Usage of modern data exploration tools in e-Lexicography: A practical approach

Alejandro Benito Santos
abenito@usal.es
http://vis.usal.es/

Towards the XXI century dictionary

1. In what ways is it different from an old dictionary?
   a. Physical support is paper?
   b. Is it a list of alphabetically sorted words + definitions?
2. How does it expose the information?
   a. The represented information is static or dynamic?
   b. Does it employ appropriate technologies?
3. How does it interact with the end user?
   a. Promotes user engagement, adapts to user's level of expertise?
   b. Allows users to build their own concept dictionary?
   c. Connects people to concepts that are meaningful to them?

If you want to build a modern dictionary, don’t use old-fashioned methods.

How?

1. Evaluate current processes
    a. Identify highly time-consuming tasks
2. Analyze your artifacts
    a. Excel tables
    b. Lists of words
    c. Manual annotations
    d. Anything that is produced in the process of compiling a dictionary
3. Keep in mind all these sub-processes are susceptible of being automated

Algorithms, NLP, Graph Theory, SNA, Data Mining, Semantic Web, GIS, DataVis, Machine Learning...

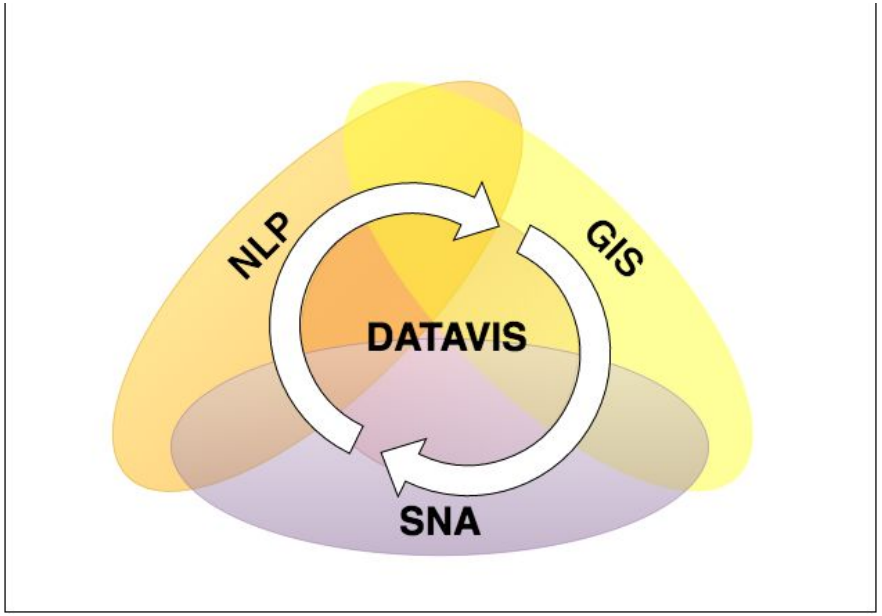
Dialectology, Cultural Studies, Lexicography, Ethnology, Etymology, History, Disambiguation...

I **HATE** this guy

This is a **WASTE** of time

3 (+1) computational pillars of DH

- NLP
- SNA
- GIS
- DataVis



Introducing ElasticSearch

- Has its origins in the information retrieval computational discipline: Apache Lucene library.
- Open source and Free (Apache License)
- Compliant with internet standards.
- Firstly employed to analyze real time machine-generated, human-readable network traffic.
    - If we look at the format these are similar to the formats typically employed to hold dictionary data and other corpora (XML: TEI, TUSTEP)

