# *Methods for Intelligent Systems*

## *Lecture Notes on Clustering (IV)*

Davide Eynard

eynard@elet.polimi.it

Department of Electronics and Information

Politecnico di Milano

# Course Schedule

| Date | Topic |
|---|---|
| 28/03/2006 | Clustering Introduction & Algorithms (I)<br>(K-Means, Hierarchical) |
| 11/04/2006 | Clustering Algorithms (II)<br>(Fuzzy, SOM, Gaussians, PDDP) |
| 16/05/2006 | How many clusters?<br>(Evaluations and tuning) |
| 20/06/2006 | Monography on Text Clustering (I)<br>(+ exercises) |
| 21/06/2006<br>(14.15 AM2) | Monography on Text Clustering (II) |

# Before we begin

*WARNING:*

*Despite its title, this lesson won't be about Text Clustering. Well, at least it won't be* ***only*** *about Text Clustering: it will deal with Internet, World Wide Web, Searching techniques, Web Robots, Regular Expressions, Databases and many of the things you always use but usually never stop to look at. You will learn how to crawl websites, how to extract only the information you're interested in from Web pages, how to build your own search engine at home.*

*And well, yes, how to cluster text too.*

# Why, oh why?

Because:

- Many of the algos you've studied are ok for TC too...
  - ... but you don't know where to start
- Text Clustering is only a technique: how can you use it?
  - You first need to collect data
  - And to do this, you first need to understand how the Web works
- Some Web basics will help you in different ways
  - You'll be able to extract data for your clustering algos
  - You'll learn how to use the Web for your own advantage, find what you want more easily, have only interesting stuff delivered to you

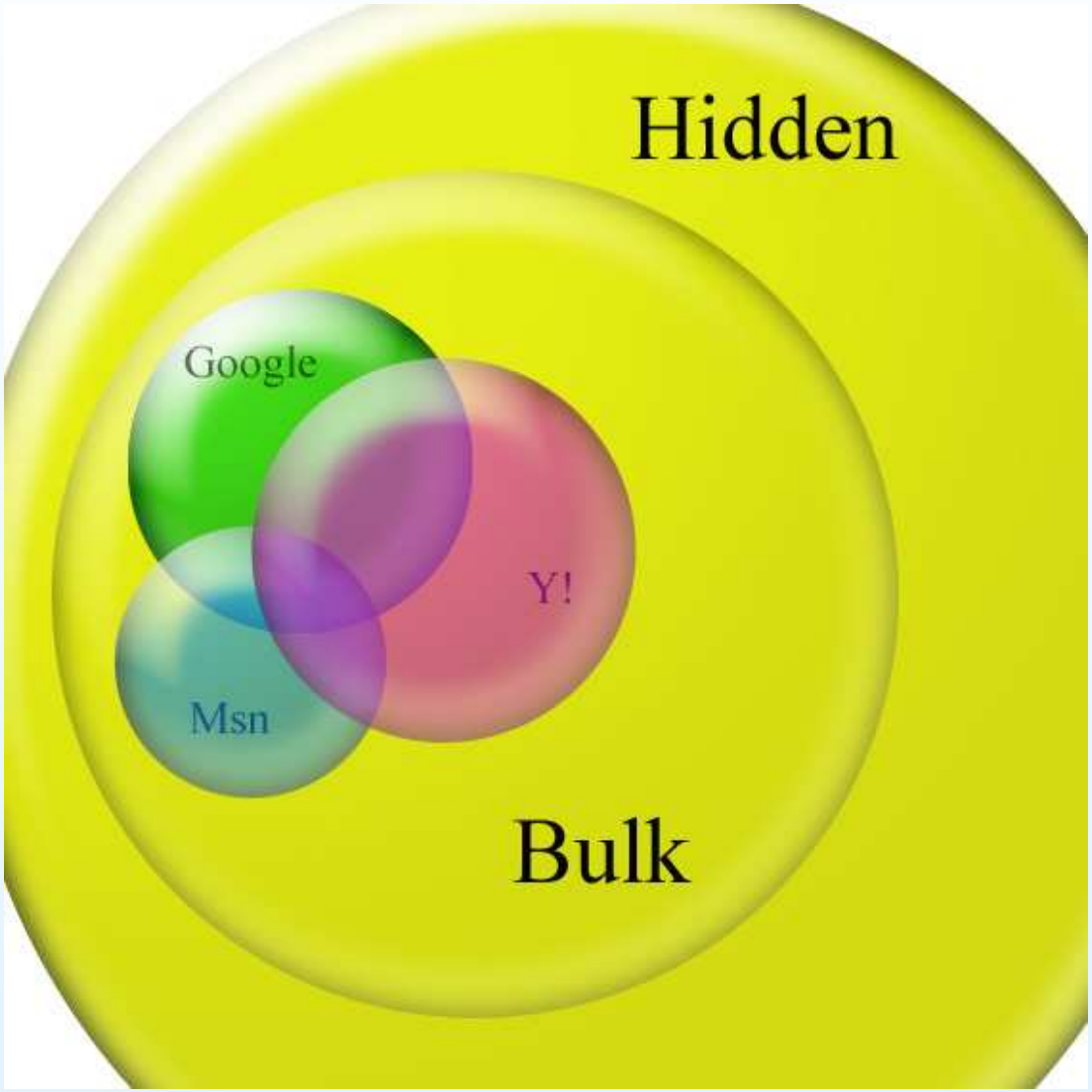
... also, it's quite funny ;)

