Content visualization of scientific corpora using an extensible relational database implementation

Eleftherios Stamatogiannakis, Ioannis Foufoulas, Theodoros Giannakopoulos, Harry Dimitropoulos, Natalia Manola and Yannis Ioannidis

National and Kapodistrian University of Athens

HELLENIC REPUBLIC
National and Kapodistrian University of Athens

Management of Data, Information, and Knowledge Group

Introduction

- Goal: content-based visualization of scientific documents
- Scientific documents: rich and diverse
- Applied on three datasets
- Application on publications that share a common funding scheme (e.g. EU FP7-ICT): funding mining submodule
- Implemented in madIS: data analysis via extended relational db
- OpenAIRE+ EU project
    - “2nd-Generation Open Access Infrastructure for Research in Europe” - 283595
    - infrastructure of publication - data repositories
    - implements EC’s open access policies
    - connects publications to research data and funding
    - automatic content clustering and classification

Content-based classification (1)

- Document d representation:
  1. tokenization
  2. stop word removal
  3. stemming
  4. term frequency $df_d(t)$ calculation
- repeat for each class c
- estimate $P(t)$ and $P(t|c)$
- build for each class c:
  - a dictionary D_c
  - an array of respective weights W_c

Content-based classification (2)

- add term t if $\frac{P(t|c)}{P(t)} > T$
- $W_c(t) = \frac{P(t|c)}{P(t)}$
- classification based on sum of logs
- actually equivalent to the Naive Bayes classifier

Visualization of text corpora, Class representation - dimensionality reduction (1)

- Goal: class representation in 2D space
- Classes of similar content are close
- A dimensionality reduction task (classes instead of samples)
- Previous step: class represented by a set of terms and weights
- Compute similarity matrix:

$$S(c_1, c_2) = \frac{\sum_{i=1}^{N_1} W_{c_2}(k : D_{c_1}(i) = D_{c_2}(k))}{\sum_{i=1}^{N_2} W_{c_2}(i)} + \frac{\sum_{i=1}^{N_2} W_{c_1}(k : D_{c_2}(i) = D_{c_1}(k))}{\sum_{i=1}^{N_1} W_{c_1}(i)} \quad (1)$$

classes c_1 and c_2, dictionaries D_{c_1} and D_{c_2}, weights W_{c_1} and W_{c_2}, number of terms per dictionary N_1 and N_2

Visualization of text corpora, Class representation - dimensionality reduction (2)

- S: class distributions in the $\mathbb{R}^M$ space (M: #classes)
- Reduce dimension to 2 using discretized class representations
- Step 1: reduce the feature space via clustering (k clusters):
  - use rows of S as samples
  - compute distance between rows i of S and clusters j: $d(i, j)$
  - each class now represented by its distances from clusters
  - k feature space
- Step 2: use SOM to map k-dimensional space to 2D
- avoid numerical - computational issues in SOM training

Visualization of text corpora, Content visualization of unlabelled corpora

- "Batch" classification + visualization for a corpus
- For every document $i, i = 1, \ldots, N_d$:
  - apply the classifier
  - soft-outputs: $P_i(c)$ for each class $c = 1, \ldots, M$
- Collection of documents is represented for each class c, using:
  - the 2-D class estimated coordinates
  - the accumulated estimated content class probability

$$[X_c, Y_c, \frac{\sum_{i=1}^{N_d} P_i(c)}{N_d}]$$

  where, X_c, and Y_c are the estimated 2-D class coordinates
- Adopt a balloon representation

Fund Mining Module

- Goal: detect particular funding schemes
- Started with EU FP7-funded
- Recently extended to Wellcome Trust projects
- Currently: handle arbitrary number of funding
- Funding information important for: funding statistics - visual analytics.
- Here funding information is used to specify the types of documents being visualized
- Module either used on individual docs or in batch mode
- Preprocessing (stopwords removal, tokenization, etc)
- Find matches against known lists of project grant agreement numbers - acronyms
- Use contextual information to filter out false matches

