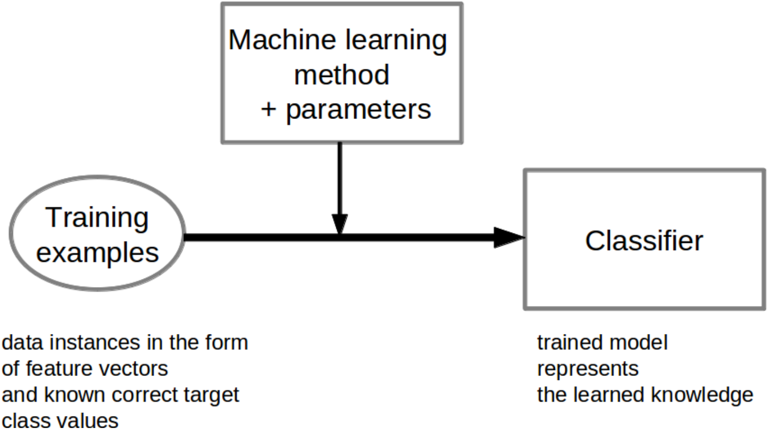
Terminology – features and target class

How different people call values that describe objects

	observed (known) object characteristics	**categories of objects to be predicted**
computer scientists	**features**	**(target) class**
mathematicians (statisticians)	attributes or predictors	response (value) or output value

Supervised learning process

Supervised Machine Learning = computer learns “essential knowledge” extracted from a large set of classified examples


Machine learning method + parameters
Training examples
Classifier
data instances in the form of feature vectors and known correct target class values
trained model represents the learned knowledge

Training data vs. test data

Supervised machine learning necessarily requires

- **Training data** = a set of examples
  – used for **learning process**
- **Test data** = another set of examples
  – used for **evaluation** of a trained model
- **Important**: the split of all available examples into the training and the test portions should be **random**!

“Examples” in supervised ML – two meanings

1) Real examples – Each real object that is already classified or that we want to classify is an example.

2) Data instances – In computer, each real example is represented as a data instance. In this sense

$$\textbf{example} = \textbf{feature vector} + \textbf{target class}$$

Data instances

Sometimes we do not know the target class value; in this case data instances are not different from feature vectors.

data instance = feature vector (+ target class, if it is known)

A data instance is either a feature vector or a complete example.

Homework exercises – general recommendations

We will assign some easy homeworks to you every day. There are three important points to remember:

1. **Homework exercises are NOT obligatory!**
2. **However, we RECOMMEND doing it!**
   – Especially if you want to do your best!
3. **Do NOT be afraid to come tomorrow if you haven't done all assigned homeworks!**

Homework 1.1

1. Download the data set `wsd.development.csv` and open it in your spreadsheet (e.g. MS Office Excel, or LibreOffice Calc).
2. Look at the data – there are
   - 11 binary features
   - 9 categorical features

   For explanation see the handout (`wsd.pdf`).
3. How many examples do you have in this file?
4. Assume that you *randomly* choose an example. What is the most likely `SENSE`? Can you quantify the probability of the most likely `SENSE`?

   – **Hint**: Use the function `countif()` in your spreadsheet.
5. Now the same question on condition that the value of the attribute `A19` is "lines". What is the most likely `SENSE` on that condition?

   – **Advanced**: Is the knowledge of `A19` useful? Can you quantify how much? Think about it!

Block 1.5
Gentle introduction to R

Goals of the practical parts of the course

- to learn how to practically analyse example data and ML tasks
- practical experience with R system for statistical computing and graphics

  `http://www.r-project.org/`
- to solve some easy particular tasks using R

What is R?

R is

- a library of statistical tools
- an interactive environment for statistical analyses and graphics
- a programming language
- a public free software derived from the commercial system S

What is R?

R is

- a library of statistical tools
- an interactive environment for statistical analyses and graphics
- a programming language
- a public free software derived from the commercial system S

R is becoming more and more popular especially for its

- effective data handling and storage facility
- large, coherent, integrated collection of tools for data analysis
- well-developed, simple and effective programming language

Recommended reading on R

There is a lot of sources about R, e.g.

- Getting started with R
  – e.g. `http://data.princeton.edu/R/gettingStarted.html`
- *An Introduction to R*
  by W. N. Venables, D. M. Smith and the R core team
- also, an introduction available on the web:
  `http://cran.r-project.org/doc/manuals/R-intro.html`
- *R for Beginners* by Emmanuel Paradis
- R Language Definition – available at
  `http://cran.r-project.org/doc/manuals/R-lang.html`

A comprehensive list of books about R available at

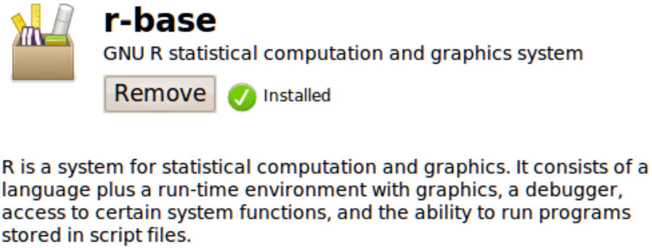
`http://www.r-project.org/`

R – how to install

R is very easy to install

Freely available for MS Windows, MacOS, and Linux as well
– see `http://www.r-project.org/`

- in Ubuntu Linux



R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.