# Why Text Clustering

Information on the Web increases every day:

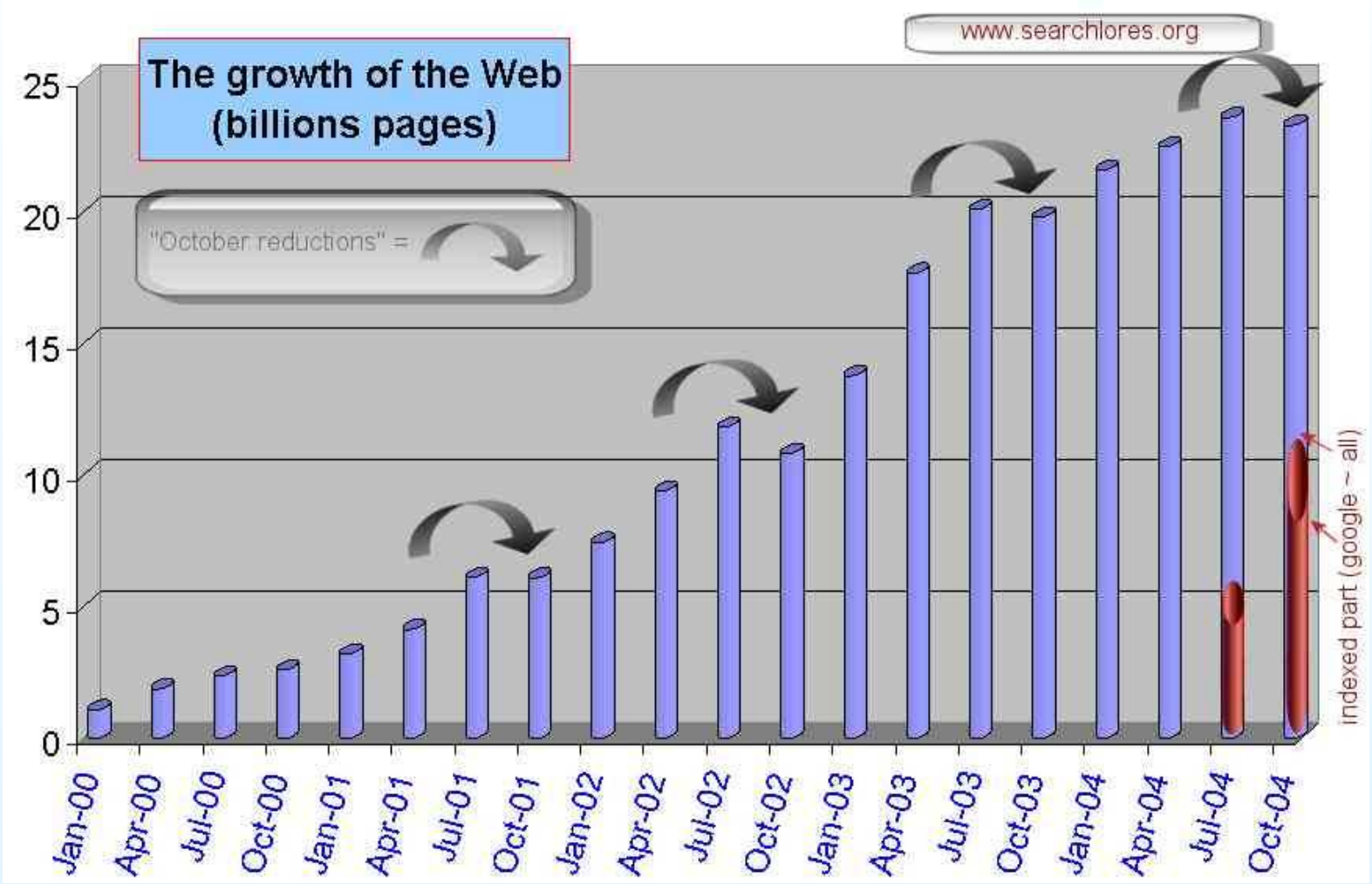
- more than 25 billion indexed by google...
- ... and more than 25 billion are not indexed by any search engine!
- thousands of new pages every day
- thanks to blogs and forums, number increases more and more

