Extracting data from the web

Donato Summa
Summary

• IaD & IaD methods
• Web Scraping tools
• ICT usage in enterprises
• URL retrieval
Why using Internet as data source?

- *It is a matter of fact that the official statistics whose “monopolistic position” with respect to data capturing is no longer the same as some years ago...*
Internet as data source (IaD)

• Nowadays Internet is simply a part of our society

• Citizens and enterprises increasingly leave behind digital footprints

• By mining these footprints it is, at least theoretically, possible to describe a wide range of socio-economic phenomena in near real-time

• (IaD) to complement or substitute traditional statistical sources.
IaD methods classification

Based on the **actual location of the measurement**, three basic types of IaD-methods can be distinguished:

1) **User centric** measurements that capture changes in behaviour at the client (PC, smartphone) of an individual user.

2) **Network-centric** measurements that focus on measuring properties of the underlying network.

3) **Site-centric** measurements that obtain data from webservers.
Why use IaD methods?

The use of IaD methods has several advantages.

The precise pros and cons of IaD versus traditional methods depend on:

- the specific context
- the specific type of data that is being collected
**IaD methods advantages**

- Allows near real time measurement of data (sharp contrast with traditional data collection methods where measurement can take months or years)
- Allows measuring online phenomena far more accurately than traditional methods
- Allows measuring some online phenomena that cannot be measured with traditional methods
- Significantly lower costs for data collection
- Lowers respondent burden (by using non-intrusive data)
IaD methods challenges

• **Practical feasibility**
  Not every IaD method can be implemented easily

• **Legal robustness**
  The automatic measurement of all kinds of data almost immediately brings up discussions concerning privacy

• **Social acceptance**
  Even if it is legal to use a new method, it is not obvious that the public will accept it
IaD methods challenges

- **Sampling**
  For statistical purposes, the unit of analysis at which the measurement is done is highly relevant

- **Unit of analysis**
  It is always purely technical data (e.g. source code from a website, headers from a data packet, an address in a browser bar)

- **Culture within NSI**
  NSI are typified by a culture focused on data quality and tradition. This makes NSI somewhat reluctant to change their methods of data collection
Site centric measurements

- Measurements that obtain data from web sites
- The most mature IaD method
- Crawlers and Web robots are already widely used in the private sector
- The major disadvantage of the use of automated agents to collect data from websites is their inability to interpret and read texts on webpages the way human agents do
- A combination of human and automated agents seems to be the best of both worlds
Site centric measurements

• From the perspective of the web server there is no difference between a human agent visiting the website and an automated agent.

• The obvious benefit of automated agents is that they can visit a lot of websites much faster than human agents can.

• A drawback is that automated agents lack the great ability of people to interpret messages, and especially to deal with rich information.

• The use of automated data collection has been focused at well-defined basic data.
Automated agents

- Automated agents are pieces of software that emulate human behaviour on web pages in order to achieve some results
- Web pages are text documents written in HTML
- An HTML page that a web browser shows is essentially a tree of nodes
- Different nodes can have a different meaning and can have different content
Automated agents

• As there is no formal agreement on when to use what kind of tree structure, different HTML pages typically have different tree structures.

• To extract the relevant information from the HTML page one can query the tree to look at parts of it, to look at particular nodes, and to retrieve specific information from a node.
The rise of dynamic web pages

• **problem:** dynamic pages are less readable than the traditional static HTML pages

• **cause:** the content generated from an underlying database is usually triggered by a specific event, usually an action initiated by the human visitor of the webpage

• **solution:** automated agents have to be able to mimic the complex behavior of human agents
Web scraping

We can distinguish two different kinds of web scraping:

• **specific web scraping**, when both structure and content of websites to be scraped are perfectly known, and scrapers just have to replicate the behaviour of a human being visiting the website and collecting the information of interest. Typical areas of application: data collection for price consumer indices (ONS, CBS, Istat);

• **generic web scraping**, when no a priori knowledge on the content is available, and the whole website is scraped and subsequently processed in order to infer information of interest: this is the case of the “ICT usage in enterprises”.

