Cross-lingual and temporal Wikipedia analysis

Göbölös-Szabó Julianna

MTA SZTAKI Data Mining and Search Group

June 14, 2013

Supported by the EC FET Open project "New tools and algorithms for directed network analysis" (NADINE No 288956)

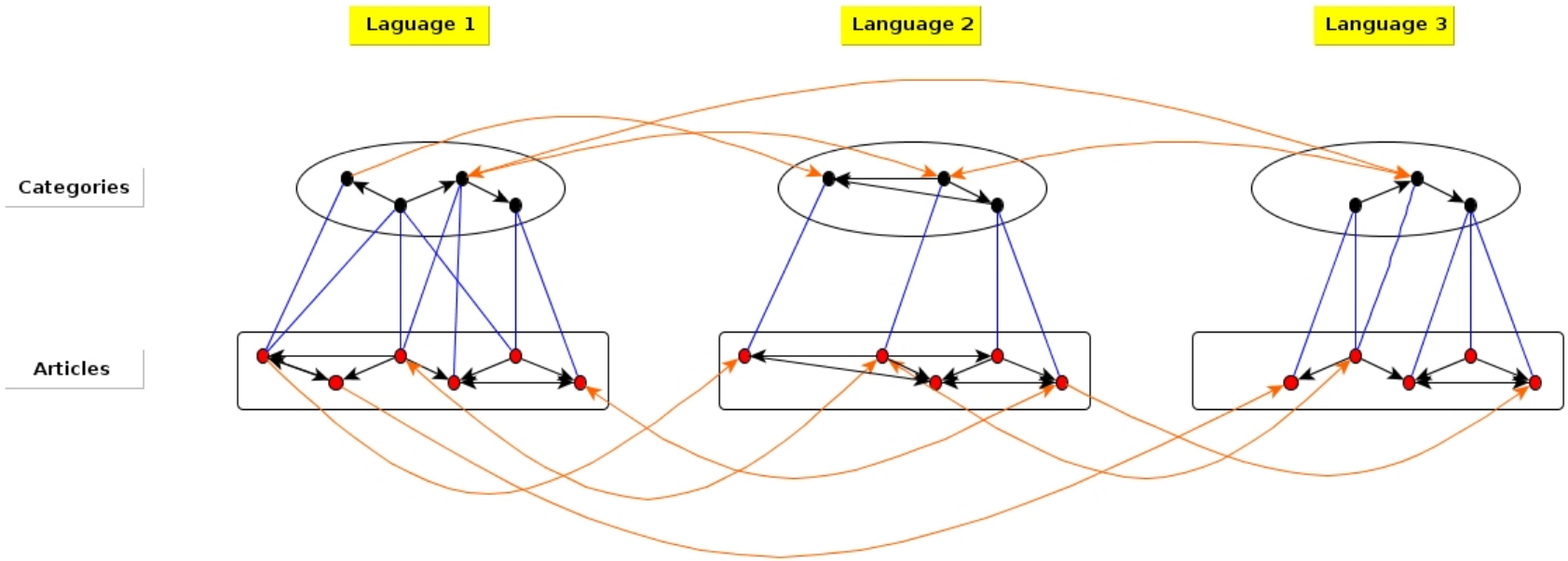
Table of Contents

Section 1

Link prediction on multilingual Wikipedia

Wikipedia articles about Erdős-number in German, French and Hungarian

Multilingual Graph model



Edge types:

- links between articles
- category-contains-article relationship
- category-hierarchy-links
- interwiki links (between languages)

Statistics

- 3 languages: German, French, Hungarian
- snapshot from March 2012

lang.	articles	categories
De	2 338 795	139 844
Fr	2 408 097	199 708
Hu	339 041	34 653

Parallel articles

De-Fr	482 196
De-Hu	108 949
Fr-Hu	119 559

Parallel categories

De-Fr	22 175
De-Hu	4 840
Fr-Hu	5 387

- Only a small fraction of pages has an equivalent version
- Category hierarchies are entirely different

Motivation:

- cleansing, expanding local Wikipedia:
  - new content from a bigger Wikipedia to a smaller
  - more detailed content from a smaller, better specified Wikipedia to the bigger one
- Tag recommendation in similarly structured networks (LibraryThing, Amazon)

Link prediction

We were focusing on:

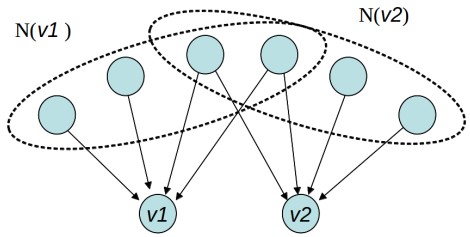
- interwiki link recommendation for categories
- category recommendation for articles
- related entity recommendation for articles

Similar methods are used:

1. Setting candidates
2. Ranking candidates (with Jaccard, SimRank, etc.)

Basic SimRank Equation

- "Two pages are similar if pointed to by similar pages" *(Jeh–Widom KDD 2002)*
- The similarity between objects a and b: $sim(a, b) \in [0, 1]$



$$sim(a,b) = \begin{cases} 1 & \text{if } a = b \\ \frac{C}{|N(a)| \cdot |N(b)|} \sum_{i=1}^{|N(a)|} \sum_{j=1}^{|N(b)|} sim(N_i(a), N_j(b)) & \text{otherwise} \end{cases}$$

- Similarity between a and b is the **average similarity** between in-neighbors of a and in-neighbors of b
- C is called **decay factor**, it is a constant between 0 and 1

Expected meeting distance is the expected time of how soon two random surfers (starting from a and from b) meet at the same node, walking **backwards** on edges.

$$EMD(a,b) = \sum_{v,l} P(\text{after } l \text{ steps } a \text{ and } b \text{ meet at } v) \cdot l$$

Expected f-meeting distance

$$f - EMD(a,b) = \sum_{v,l} P(\text{after } l \text{ steps } a \text{ and } b \text{ meet at } v) \cdot f(l)$$

Usually $f(x) = C^x$ is choosen with $C \in (0,1)$, since it transformes distance to similarity.

Let's define

$$s(a,b) = \sum_{v,l} P(\text{after } l \text{ steps } a \text{ and } b \text{ meet at } v) \cdot C^l$$

- It is easy to show that $sim(a,b)$ is the same as $s(a,b)$

Corollary: SimRank can be approximated with (backwards) random walks.

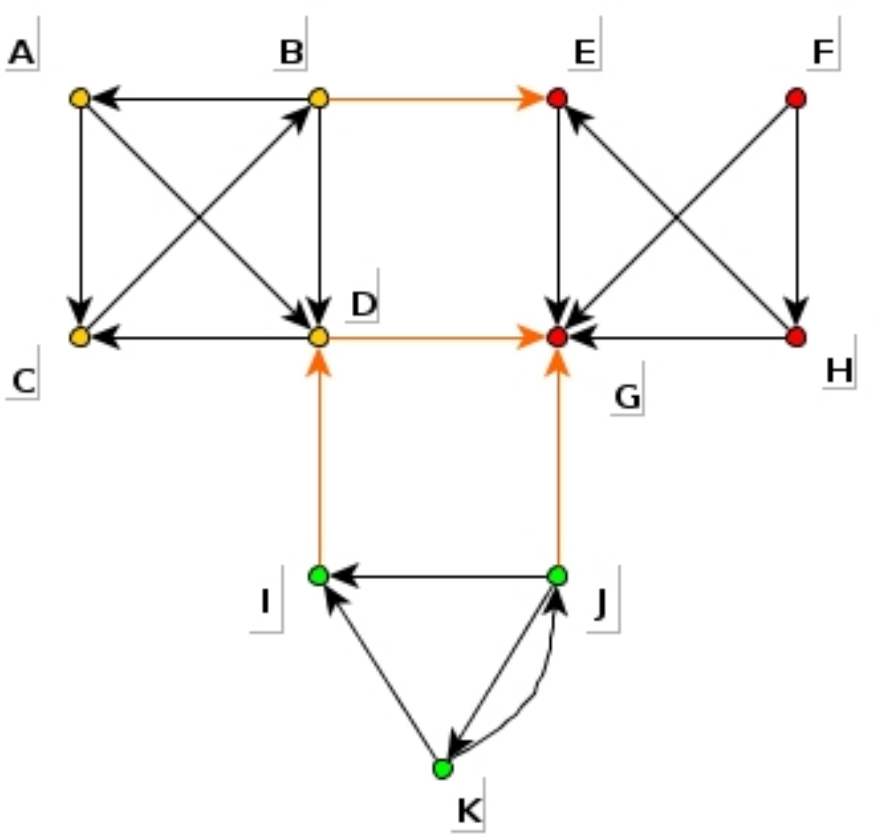
Random walk:

1. decide, whether we continue the walk
   - on a "normal" edge (with α probability)
   - or on an interwiki link (with $1 - \alpha$ probability).
2. select uniformly an edge with the type determined above

Equivalent:

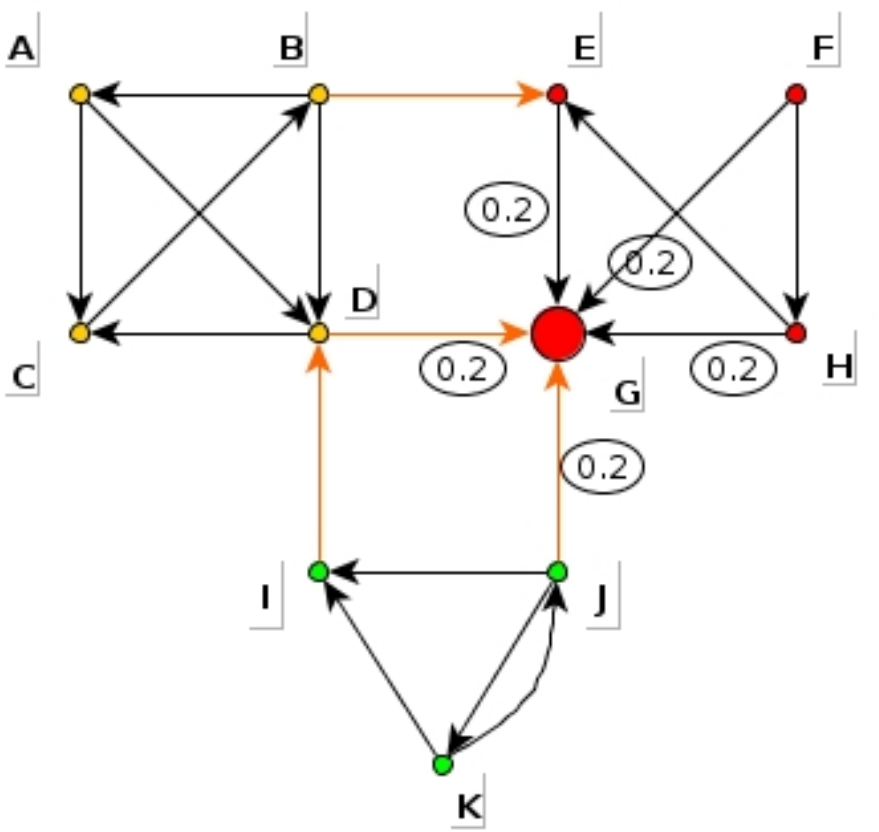
generating random walk on an edge-weighted graph