# The Structure of the Web

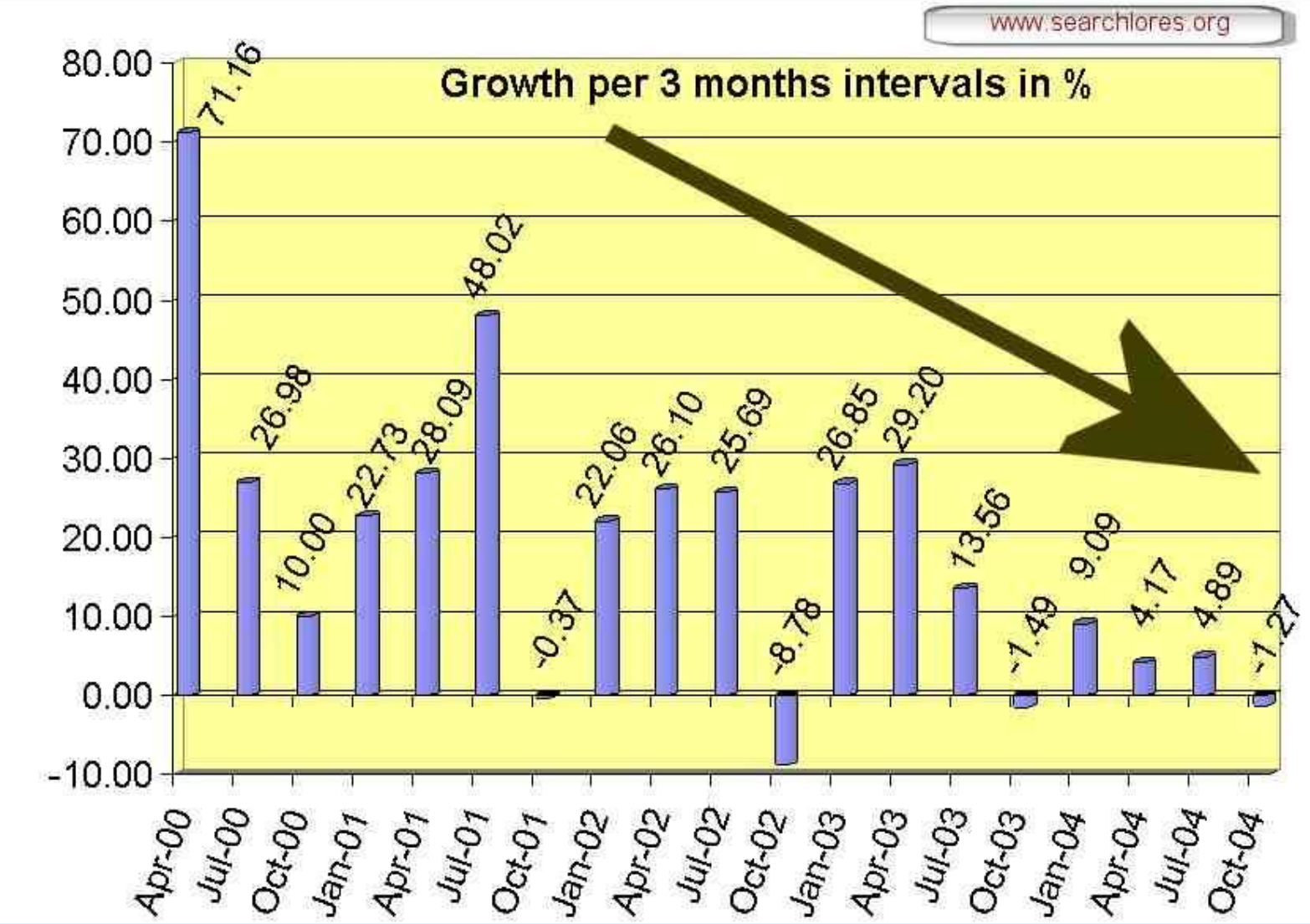



(courtesy of http://www.searchlores.org)

# The Structure of the Web



# The Structure of the Web



# So, what?

We saw the Web is

- Not only the Web (irc, ftp, usenet, etc.)
- Not completely covered by search engines
- Growing very quickly

How can we actually *find* information on the Internet?

- Approach the problem from a "normal user" perspective
- Adapt it for a "search engine" point of view
  - Why is Google better than many other SE?