Web scraping tools

There are 2 categories of web scraping tools:

• Tools aimed to scrape very specific information of a site (eg. a value in a row of a table in a page)
  - iMacros
  - iRobotSoft

• Tools aimed to scrape the whole content of an arbitrary number of sites (so it is up to you to retrieve any specific information)
  - Nutch
  - HTTrack
iMacros

• SW for Web Automation and Web Testing
  • Commercial software with a limited free version
• enables users to capture and replay web activity such as:
  • form testing
  • uploading or downloading text and images
  • importing and exporting data to and from web applications

• allows speeding up the acquisition of textual information on the web
iMacros

- can be used with the help of programming languages and scripting (e.g. Java, JavaScript)

- iMacros tasks can be performed with the most popular browsers

- APIs are available, allowing to interface and integrate iMacros with other applications and legacy environments
iMacros

Internet Macros records the web activity in an easy to understand plain text macro language. All commands are fully documented.
IRobotSoft

- a visual Web automation and Web scraping software using HTQL (a simple language for querying and transformation of HTML Web documents)
  - Available as free software
- provides an internal scheduler that can be set up by users to have robots running at a particular time or frequency
- strength: automated Web robot generation technology by recording some user Web explorations from an embedded browser
- weakness: windows-only, difficult to use
IRobotSoft

- for common Web users who have very limited programming skills

- but also for skilled programmers due to its powerful data manipulation language that supports complex Web computations

- Not very user friendly
IRobotSoft

Navigation Task Name: Run Rename

Action
Go to URL: http://www.ncbi.nlm.nih.gov
Submit form with 'pubmed-key'
Get a list of links like 'Carrageen E'.
Title: Extract data like 'Does minor'
Journal: Extract data like '1': Spine
Author: Extract data like 'Carrageen'
Abstract: Extract data like 'STUDY'

My Robots

Local Robots
baidu.irc
checkmains.irc
hotomot.irc
pubmed.irc

Recent Robots
C:\Downloads\Internet\irobot-eval\pubmed.irc

Search for all pubmed and save the abstract.
It also follows all to
to download a list of.
to download a single.
Advantages:
1) Don't need to copy
2) Abstracts are saved
other word processors
3) Build abstract col
file.
It includes:
- Go to URL: http://...
“Specific web scraping” tools

• Other solutions are available (both free and proprietary) that can be evaluated before making a decision
  • just search web scraper on Google

• It’s up to you to decide which SW best suits your needs (and/or your coding skills)

• Other examples:
  • Scraper (Chrome plugin)
  • https://www.import.io/
Recent case studies in NSI

- Harmonised Index of Consumer Prices (Istat)
- Consumer Price Index (CBS)
- **ICT usage in enterprises** (Istat)
ICT usage in enterprises

• The Community survey on ICT usage and e-commerce in enterprises (in short, ICT in enterprises) aims at producing information on the use of Internet and other networks by enterprises for various purposes (e-commerce, e-skills, e-business, social media, e-government, etc.)

• Data are historically collected by means of paper or web questionnaire
ICT usage in enterprises

• The web questionnaire is used to collect information on the characteristics of the websites owned or used by the enterprises:

<table>
<thead>
<tr>
<th>Use of a Website or Home Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>B7.</strong> In January 2013, did your enterprise have a Website or Home Page? <em>(Filter question)</em></td>
</tr>
<tr>
<td><em>(&gt; go to B9)</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>B8.</strong> In January 2013, did the Website or Home Page have any of the following?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Online ordering or reservation or booking, e.g. shopping cart</td>
</tr>
<tr>
<td>b) A privacy policy statement, a privacy seal or certification related to website safety</td>
</tr>
<tr>
<td>c) Product catalogues or price lists</td>
</tr>
<tr>
<td>d) Order tracking available online</td>
</tr>
<tr>
<td>e) Possibility for visitors to customise or design the products</td>
</tr>
<tr>
<td>f) Personalised content in the website for regular/repeated visitors</td>
</tr>
<tr>
<td>g) Advertisement of open job positions or online job application <em>(Optional)</em></td>
</tr>
</tbody>
</table>
ICT usage in enterprises

• During 2013 Istat began to explore the possibility to use web scraping techniques, associated, in the estimation phase, to text and data mining algorithms

• The aim is to replace traditional instruments of data collection (in order to reduce respondent burden) and estimation, or to combine them in an integrated approach
ICT usage in enterprises

• In a first phase, the aim of the experiment was to predict values of questions from B8a to B8g using machine learning techniques applied to texts (text mining) scraped from the websites.