Datasets

- arXiv (arxiv.org)
  - coverings 7 categories
  - 2 level hierarchy (2nd: 130 classes 2nd level - 7 general categories)
  - only use 2nd level labels
  - arXiv.org API used to retrieve the docs (450K abstracts used)
- BASE (www.base-search.net).
  - open access archive of scientific docs
  - operated by Bielefeld University Library
  - DDC categorization (Dewey Decimal Classification)
  - 35K annotated documents in the English language
  - 2 DDC levels (i.e. 100 classes)
- WoS (Web of Science)
  - http://thomsonreuters.com/web-of-science
  - 180 class labels (non-hierarchical)
  - 18K labelled abstracts

Implementation Issues (1)

- Training and testing (both classification and visualization modules):
  - implemented in Python
  - external libraries, e.g. NumPy, NLTK,...
- *But:* need a release version for the testing case:
  - implementation in the context of a data processing workflow,
  - easily transferred to a distributed environment
- Achievable? Yes: adopted scheme only involves text segmentation, dictionary terms retrieval and a simple weights computation

Implementation Issues (2)

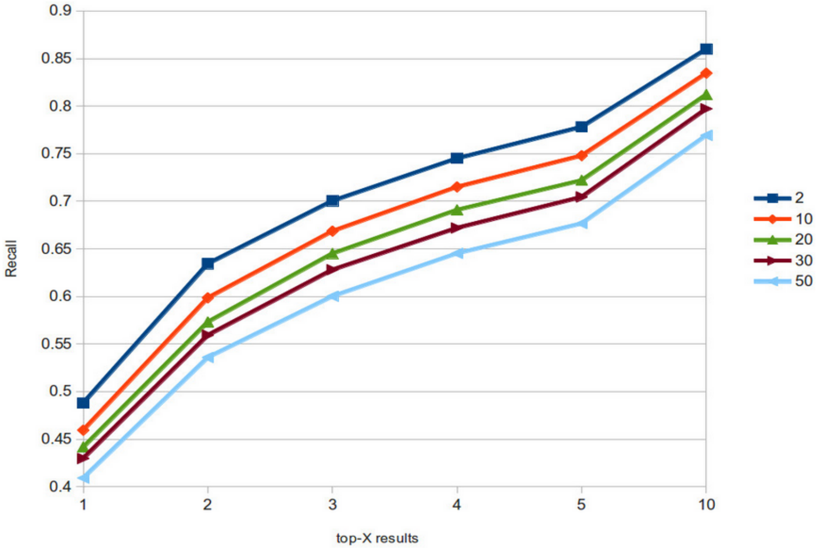
- Implemented in madIS (https://code.google.com/p/madis/)
- Data analysis via an extended relational database
- Built on top of the SQLite database with Python extensions
- Feels like Hadoop SQL without the overhead but also without the distributed processing capabilities
- madSQL, an SQL-based: extended with UDFs (User Defined Functions)
- Eliminates the effort using UDFs (UDFs are first class citizens in the query language itself)
- UDF categories:
  - row: analogous to the Map operator of Map/Reduce systems
  - aggregate: capture arbitrary aggregation functionality beyond the one predefined in SQL (SUM(), AVG(), ...).Analogous to the Reduce operator
  - virtual table: (table functions in Postgresql and Oracle) used to create virtual tables

Implementation Issues (3)

- UDF functionality + traditional relational DB facilities (UDFs closely tied to the relational DB engine): eliminates the communication cost between the two execution layers (functional/relational)
- Naive bayes classification example:
  - use a UDF to split documents into words
  - use the relational facilities to calculate word frequencies
  - use aggregate UDFs to compute sum of logs
  - ALL done in one madSQL query, completely within madIS
- every process (classification, visualization) is implemented in terms of an (extended) SQLite query:

  create temp table if not exists resultsTable as select ontop(5, p, title,class,matches,p) from (select title,class,jgroup(term,p) as matches,sum(p) as p from (select * from (select title,textwindow((summary),0,0,2) from abstractTable),arxiv where middle = term or regexpr('(\S+)(\s)(\S+)',middle,'1') = term) group by title,class) group by title ;