# How do you browse today?

Default browser (IE for Windoze) means you don’t have the chance to customize many things. As a result, you see only what others want you to see:

- images (often banners)
- popups
- tons of unuseful HTML code (which you don’t see, but you have to download anyway)

# How do you browse today?

You see only what others want you to see, in the way they want you to see it:

- with "active", non accessible contents
- inside fixed-size windows
- following predefined paths (such as in Autogrills, every site has its own *noce di pepe*)

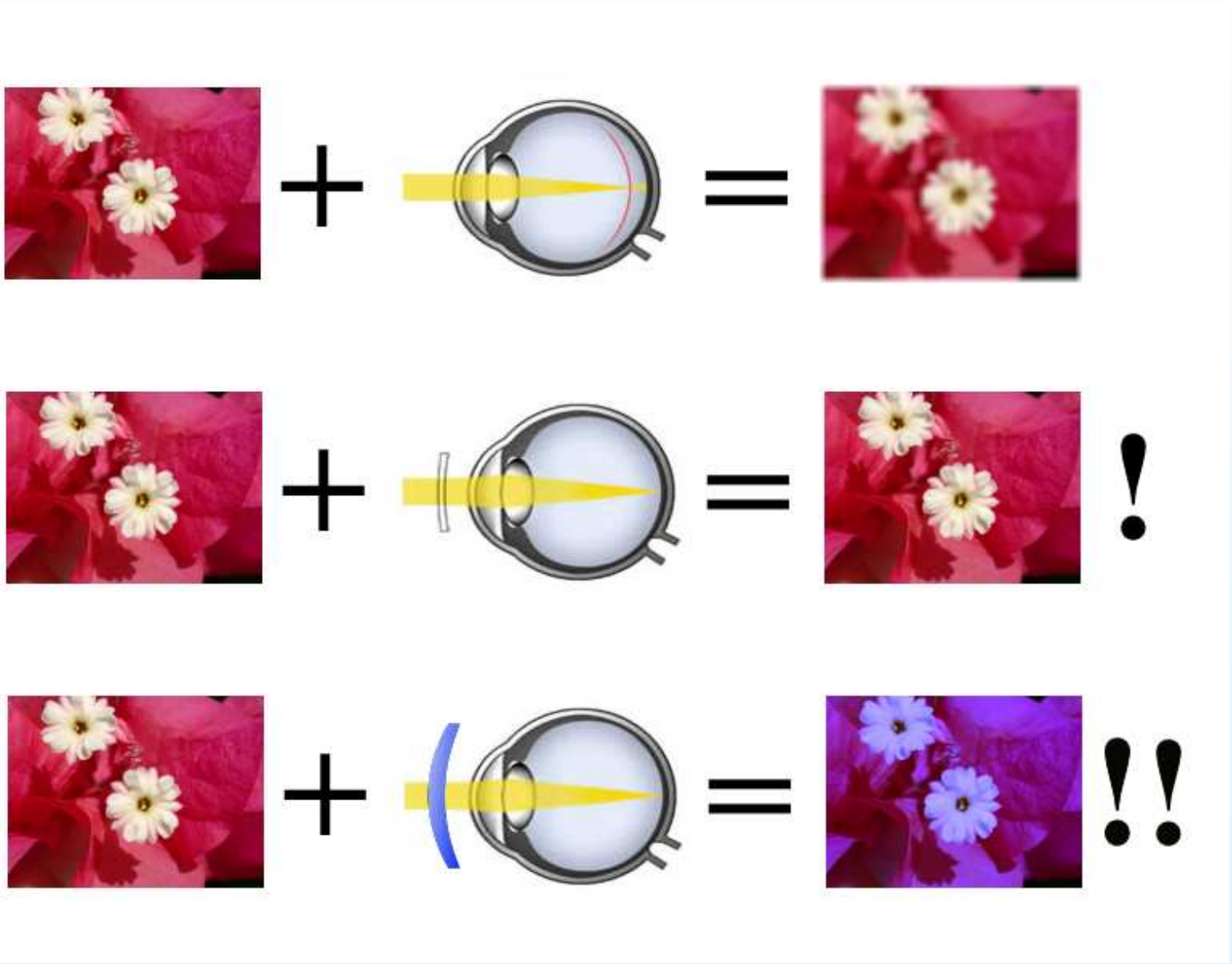
But well, this is what we’re given, right?