• Particular effort was dedicated to question B8a (“Web sales facilities” or “e-commerce”).
ICT usage in enterprises

- The scraping task done in this work goes a step further with respect to previously described experiences.
- We tried a data collection *without any assumption* on the structure of the websites and by providing the *ability to scale up* to a huge number of them.
ICT usage in enterprises

• Excluding the analysis part of the job we can distinguish 4 phases:

Crawling  Scraping  Indexing  Searching
ICT usage in enterprises

- **Crawling**: a Web crawler (also called Web spider or ant or robot) is a software program that systematically browses the Web starting from an Internet address (or a set of Internet addresses) and some pre-defined conditions (e.g., how many links navigate, the depth, types of files to ignore, etc.).
ICT usage in enterprises

- **Scraping**: a scraper takes Web resources (documents, images, etc.), and engages a process for extracting data from those resources, finalized to data storage for subsequent elaboration purposes.
ICT usage in enterprises

• **Indexing/Searching**: searching operations on a huge amount of data can be very slow, so it is necessary (through crawler) to index contents.

• Analysers tokenize text by performing any number of operations on it, which could include: extracting words, discarding punctuation, removing accents from characters, lowercasing (also called normalizing), removing common words, reducing words to a root form (stemming), or changing words into the basic form (lemmatization). The whole process is also called “tokenization”, and the chunks of text pulled from a stream of text are called “tokens”.
ICT usage in enterprises

Scraping systems evaluated:

• The Apache Stack: Nutch/Solr

• HTTrack

• JSOUP
The Apache Stack: Nutch / Solr

- Apache Nutch is a highly extensible and scalable open source web crawler

- it facilitates parsing, indexing, creating a search engine, customizing search according to needs, scalability, robustness, and scoring filter for custom implementations

- built on top of Apache Lucene and based on Apache Hadoop, Nutch can be deployed on a single machine as well as on a cluster, if large scale web crawling is required
  - **Apache Lucene: a popular Java text search engine library**
The Apache Stack: Nutch / Solr

• Apache Solr is an open source enterprise search platform that is built on top of Apache Lucene
  • [https://lucene.apache.org/solr/](https://lucene.apache.org/solr/)
• It can be used for searching any type of data (in this context it was used to search web pages)
  • Indexes not only text but also binary documents
• Its major features include full-text search, hit highlighting, faceted search, dynamic clustering, database integration, and rich document handling.

• Providing distributed search and index replication, Solr is highly scalable
The Apache Stack: Nutch / Solr

• Technological Stack

• Apache Platform:
  - Nutch: Crawler & Scraper
  - SOLR: Indexing & Searching

• Steps:
  - Configure and launch Nutch
  - Configure and launch SOLR
  - Access SOLR index via LUCENE API for further processing
HTTrack

- HTTrack is a free and open source software tool that permits to “mirror” locally a web site, by downloading each page that composes its structure.
  - [http://www.httrack.com](http://www.httrack.com)

- It is a web crawler and an offline browser that can be run on several operating systems.

- HTTrack’s strength points are ease of use, fine parameters configurability.

- It can be run via graphical user interface or in batch mode via command line.
HTTrack
JSoup

• JSOUP is an open source Java library for working with real-world HTML
  • [http://jsoup.org](http://jsoup.org)

• It provides a very convenient API for extracting and manipulating data, using the best of DOM (Document Object Model), CSS (Cascading Style Sheets), and jQuery-like methods.

• More in detail this tool allows:
  • scraping and parsing HTML from a URL, file or string
  • finding and extracting data, using DOM traversal or CSS selectors
  • manipulating the HTML elements, attributes, and text
Tested platforms

- Virtual machines
- Workstation
- Server
- Big data sandbox
- CINECA
Big Data Sandbox

• In 2014 a “Sandbox” environment has been created, with support from the Central Statistics office (CSO) of Ireland and the Irish Centre for High-End Computing (ICHEC).

• It provides a technical platform to load Big Data sets and tools.

• the time spent for a Nutch job (the same executed on a single server) using the sandbox was reduced by 73.41%
Cineca PICO

- Cineca is a non profit Consortium, made up of 69 Italian Universities and 3 Institutions.

- SCAI (SuperComputing Applications and Innovation) is the High Performance Computing department of CINECA, the largest computing centre in Italy and one of the largest in Europe.

