Data Collection

Andrea Esuli

Data Collection

Data collection is a crucial step of the process, since it determines the knowledge base on which any successive process will work on.

- Identify the information need
  - I want to identify the most cited aspects of hotels from reviews written by their customers
- Find the source/sources
  - Social platforms?
  - Review websites?
  - Questionnaires directly sent to customers?

Data Collection

Data collection is a crucial step of the process, since it determines the knowledge base on which any successive process will work on.

- Set up the data collection method
  - API, scrapers, questionnaires...
  - Is it legal to get this data?
  - Is my use of the data legal?
- Get the data
  - How much data do I need?
  - How much resources (space, network, time, money) will it take?
  - Monitor the collection process.

Data Collection

Data collection is a crucial step of the process, since it determines the knowledge base on which any successive process will work on.

- Inspect the data
  - Does the data contains the information I need?
  - Is the data complete?
  - Is the data clean from noise?
  - Is the data correct?
  - Is the data correctly linked to the entities I'm interested in?
- Prepare data for successive steps
  - Would some preprocessing help the successive steps?

Data Collection

Depending on problems and goals, there are many possible data sources.

Web based:

- Online survey, e.g., SurveyMonkey, Google survey, Google forms.
  - Survey services offer demographic targeting
- Web feeds, e.g., RSS, Atom.
  - Most news companies offer a RSS version of their content organized by topic.
- Social networks' APIs. E.g., twitter, facebook, and many other.
- Archives. E.g., Reddit, archive.org.
- Custom web crawling and scraping. E.g., Scrapy.

Data Collection

Companies and organizations may accumulate information from other sources:

- feedback channels (email, telephone, sms, handwritten)
- note in customer profiles

...or more traditional questionnaires and interviews:

- Computer-assisted telephone interviewing (CATI),
- Automated Computer Telephone Interviewing (ACTI)

in some cases digitalization of text is required (handwritten text recognition, speech to text).

Ready made scrapers

Many custom scrapers are available for a number of review platforms:

- TripAdvisor
- Amazon
- Steam
- AirBnB
- Booking

Scraping the Web

The requests and urllib packages can be used to make request for web pages and to build a custom scraper.

When scraping a server:

- be correct with respect to the robots.txt file
- be fair with sending requests to server.
  - Always put a pause between one request and the next one.

Scraping is typically a matter of finding a url scheme and collecting urls from index pages or search form results.
Getting urls and actual text requires parsing the HTML content returned by the server.

Many tools provide crawl and scrape methods at various abstraction levels.

Extracting (clean) text from the Web

The BeautifulSoup package implements methods to navigate, modify, and extract data from HTML and XML data.

It can be used on Web pages to extract information of interest.

```
from urllib import request
url = "http://www.esuli.it"
response = request.urlopen(url)
html = response.read().decode('utf-8')
```