## How does it work, instead?

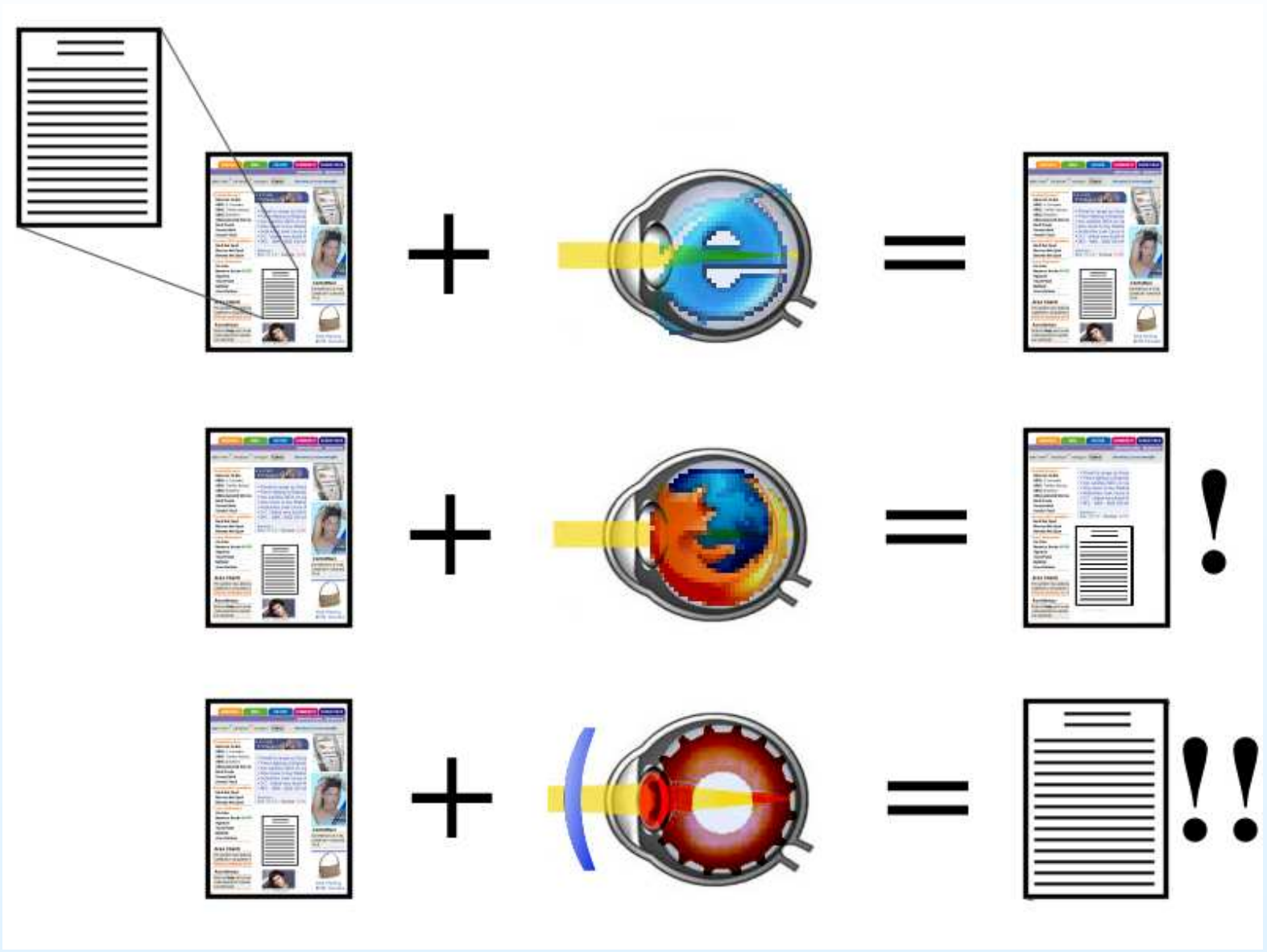
**WRONG!** A PC is not a TV, you can make it do whatever you want, such as:

- downloading only what you want
- showing Web pages the way you like
- collecting and analyzing data for you

# Real Glasses



# Power Glasses



# Techniques and technologies

What are the techniques a user could use to get the best out of the Web?

- Some basic ones
  - alternative browsers
  - leechers and teleporters
  - spiders and scrapers
  - proxy-like softwares
- Some advanced ones
  - learn oneliners with curl, wget, lynx
  - learn how to search
  - learn how to extract data from Web pages
  - learn how to search inside extracted data

# Learn how to search

Learning how to search, you'll also learn something more about how current Search Engines work, and get some ideas for your SE:

- Word-based searches (the "star" example)
- The "index of" trick (and how spammers exploited it, making things harder to find)
- Try different search engines (Teoma - Clustering example)
- Try *clustering* search engines
  - What does *clustering* mean in this case?
- Try folksonomies
- Try blogs and forums

For more info about search strategies, give a look at this (warning: it contains LOTS of examples!)

# Bot Basics

(or: learn how to extract data from Web pages)

- What is a bot?
- What should a bot do for us?
  1. visit a website, following links
  2. extract useful information
  3. work on data (or just save them to allow another app to use them)
- How can I create a bot?
  - tradeoff between performances and complexity
  - any programming language is fine (the faster the better)
  - check you have the libraries you need (http, text parsing etc.)

# Recognizing Web Patterns

- Patterns in presentation/browsing
- Patterns within a website/a class of websites
- Tools: your brain ;-)

In both cases, automatically generated code helps much

# Browsing with bots

Your bots will have to:

- download Web pages
- follow or collect links which satisfy particular conditions (on the tagged text or on the link itself), until a particular depth or forever
- fill forms (!)

## Web Technologies

There are some things you should know to make a well-behaving bot:

- HTTP
  - GET and POST
  - Referer
  - UserAgent
  - Cookie
  - Proxy
- HTML
  - Form
  - Dynamically generated code

I suppose you already know at least the basics of these technologies. If you don’t, you can give a look at this tutorial.