- PICO is a super-computer in SCAI (74 nodes)

- The results were encouraging (-90% time spent)
Common scraping problems

• Scrapers can be blocked, as by design some parts of the sites were simply not made accessible by software programs (e.g., CAPTCHA usage)

• Usage of some technologies like Adobe Flash simply prevents from accessing contents because they do not expose text directly

• Robots.txt: a file that is stored on the web server that specifies the policies about scraping accessibility

• Network problems
# Efficiency features of the 3 systems

<table>
<thead>
<tr>
<th>Tool</th>
<th># websites reached</th>
<th>Average number of webpages per site</th>
<th>Time spent</th>
<th>Type of Storage</th>
<th>Storage dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nutch</td>
<td>$7020/8550=82,1%$</td>
<td>15,2</td>
<td>32,5 hours</td>
<td>Binary files on HDFS</td>
<td>2,3 GB (data)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5,6 GB (index)</td>
</tr>
<tr>
<td>HTTrack</td>
<td>$7710/8550=90,2%$</td>
<td>43,5</td>
<td>6,7 days</td>
<td>HTML files on file system</td>
<td>16, 1 GB</td>
</tr>
<tr>
<td>JSOUP</td>
<td>$7835/8550=91,6%$</td>
<td>68</td>
<td>11 hours</td>
<td>HTML ADaMSOFT compressed binary files</td>
<td>500MB</td>
</tr>
</tbody>
</table>
ICT usage in enterprises

At the moment we are testing an ad hoc solution based on:

Jsoup
Crawler4j [https://github.com/yasserg/crawler4j](https://github.com/yasserg/crawler4j)
+
Apache Solr
ICT usage in enterprises : text mining

• 2013 and 2014 rounds of the survey have both been used in the experiment.

• For all respondents declaring to own a website, their website have been scraped, and collected texts submitted to classical text mining procedures in order to build a “matrix terms/documents”.

• Different learners have been applied, in order to predict values of target variables (for instance, “\texttt{e-commerce (yes/no)}”) on the basis or relevant terms individuated in the websites.

• The relevance of the terms (and consequent selection of 1,200 out of 50,000) has been based on the importance of each term measured in the correspondence analysis

• 2013 data have been used as “\texttt{train}” dataset, while 2014 data have been used as “\texttt{test}” dataset.
ICT usage in enterprises: text mining

The performance of each learner has been evaluated by means of the usual quality indicators:

• **accuracy**: rate of correctly classified cases on the total;

• **sensitivity**: rate of correctly classified positive cases on total positive cases;

• **specificity**: rate of correctly classified negative cases on total negative cases.
# ICT usage: e-commerce prediction

<table>
<thead>
<tr>
<th>Learner</th>
<th>Quality Indicators for e-commerce</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>GLM (Logistic)</td>
<td>0.69</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.79</td>
</tr>
<tr>
<td>Neural Network</td>
<td>0.70</td>
</tr>
<tr>
<td>Boosting</td>
<td>0.67</td>
</tr>
<tr>
<td>Bagging</td>
<td><strong>0.82</strong></td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.75</td>
</tr>
<tr>
<td>LDA</td>
<td>0.66</td>
</tr>
<tr>
<td>RPART (Tree)</td>
<td><strong>0.82</strong></td>
</tr>
</tbody>
</table>
URLs retrieval

• In order to be able to crawl and scrape content on firm websites you need first of all to know the list of that website addresses

• This consideration leads us to . . .
The initial problem

- During the ICT in enterprises case study it was necessary get the full list of enterprises' websites
- We tried several traditional ways in order to acquire the full list without success
The starting point

- A list with about 200K enterprise records containing several information:
  - ISTAT code (primary key)
  - Official name of the enterprise
  - Certified mail
  - VAT
  - City
  - Province
  - Telephone number
Get URLs

3 starting sources:
ICT2013 / 2014 / 2015 (Survey data)
Consodata2013 / 2014 / 2015 (Administrative data)
Asia (Business register)

For each enterprise we have a minimum of 0 and a maximum of 6 URL sometimes different each other but we have to select only one so:

If we have only 1 url we take that

If we have more than 1 url:
The most recent survey URL is the best
The most recent administrative URL is better than other URL that we can have using other available sources
Get URLs

We applied a validation algorithm on official urls:

- Removal of non valid characters (e.g. commas, spaces, @, ecc.)
- Removal of only numerical url (e.g. 3456)
- E-mail addresses used instead of web addresses
- Free text (e.g. “we don’t have a site”, “nothing”, “n.d.”, ecc.)