Results - Classification Evaluation on the arXiv dataset



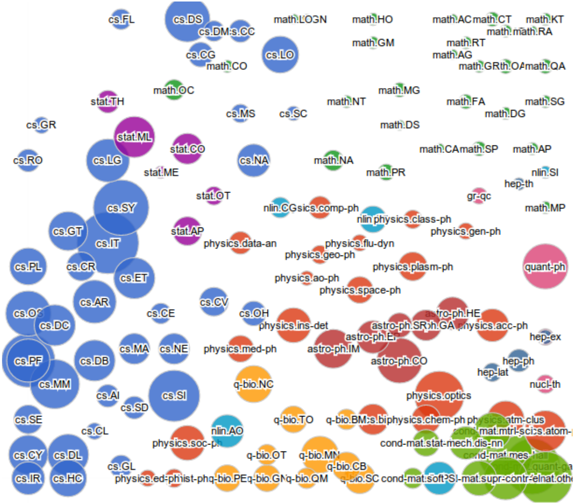
(different probability ratio thresholds)

Results - Classification Module Execution Times

Table: Average execution times per abstract. Higher dictionary thresholds (less dictionary terms) obviously lead to faster classifications. Times are in msecs

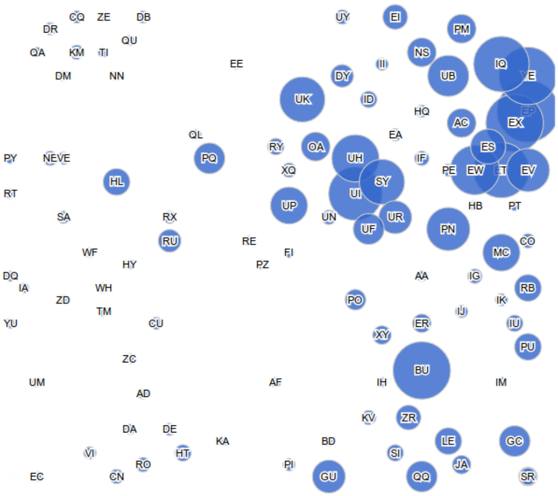
# abstracts	T = 2	T = 10	T = 20	T = 30	T = 50
10	63	27	21	20	17
15000	57	23	15	13	10

Results - Visualization Example: FP7 - ICT calls - ARXIV



arXiv	
astro-ph.GA	Astroph. - Galaxy Astrophysics
astro-ph.SR	Astroph. - Solar and Stellar
cs.DB	CS - Databases
cs.DL	CS - Digital Libraries
cs.IT	CS - Information Theory
cs.LG	CS - Machine Learning
cs.PF	CS - Performance
cond-mat.other	Condensed Matter - Other
hep-lat	High Energy Physics - Lattice
hep-th	High Energy Physics - Theory
physics.geo-ph	Physics - Geophysics
physics.optics	Physics - Optics
physics.space-ph	Physics - Space Physics
quant-ph	Quantum Physics
q-bio.CB	Quantitative Biology - Cell Behavior
stat.ML	Statistics - Machine Learning

Results - Visualization Example: FP7 - ICT calls - WOS



WOS	
BU	Astronomy - Astrophysics
GC	Geochemistry & Geophysics
GU	Ecology
IQ	Engineering, Electrical & Electronic
EX	Computer Science, Theory & Methods
HL	Health Care Sciences
LE	Geosciences, Multidisc.
PU	Mechanics
RU	Neurosciences
UB	Physics, Applied
UK	Physics, Condensed Matter
UI	Physics, Multidisc.
UP	Physics, Particles & Fields
YE	Tellecom.

OpenAIRE mining service BETA

- http://hatter.madgik.di.uoa.gr:8080/openaireplus/classifier
- returns a jason-like result (for several taxonomies, not only arXiv)

Ongoing work

- Detailed *Evaluation* (visualization functionality)
- Visualization enhancement (e.g, a tag cloud for each estimated class probability)
- Semi-supervised techniques (e.g., probabilistic topic modeling).