# Some bot examples

Simple bots are not hard to create... here are some examples:

- A really basic bot
- A more advanced one
- The cinema example
- The comics example

# A really basic bot

```
#!/usr/bin/perl
use LWP::Simple;

$content = get ("http://www.google.it");
print $content;
```

- LWP::Simple is the library you need to "get" pages from the Web
- the "get" command is quite straightforward: it downloads an URL and puts its contents inside the variable $content
- the "print" command just prints the page source code on the terminal

# A (slightly) more advanced bot

```
#!/usr/bin/perl
use LWP::UserAgent;

my $ua = LWP::UserAgent->new;
   $ua->agent('NewsBot/0.1');

my $res = $ua->get("http://news.google.it");

if ($res->is_success){
    if ($res->content =~ /berlusconi/si){
        print "Today we're speaking again about Berlusconi!\n";
    }else{
        print "This is a Berlusca-free day!\n";
    }
}else{
    die $res->status_line;
}
```

(note the $ua->agent identification string, needed to download Google News page!)

# The cinema and comics examples

Source code is too long to display, check them here:

- Cinema
- Comics

# Data extraction with regular expressions

In the bots you've seen data extraction is performed with regular expressions:

- Search "perlre" with google
- Hard to read, but very powerful
- Regexp examples

# A complete example

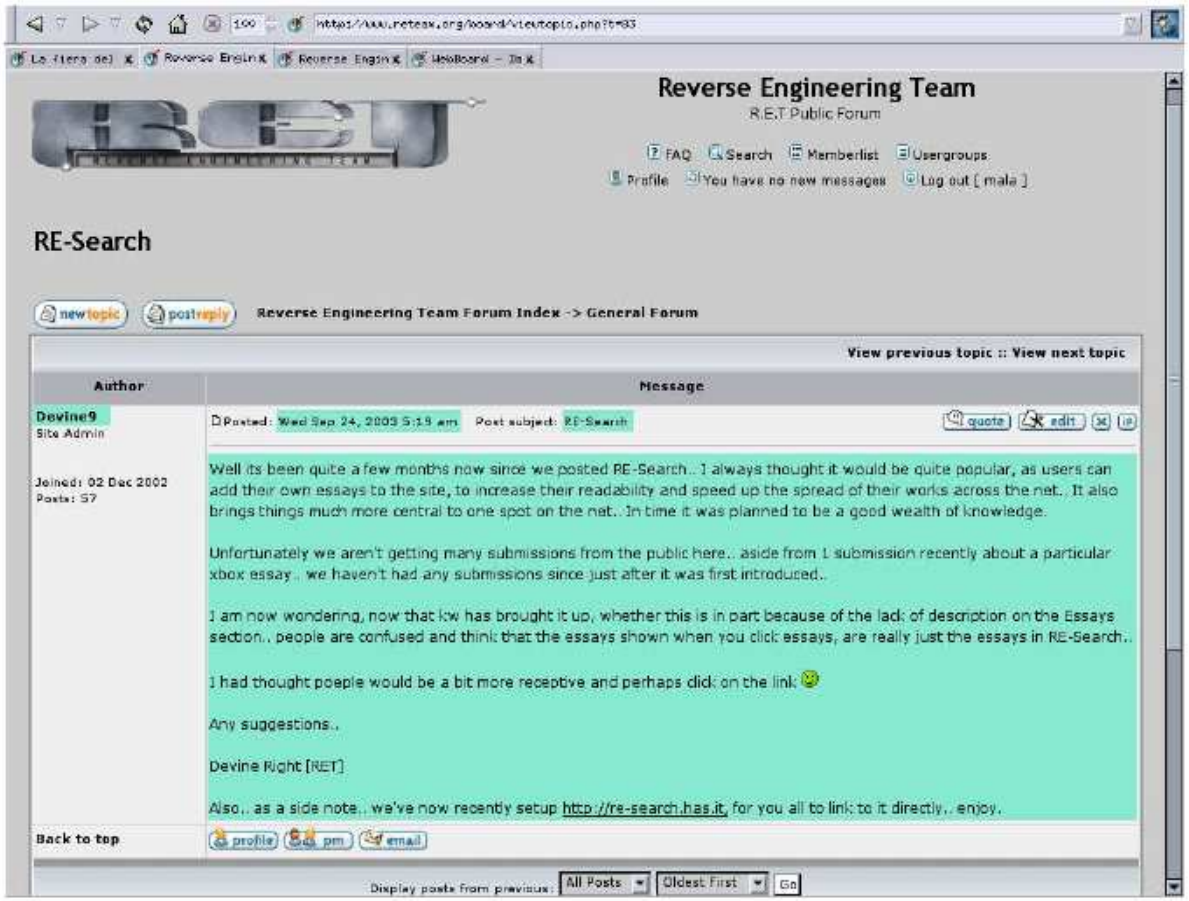
TWO (The Working Offline forum reader) is an old project of mine, which could help you understand how all these technologies work together.