Automatic corrections such as:

- “http” substituted with “http”
- “ww” substituted with “www”
- “\” substituted with “//”

Finally some manual corrections such as:

- www.officinapellegrini.ne changed in ww.officinapellegrini.net
Get URLs

Starting from a list of about 200,000 enterprises records we retrieved 81451 “official” urls

There are still about 119K missing urls (if every enterprise has a site)

A possible solution:

An automated procedure to be improved with Crowd sourcing techniques
The automated URL retrieval procedure

- Idea: obtain an enterprise web site (if it exists) starting from a search of the enterprise name in a search engine
The automated URL retrieval procedure

1. Input from existing sources (Register plus other archives): denomination, address, telephone number, fiscal code, ...

2. For each enterprise in the target population:
   a) Introduce the denomination into a search engine
   b) Obtain a list of the first k resulting web pages
   c) For each one of these results, calculate the value of binary indicators. For instance:
      o the URL contains the denomination (Yes/No);
      o the scraped website contains geographical information coincident with already available in the Register (Yes/No);
      o the scraped website contains the same fiscal code in the Register (Yes/No);
      o the scraped website contains the same telephone number in the Register (Yes/No);
   d) Compute a score on the basis of the values of the above indicators.
The automated URL retrieval procedure

3. On the subset of enterprises for which the URL is known (training set), model the relation between the binary indicators plus the score, and the success/failure of the found URL.

4. Apply the model to the subset of enterprises for which the URL is not known, in order to decide if the found URL is acceptable or not.
List of names and ids of enterprises

ICTScraper

File with urls retrieved from Bing

FromTxtToCSV

Java

List of URLs

CSV

Eurostat
List of URLs

Filters on sites such as paginegialle

AssignScores

Final result

nutch

Apache Solr

Java

Excel

CSV
Tested search engines

• Google
• Istella
• Bing
• Yahoo
We tried to execute a batch of 10000 automatic queries without success and we tested the restrictiveness of Google policy.

It is not possible to make automatic massive queries for free.

We tried also within their network but without success.

We decided to discard Google while these problems persist.
istella

- An Italian search engine focused on the Italian establishment and way of searching
- Idea: for this particular target could be better than Google
- We executed with success 10000 automatic queries in 1:45 hours without any restrictions
- Search on 7141 enterprise with a known site
- For each enterprise name we collected the first 10 results from the search engine and compared them with the known site
- Only 50% of matches and not always in the first positions
• We executed with success 10000 automatic queries in less than 2 hours without any restrictions

• Same tests conducted on Istella with a little better results

• From about 50% to about 65% and also in this case the matching results links were not always in the first positions
Like Google, Yahoo’s policies stopped us after about 800 consecutive automatic queries.

However the percentage of matching links for the small amount of enterprise names tested was very similar to Bing.

We decided to use Bing due to its non blocking policies and greater performances in terms of “official web site” finding.
Used software

- ICTScraper
- FromTxtToCsv (now included in ICTScraper)
- Apache Nutch (now JCrawler)
- Apache Solr
- AssignScore
- MS Access & Excel
ICTScraper

• Custom Java program used to make queries

• Inputs:
  - a file containing the ordinated list of names
  - a file containing the ordinated list of IDs

• Output:
  - one txt file for each enterprise containing the results links
FromTxtToCsv

- Custom Java program used to generate a single CSV file starting from a number of structured txt files obtained by ICTScraper

- The CSV file will be used to easily load data in a DB for further processing

- We load the CSV in MS Access in order to produce a suitable seed.txt file for Nutch, that is the initial urls list to crawl
• Crawl, Fetch and Parse starting from the url list in seed.txt
• It indexes the content parsed by Nutch and allows searching on scraped content (back end functionalities)
MS Access & Excel

• In this experimental phase of the project in order to speed up some operations of analysis and construction of the input to various SW we used MS Office suite of products

• Maybe in the near future we can think to substitute these SW with other more specific or more powerful (eg. Oracle)

• In general the whole process still contains some manual operations that it is better to automate via some custom SW
AssignScores