Are they really so different?

```
<entry xml:id="l564_qdb-d1e949" xml:lang="bar">
   <form type="hauptlemma">
      <orth>li-li</orth>
   </form>
   <gramGrp>
      <pos>Interj</pos>
   </gramGrp>
   <form type="lautung" n="1">
      <pron notation="tustep">li li li li</pron>
      <pron notation="ipa" resp="#JB" change="01">li
   </form>
   <ref type="archiv">HK 564, l5640509.lei^#42</ref>
   <ref type="quelle">Stannern Polack</ref>
   <ref type="quelleBearbeitet">{wb0:CZ-Ig} sIgl.:nba
   <ref type="bibl">
      <bibl>Polack [FB]</bibl>
   </ref>
   <ref type="fragebogenNummer">EFb.9/68: Lockruf f.
</entry>
<entry xml:id="l564_qdb-d1e970" xml:lang="bar">
   <form type="hauptlemma">
      <orth>li-li</orth>
   </form>
   <gramGrp>
      <pos>Interj</pos>
   </gramGrp>
   <form type="lautung" n="1">
      <pron notation="tustep">lililili</pron>
      <pron notation="ipa" resp="#JB" change="01">lil
   </form>
   <ref type="archiv">HK 564, l5640509.lei^#43</ref>
   <ref type="quelle">Ilz Hofer</ref>
   <ref type="quelleBearbeitet">{3.5g11} uFeistritzt.
```

```
/11/11 00:51:23 | --> Monitor Started as Daemon
/11/11 00:51:23 | Launching a Service...
/11/11 00:51:24 |
/11/11 00:52:09 | Nov 11, 2012 12:52:09 AM org.apache.catalina.startup.E
/11/11 00:52:09 | INFO: Starting tomcat server
/11/11 00:52:09 | Nov 11, 2012 12:52:09 AM org.apache.catalina.core.Stan
/11/11 00:52:09 | INFO: Starting Servlet Engine: Apache Tomcat/6.0.35
/11/11 00:52:09 | Nov 11, 2012 12:52:09 AM org.apache.catalina.startup.D
/11/11 00:52:09 | WARNING: Could not get url for /javax/servlet/jsp/reso

/11/11 00:52:10 | Nov 11, 2012 12:52:10 AM org.apache.catalina.startup.D
/11/11 00:52:10 | WARNING: Could not get url for /javax/servlet/jsp/reso

/11/11 00:52:12 | Nov 11, 2012 12:52:12 AM org.apache.catalina.startup.C

/11/11 00:52:12 | INFO: No default web.xml
/11/11 00:52:13 | Nov 11, 2012 12:52:13 AM org.apache.catalina.core.Appl
/11/11 00:52:13 | INFO: Initializing Spring root WebApplicationContext
/11/11 00:52:14 | Nov 11, 2012 12:52:14 AM org.zkoss.zk.ui.http.WebManag
/11/11 00:52:14 | INFO: Starting ZK 5.0.10 EE (build: 2012010610)
/11/11 00:52:14 | Nov 11, 2012 12:52:14 AM org.zkoss.zk.ui.sys.ConfigPar
/11/11 00:52:14 | INFO: Loading system default
/11/11 00:52:15 | Nov 11, 2012 12:52:15 AM org.zkoss.zk.ui.sys.ConfigPar
/11/11 00:52:15 | INFO: Parsing jndi:/localhost/WEB-INF/zk.xml
/11/11 00:52:18 | Nov 11, 2012 12:52:18 AM java.util.prefs.FileSystemPre
/11/11 00:52:18 | INFO: Created system preferences directory in java.hom
/11/11 00:52:21 | Nov 11, 2012 12:52:21 AM org.apache.catalina.core.Stan
/11/11 00:52:21 | INFO: Starting Servlet Engine: Apache Tomcat/6.0.35
/11/11 00:52:22 | Nov 11, 2012 12:52:22 AM org.apache.catalina.startup.C

/11/11 00:52:22 | INFO: No default web.xml
/11/11 00:52:22 | Nov 11, 2012 12:52:22 AM org.apache.catalina.core.Appl
/11/11 00:52:22 | INFO: Initializing Spring root WebApplicationContext
/11/11 00:52:32 | Nov 11, 2012 12:52:32 AM org.apache.coyote.http11.Http
/11/11 00:52:32 | INFO: Initializing Coyote HTTP/1.1 on http-80
/11/11 00:52:32 | Nov 11, 2012 12:52:32 AM org.apache.coyote.http11.Http
/11/11 00:52:32 | INFO: Starting Coyote HTTP/1.1 on http-80
/11/11 00:52:32 | Nov 11, 2012 12:52:32 AM org.apache.coyote.http11.Http
/11/11 00:52:32 | INFO: Initializing Coyote HTTP/1.1 on http-443
```

The answer is no

- They both contain textual information structured in domain-specific standards.
- Feature extraction process.
  - Very similar techniques are applied in both cases
- Size of data is also similar (Text-only documents).
- Data is related to a certain time and space:
  - Geolocation of IP addresses / Localization of texts
  - Network time analysis / Source datation
    - Scales are different (ms vs years)

Introducing ElasticSearch (I)

- Google-like search engine on top of our corpus.
- Incorporates many useful NLP features:
  - Stemmers
  - Language Analyzers
  - N-Gram generation
  - Removal of stop words
  - Misspelling detection
- Performs in real time
- Allows for statistical analysis of textual and numerical features.