SimRank for edge-weighted graphs



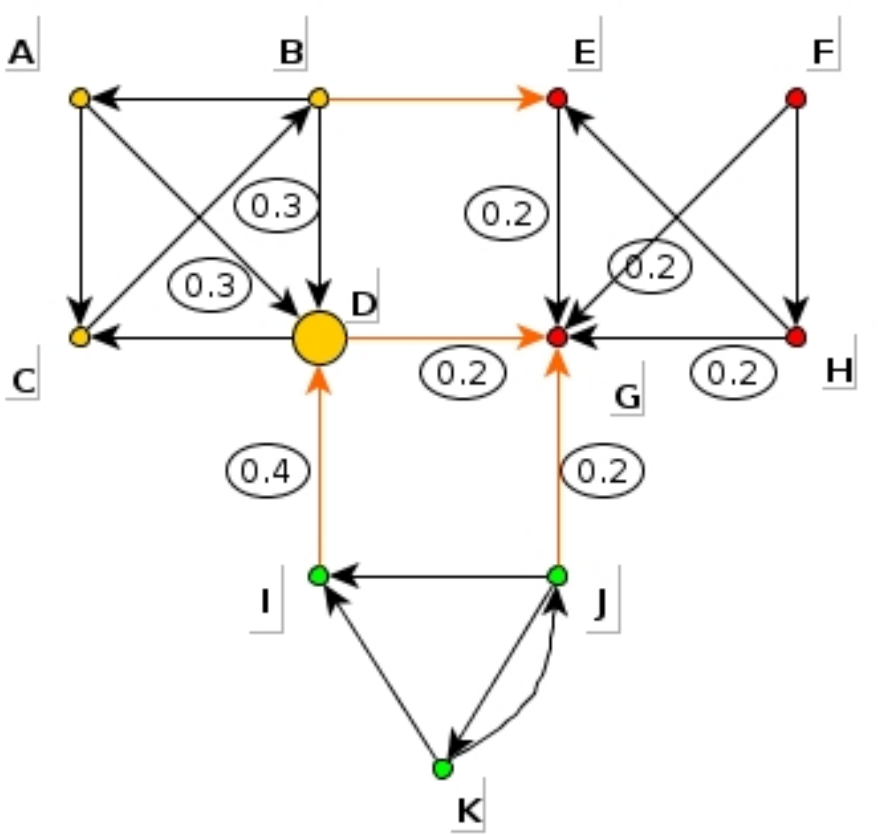
Let's start a walk from G with $\alpha = 0.6$

SimRank for edge-weighted graphs



We choose according to the following probabilities. Let's go to D!

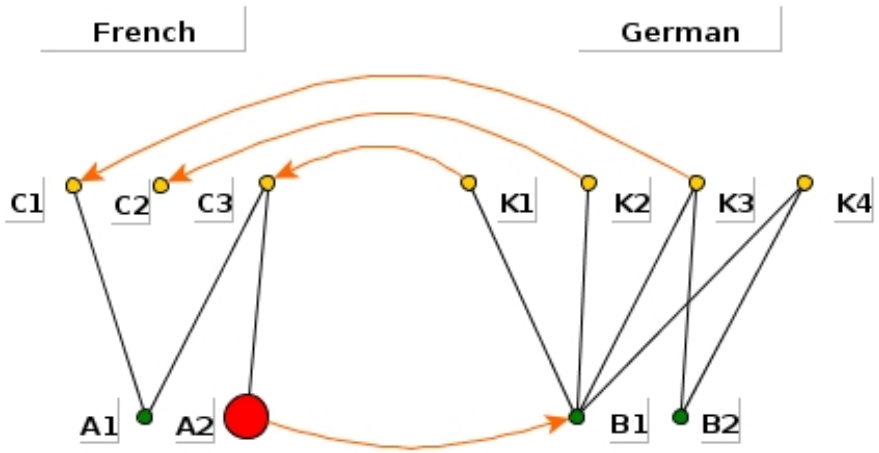
SimRank for edge-weighted graphs



Standing in D we have the following oportunities.