Running R in terminal

```
martin@dragon:~> R

R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i486-pc-linux-gnu (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

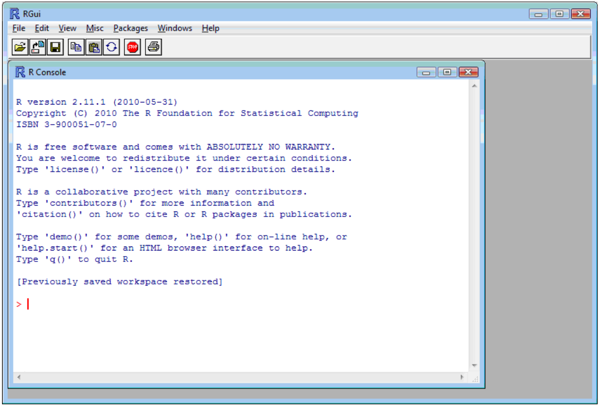
Running RGui (e.g. under MS Windows)

If you like graphical user interfaces, you can run "RGui"

Or you can install and run RStudio – `http://www.rstudio.com/ide/`

Also for Linux – "R commander", "RKWard"

There is no difference in coding in comparison with the terminal session.



```
R version 2.11.1 (2010-05-31)
Copyright (C) 2010 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

R as a calculator

Using R you can compute arithmetic expressions

```
> (5*2-3)^5
[1] 16807
> factorial(10)
[1] 3628800
> factorial(20)
[1] 2.432902e+18
>
```

“Almost every” object in R is a vector or a function

To **assign a new value** to an object use the operator `<-`.

Very often data objects in R are **vectors**.

Numerical vector is an **ordered list of numbers**.

To display the internal ***str*ucture** of an R object use the function `str()`.

To **create a vector** with more than one element use the function `c()`.

```
> x <- 2 + 3
> x
[1] 5
> str(x)
 num 5
> length(x)
[1] 1
> y <- c(9,x,2,x,1,2)
> str(y)
 num [1:6] 9 5 2 5 1 2
> mean(y)
[1] 4
```

Vectors in R

Vector arithmetics is applied on vector elements

```
> y
[1] 9 5 2 5 1 2
> z <- 1:6
> z
[1] 1 2 3 4 5 6
> y + z
[1] 10  7  5  9  6  8
> (y + z)^2
[1] 100  49  25  81  36  64
>
```

You can select vector elements using []

```
> (y + z)[2:4]
[1] 7 5 9
> y[c(2,5,6)]
[1] 5 1 2
>
```

Random sequences

```
> sample(1:6)
[1] 4 6 1 5 3 2
>

# also, the same can be obtained by using
> sample(6)
[1] 4 6 2 1 3 5
>
```

Using help

The built-in help is really helpful

Use `help(<function>)` or `?<function>`.

```
> ?sample

sample                  package:base                R Documentation

Random Samples and Permutations

Description:

     ‘sample’ takes a sample of the specified size from the elements of
     ‘x’ using either with or without replacement.

Usage:

     sample(x, size, replace = FALSE, prob = NULL)

     sample.int(n, size = n, replace = FALSE, prob = NULL)
 . . .
```

Loading data from a .csv file

Examples in the form of data instances can be easily stored in a `.csv` text file – rows are instances, columns are features. The whole table can be directly loaded into R using the function `read.table()`.

```
> examples <- read.table("wsd.development.csv", header=T)
>
```

Also, you will need to set your working directory, e.g.

```
> setwd("/home/martin/ESSLLI2013/data")
> list.files()
[1] "wsd.attributes.ods"
[2] "wsd.development.csv"
>
```

Data frames in R

The object ‘`examples`’ is a data frame

```
> str(examples)
'data.frame': 3524 obs. of  21 variables:
 $ SENSE: Factor w/ 6 levels "cord","division",..: 1 1 1 1 ...
 $ A1   : int  1 0 0 0 0 0 0 1 1 0 ...
 $ A2   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ A3   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ A4   : int  0 0 0 0 0 0 0 0 0 0 ...
. . .
```

Data frame in R

- is a 2-dimensional data structure
- is a list of vectors of the same length
- vectors in a data frame can be of different types

Homework 1.2

1. Download and install R on your computer.
   Use the CRAN archive available at `http://www.r-project.org/`.

2. Run R.

3. Download the data set `wsd.development.csv` and load it into R.

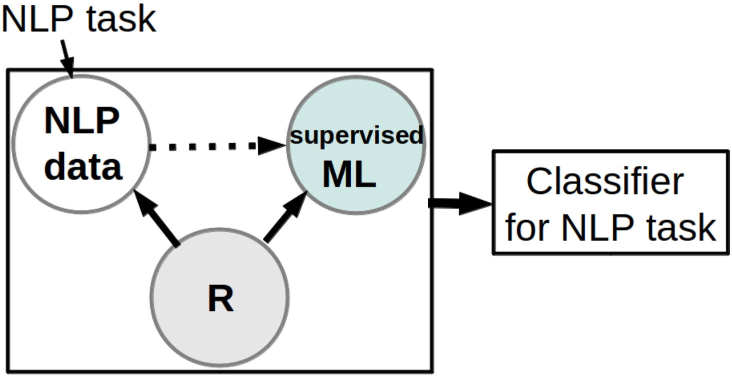
4. Look at the data. Use the function `str()`. What is the most likely `SENSE` in this data set?

   – **Hints**:

   - Use the function `table()` – see `help(table)`
   - Then try `table(examples$SENSE)`
   - Try also `sum(table(examples$SENSE))`

Summary of Day 1

Supervised ML in NLP
– the case of classification task


NLP task
NLP data
supervised ML
R
Classifier for NLP task