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html, "html5lib")
```

```
soup.get_text()
```

← *simple way or* ↓ *more control*

```
[''.join(s.findAll(text=True))for s in soup.findAll('p')]
```

Twitter API

Twitter publishes a rich API to explore its network structure and get content.

The API provides a number of endpoints:

- Sampled stream: get a random 1% of the global tweet stream, discover what's going on.
- Filtered stream: get a query-specific stream of tweets, collect task-/topic-specific data.
- Recent tweet search: query tweets up to a week back in time, expand your data back in time after you identify a relevant topic.
- Lookup a specific tweet or user to get more information, enrich your data.

Twitter API

The API requires a registration as a developer.

Applications are defined as projects, and each have secret keys that enable access to endpoints.

Free use of API has limitation on the number of queries in time.

Endpoints can be accessed by simple HTTP requests or using libraries available for many languages.

Data annotation

Labeling data

When dealing with supervised tasks (annotation, classification), supervised information is likely to be not available for the task of interest, e.g.:

- sentiment of tweets is not marked explicitly
- spam mail tries to hide its nature
- aspects discussed in a review are defined implicitly by the attributes cited
- entities are not marked in text

Supervised information must then be produced by **annotators** that make such information **explicit** so as to produce train data for ML algorithm.

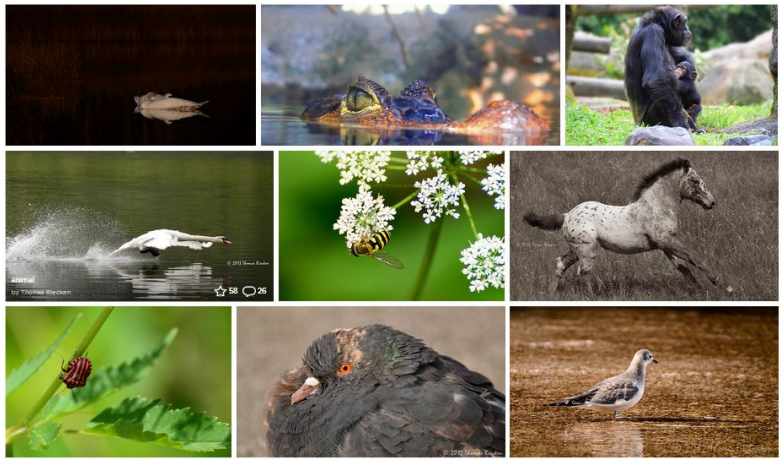
Checking data labeling quality is important, especially in the case data is labeled from scratch specifically for the task.

Garbage In - Garbage Out

The **quality** of the **output** of any supervised learning process **is limited by** the **quality** of the supervised information fed in **input**.

$\Phi(i)$ = cat

$\Phi(i)$ = not cat



Garbage In - Garbage Out

The **quality** of the **output** of any supervised learning process **is limited by** the **quality** of the supervised information fed in **input**.

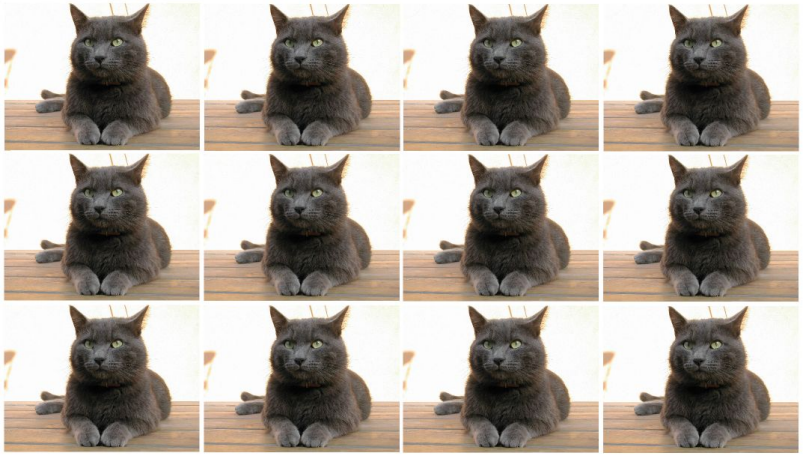
$\Phi(i)$ = cat

$\Phi(i)$ = not cat

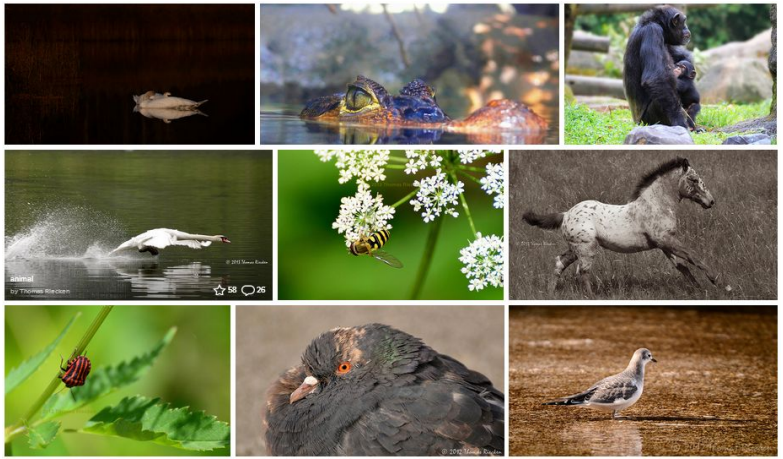
Garbage In - Garbage Out

The **quality** of the **output** of any supervised learning process **is limited by** the **quality** of the supervised information fed in **input**.

$\Phi(i)$ = cat



$\Phi(i)$ = not cat

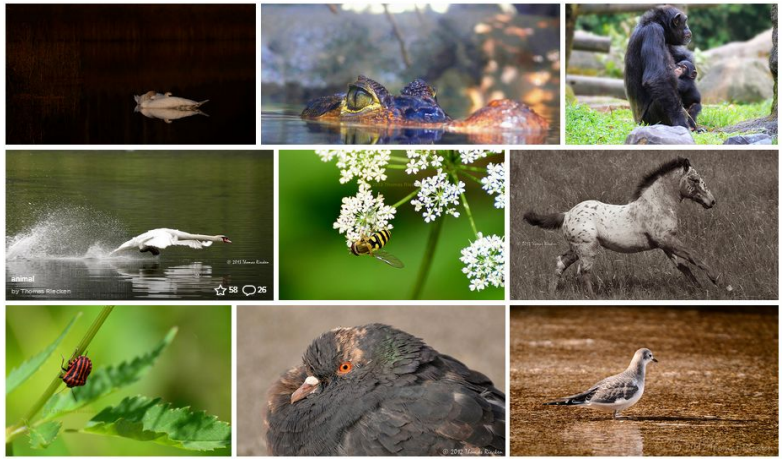


Garbage In - Garbage Out

The **quality** of the **output** of any supervised learning process **is limited by** the **quality** of the supervised information fed in **input**.

$\Phi(i)$ = cat

$\Phi(i)$ = not cat



Garbage In - Garbage Out

The **quality** of the **output** of any supervised learning process **is limited by** the **quality** of the supervised information fed in **input**.

- How to the get input data of good quality?
- How to measure the quality of input?

Corollary: output will be likely worse than input, due to the **inherent difficulty** of the task to be learned with respect to the **representativeness** of the training data.

- How to determine the best quality of output we can expect?
- How to measure the quality of output?

Inter-annotator agreement

Whenever possible, the annotation should be performed by more than one annotator.

- Annotators work **together** on an initial set of documents, to agree/align on how to annotate documents.
- Annotators work **separately** on a **shared** set of documents, to make possible to measure the **inter-annotator agreement.**
- Each annotator works a **distinct** set of documents, to increase the **coverage** of the training set (i.e., a larger number of different documents is annotated)

Inter-annotator agreement

Given a set of documents independently annotated by two or more annotators, it is possible to measure the agreement between annotators.

- Considering in turn the annotations of one annotator as the correct ones
- Then considering those produced by another annotator as predictions and evaluating its accuracy/recall/precision/f1/...

It will be hard for a ML predictor to score a level of accuracy better than the one measured between humans.

Inter-annotator agreement defines a good **upper bound** on the achievable accuracy.

- Yet, super-human performance happen [1] [2] [3] [4]