• The last step before analysis operations

• For each link scraped by Nutch/JCrawler the SW generates a vector of numbers

• Every position in the vector contains a number that means “a specific characteristic in the web page was found or not” (eg. presence of the telephone number)
  • Example on slide 77

• We multiply each number with a specific coefficient and sum all the results in order to obtain a score for that link
The experiment

We applied the automatic procedure on 20000 enterprises with a known web site

AssignScore algorithm:

- Simple url (e.g. www.name.com)
- VAT
- city
- province code (e.g. NA)
- link position (e.g. 2)
- telephone number
- url similarity with the name of the enterprise
- presence of words contained in the name of the enterprise in the web page text
- Pec (certified mail)
The experiment

For each element:

- we calculated the confusion matrix that would have been obtained if we used just that element for classification purposes

- based on that confusion matrix we computed standard performance measures:
  - precision
  - recall
  - F-measure

\[
\text{Precision} = \frac{tp}{tp + fp} \quad \text{Recall} = \frac{tp}{tp + fn}
\]

\[
F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}
\]

\(tp\): true positive
\(fp\): false positive
\(fn\): false negative
The experiment

Then we assigned as raw weights to elements the f-measures of the corresponding confusion matrixes.

Lastly we have normalized the raw weights so that the sum of the final (adjusted) weights is 1000.

In order to assign a final score to a link we summed the normalized weights of those elements that were actually present in the vector returned by AssignScores.
The experiment

Eg. for the following link vector (102101) we have:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
<th>Position</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephone</td>
<td>1</td>
<td>1</td>
<td>0 +</td>
</tr>
<tr>
<td>Simple URL</td>
<td>0</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td>Link position</td>
<td>2</td>
<td>233</td>
<td>233 +</td>
</tr>
<tr>
<td>VAT</td>
<td>1</td>
<td>248</td>
<td>248 +</td>
</tr>
<tr>
<td>City</td>
<td>0</td>
<td>0</td>
<td>0 +</td>
</tr>
<tr>
<td>Province code</td>
<td>1</td>
<td>124</td>
<td>124 =</td>
</tr>
</tbody>
</table>

605 (obtained score)
Obtained results

• For each firm considered in the test set we compared the most probable official URL found by the procedure with the official URL that we already knew.

• In particular we compared only the domain part of the url (eg. www.rossi.it/aboutUs )

• We foud exact match in 64% of the cases
Considerations

The real success percentage is probably bigger than obtained as sometimes the official web site that we know is not correct because:

- it is outdated (the url is changed)
- it does not exist anymore
- wrong domain (e.g. “rossi.it” instead of “rossi.com”)
- it is only an information page with contacts (paginegialle)
- it is the site that sells the products of the enterprise (e.g. mediaworld.it)
- it is the site of the mother company (franchising enterprises)

Probably in this cases Bing and our algorithm find the correct site but we consider it uncorrect because different from the one we know
Crowdsourcing URL retrieval approach

• Another possible solution for selecting the right url of a firm from a list of probable urls is to use a Crowdsourcing platform.

• We are planning to use the Crowdsourcing approach in order to obtain the correct website for those enterprises that obtained a score with an insufficient precision.
Crowdsourcing

It is the process of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, especially from an online community, in a nutshell:

Huge list of Little problems

Community of workers

Huge list of solutions

A kind of human Hadoop!
CrowdSearcher

- A web platform developed by Politecnico di Milano, http://crowdsearcher.search-computing.it/home
- It is used for designing, deploying, and monitoring crowd-based applications on top of social systems, including social networks and crowdsourcing platforms.
Crowdsourcing Task objectives

- Link the most probable official URL from a list of URLs to a given firm name.
- Do this operation for a very long list of firm names in a reasonable amount of time.
Crowdsourcing Task design features

• A group of (100 – 200) identifiable volunteer workers

• For each firm the user has to select the most probable official url from a list of (2 – 10) proposed urls
  - For each proposed url the system shows to the user the webpage
  - The url are sorted by a score computed in a previous step
  - The first choice proposed is the default “select an URL”
  - The last choice proposed is “none of the previous”
  - “offline” choice

• Before the final submission the system
  - Checks if there are still “select an URL” selections and invite the user to make a decision
  - Acquires the data
Crowdsourcing

In the next months:
- Selection of the appropriate URLs set to submit
- Selection of the workers
- Launch of the Task

For now it is still a work in progress.
Thank you for your attention!