Introducing ElasticSearch (II)

- Works great when dealing with space & time analysis and the exploration of massive data sets (>1M)
- Full-text & faceted search
- But…
  - This powers come at a price.
  - It has a steep learning curve.
  - Requires expert-level computer science skills.
  - Under heavy development. Difficult to maintain.

Introducing Kibana

- Kibana is an open source data visualization plugin for ElasticSearch.
- Easy to use: Only requires general digital literacy.
- Entry-point to big data visualizations.
- No programming experience required.
- Despite employing simple visualizations it is good enough for novel users to learn the standard visual language.
- Offer ready-to-use web interface

2. Data import stage

- XML is the most common data format standard employed in linguistics.
  - Other data formats can (and should be!) supported, specially if we want to connect the data with other sources.
- Data & citizen scientists and computational linguists should get involved at this stage.
- **Data model** holds expert knowledge on the topic and it is key to achieve the goals of the research.
- Feature extraction: (i.e: time & space)

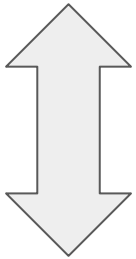
2. Data import stage: Enriching and connecting (II)

- Cross data with other sources:
- RDF and Open-linked data:
  - Europeana
  - Geonames -> Historical disambiguation
  - WordNet
  - Services from other institutes? -> Multilinguality
- Citizen science approaches:
  - social networks
- Related corpora
  - Other historical dictionaries/sources, books, etc.

dbo@ema dataset
Historical geocoding data

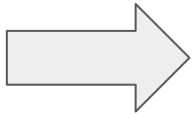
XML

```
SELECT * FROM gemeinde WHERE nameKurz LIKE ? OR
nameKurz LIKE ? OR originaldaten LIKE ? OR originaldaten
LIKE ?
```



```
/(1\d{3})(-\d{2})*|(1\d{1}.x):(\d{2})*-(\d{2})*/g
ort = ort.replace(/\/[A-za-z]+/i, "");
```

- Time
- Space
- Other features




```
"wordType": "adjective",
"mainLemma": "ôrecht",
"leftLemma": "flosch",
"ortName": "Weibern OÖ",
"startYear": "1913",
"endYear": "1967",
▼ "gisOrt": {
    "lat": 48.185371221064,
    "lon": 13.703274111539
},
▶ "gisGemeinde": { "type": "polygon","
▼ "tustep": {
    "A": "HK 634, o6340727.kro^#6",
    "NL": "(flolsch)ôrecht:2",
    "QU": "Weibern OÖ Roi.",
    "QDB": "{5.2b46} mHausrv.:Hausr
    "LT1": "fl-oš,e,orAd",
    "BD/LT1": "mit abstehenden Ohren'
    "BILDDAT": "x",
    "recordNumber": "3234",
    "fileName": "/Users/alexb/Documer
    "orig": " *A* HK 634, o6340727.kro
    Slg./FbB.ROITINGER· (19xx) [Slg."
    *BILDDAT* x "
}
```


HEURISTIC RULES
+
SUPERVISED PROCESS

2. Data analysis stage (I)

1. Identify hidden relationships in the data.
2. Run queries against computed fields.
3. Count occurrences, run statistical analyses, study distribution of query results.
4. Project data in one or more dimensions.
5. Aggregate / cluster data according to your research needs.
6. Repeat until done.

2. Data analysis stage (II): Aggregations

1. Visualizations are built on top of them.
2. An aggregation is a slice of data based on a particular setting of one of its dimensions for a certain query.
3. Very flexible: They can be nested so we have close to infinite possible combinations to build a visualization that serves our purpose.

Here comes a small demo.

That’s it!

Thank you for listening.