Visualizing a large collection of Open datasets: an experiment with proximity graphs

Tianyang Liu¹, Durdana Bangash Ahmed¹, Fatma Bouali¹,², Gilles Venturini¹

¹University François-Rabelais Tours, Computer Science Lab., France
²University of Lille2, France

{tianyang.liu,durdana.bangash}@etu.univ-tours.fr,
Fatma.bouali@univ-lille2.fr, venturini@univ-tours.fr
Talk outline

• Introduction (motivations/objectives):
  - User access to Open data

• Method:
  - Feature extraction with text mining techniques
  - Proximity graph building with KNN graph
  - Graph interactive visualization

• Results on French Open datasets

Overview of the collection
Details on a cluster of datasets
Local links based on similarity
Introduction/motivations

- **Open datasets** = large amount of information
  - [www.data.gouv.fr](http://www.data.gouv.fr): over 353,000 datasets
- How can users/citizens browse such a collection?
- For most Open data web sites =
  - Search engines with keywords and with a basic interface
  - Visual and interactive interfaces are rare (see [www.data.gov](http://www.data.gov), [data.gov.uk](http://data.gov.uk))
- Can we do better than that?

Query = health children (implicit OR)
Results = 499 datasets
3 relevant datasets in the first pages
Followed by a lot of Census datasets about cities …
Helping users to find Open datasets of interest in a large collection

- **Inspiration from Ben Schneiderman** « *Overview first, zoom and filter, then details on demand* »

1. **Provide the user with a visual and interactive overview of the complete collection:**
   - A kind of *map* with navigation, zoom, filtering, ... and opening of the dataset
   - Discovery of *clusters* of datasets (with similar topics), *relations* between clusters, outliers (rare topics), ...

2. **Suggest datasets to explore**
   - User is exploring one dataset, other *similar datasets*?
   - A content-based search guided by links based on local *similarity*

---

1. Datasets **download** and pre-processing
2. Feature **extraction** with text mining techniques
3. Proximity graph **building**
4. Graph **visualization** and **exploration**
Method/Feature extraction

- An Open dataset =
  - Meta-data: title, keywords, description
  - Data file: rows and columns but also several tables, texts, images, ...

- Meta-data = **textual** information, well formatted
- Data file = **unstructured** => not adressed in this paper

- Feature extraction with **text mining techniques**:
  - **Detect** words + years + zip codes, Stop list, Truncation,
  - **Extract** features with 1) **bag of words** or 2) **N-grams**
    - 3 gram: matrix -> mat, atr, tri, rix
  - **Compute** features frequencies in each dataset
  - Zipf law and the **TFIDF** scheme

=> Data matrix: n documents x m features

M. W. Berry and M. Castellanos. Survey of Text Mining II: Clustering, Classification, and Retrieval. 1 edition.
• Proximity graph = given $n$ data + distance, create edges between data

• KNN graph:
  - connects each data to its K nearest neighbors
  - complexity = $O(n^2)$ but possibly $O(n \log n)$ with KD tree optimization,
  - at least K datasets to suggest for each node
  - can create several connected components


Selection of a graph visualization method:

- **Node/link representation** (1 node = 1 dataset, edges from KNNG, length of edges = f(similarity))

- Size of the graph => layout with multi-level approaches like the $FM^3$ method

- **Tulip software**:  
  - various algorithms for graphs  
  - interface for interactive exploration  
  - added plug-in: clicking on a node => downloads + opens the dataset

---


Results/initial experiments

• Downloading of 293,769 datasets from www.data.gouv.fr (in June 2012)

Feature extraction:
• bag of words, 4-grams, 3-grams => too many features => data matrix is too large for building the graph

• 2-grams => m = 650 => building the graph is possible
• Resulting data matrix: \( n \times m = 191.10^6 \) values, 1.456GB

Building the graph:
• \( n = 293,769, K = 4 \) => 881,307 edges, too large for Tulip
• use of sampling, \( n \) reduced to 151,460 datasets
• \( K = 3 \) => 454,280 edges
• \( K = 4 \) => 605,840 edges and 34 connected components
• Overview with annotations

• 1st largest component
• 5 large sub-clusters
  ⇒ Census datasets
  ⇒ predefined categories:
  RPTL: Resident
  RPTEPA: Employ-Population
  RPTCE: Job Characteristics
  RPTCFM: Couples - Families
  RPTESP: Population structure

• Central hub
  ⇒ Miscellaneous, non-Census datasets

• 2nd largest component ⇒ Census datasets about Diplomas and training

• Small disconnected clusters
  ⇒ KNNG
• Other small clusters of interest
• Zoom on cluster 30.6 about Entertainment
Results/Visual map

- Zoom on cluster 30.6 about Entertainment
- Preferences and usage
- Frequency
Consider a user who is exploring a dataset about «health children»

Suggest the immediate neighbors of this node in the graph:

With the search engine, 2 relevant documents in the first 30 pages (i.e. 240 returned documents)
Conclusions

• Operational approach for the visual and interactive exploration of a large collection of Open datasets
• Combination of existing techniques:
  • text mining + proximity graphs + visual and interactive graph layout
• First experiment with positive results:
  • visualization of half of the collection, clusters and links seem to make sense

• Limitations and perspectives
  • What about the other half of the collection?
    ⇒ Gephi?
  • Taking the content into account
  • 2-grams are basic:
    ⇒ Select non census data (between 1% and 2% of the collection)
    ⇒ Test « bag of words » on them
  • improve the suggestions and local exploration of the subgraph:
    ⇒ more neighbors but connected short edges

• User evaluation: comparison between our tool and the search engine (web site)
• Proximity graph = given \( n \) data + distance, create edges between data:

- **KNN graph:**
  - complexity = \( O(n^2) \) but possibly \( O(n \log n) \) with KD tree optimization,
  - at least \( K \) datasets to suggest for each node
  - KNN graph can have several connected components

---