- A component downloads pages from Web forums
- Another one extracts information from them
- Finally, data is normalized and saved inside the DB

Of course it’s free (as in freedom): you can download it from http://two.sf.net

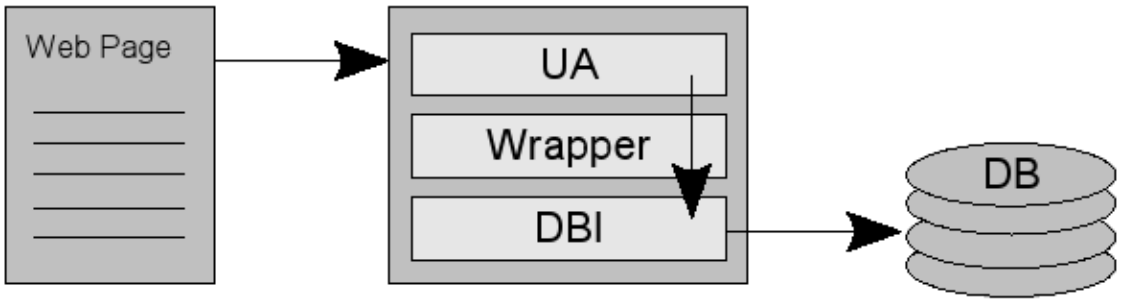
# The idea

**Figure 2.1: To view this page in your browser you have to download about 140KB. The interesting (highlighted) data are less than 1.4KB, that is a 1% ratio!**



# TWO structure

**Figure 4.2: Input component's structure.**



# TWO performances

```
------------------------------
Test started at       22:49:00
Test finished at      00:28:00
------------------------------
Total test time       01:39:00
------------------------------


------------------------------
Downloaded pages          2245
Saved messages           13693
Bytes count           94967139
------------------------------


------------------------------
DB size before test (KB) 3288
DB size after test (KB) 11180
------------------------------
Total data size.(KB).... 7892
------------------------------


------------------------------
Forum data size (KB)     92741
TWO's data size (KB)      7892
------------------------------
Saved space.(KB).........84849
Saved space (perc).........91%
------------------------------
```

```
Pages count: 2244
Bytes count: 94943907
Messages count: 13692
Saving message #17986
----
GET http://board.anticrack.de/viewtopic.php?t=2491
User-Agent: Two/0.01
Cookie: phpbb2mysql_sid=d68ccd982732c219e55cff36ec5fa44f;
Cookie2: $Version="1"



Pages count: 2245
Bytes count: 94967139
Messages count: 13693
Saving message #17985
mala@kami:~/prj/last$
```

```
kami:/var/lib/mysql# du two
3288    two
kami:/var/lib/mysql# du two
11180   two
kami:/var/lib/mysql#
```

# How can we use data from TWO?

We've done *almost* everything we wanted:

- learned how to (automatically) browse websites
- learned how to extract data from web pages
- saved our data inside a database

Now we can work on these data:

- creating statistics
- browsing them offline
- searching inside them

# Two different search approaches

We could allow a search inside documents

- just based on word count
- or checking distance between documents

NOTE: this part has been explained with hand-made drawings and examples. Anyway, you can still find much information on these websites:

- Mining The Web
- Vector space model on Wikipedia
- "A Vector Space Model for Automatic Indexing"

# Bibliography


- http://www.searchlores.org
- http://www.powerbrowsing.org
- G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing"
- As usual, more info on del.icio.us