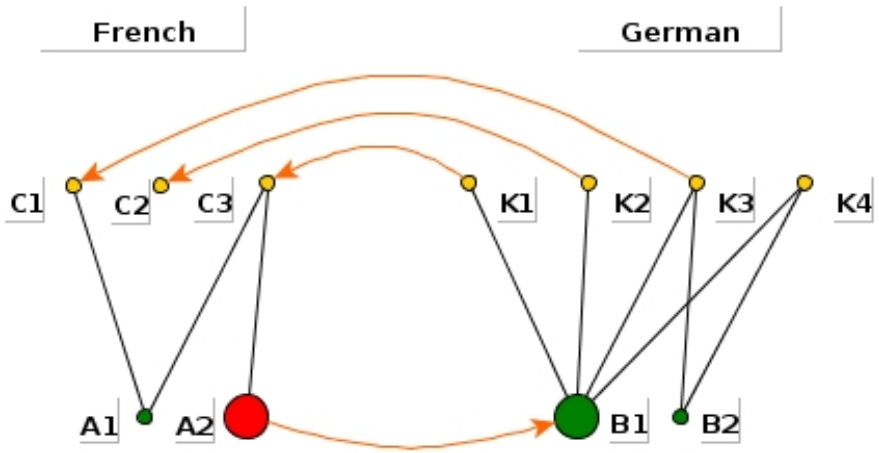
Category recommendation for an article

Given German and French Wikipedia and we want to find a new category for article A_2



Category recommendation for an article

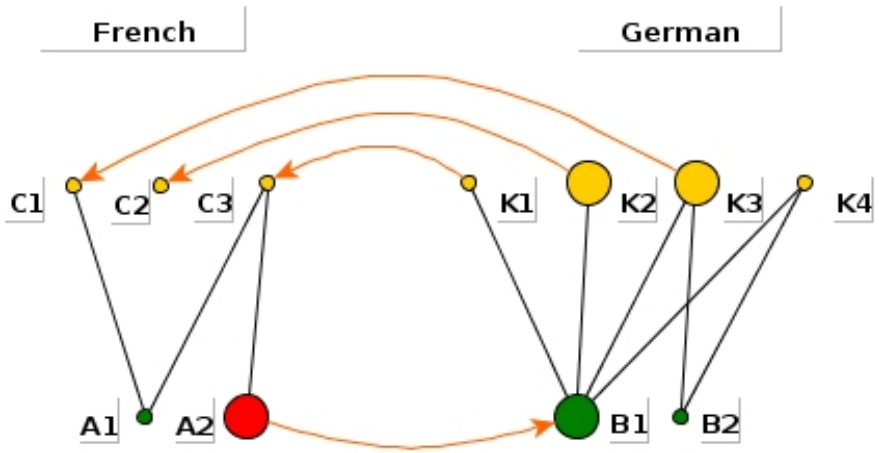
Given German and French Wikipedia and we want to find a new category for article A_2



1. Take B_1, the equivalent article in German

Category recommendation for an article

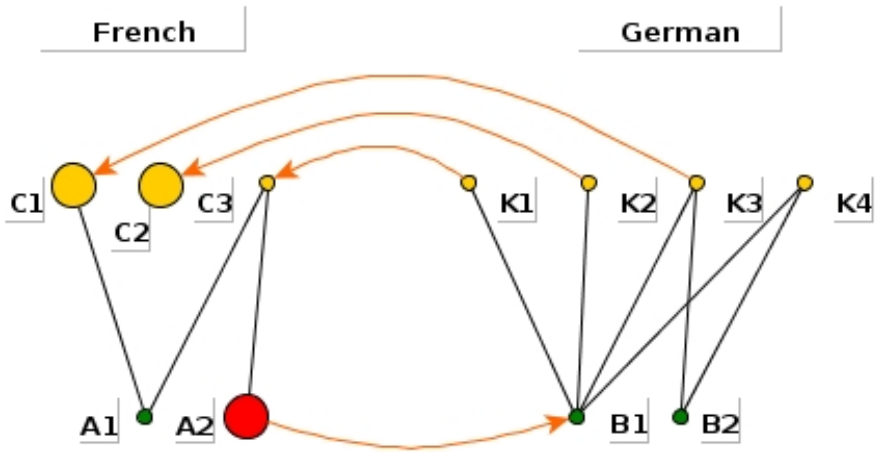
Given German and French Wikipedia and we want to find a new category for article A_2



1. Take B_1, the equivalent article in German
2. Take the categories of B_1 but discard trivial ones (K_1's equivalent is already the category of A_2, K_4 doesn't have a pair in French)

Category recommendation for an article

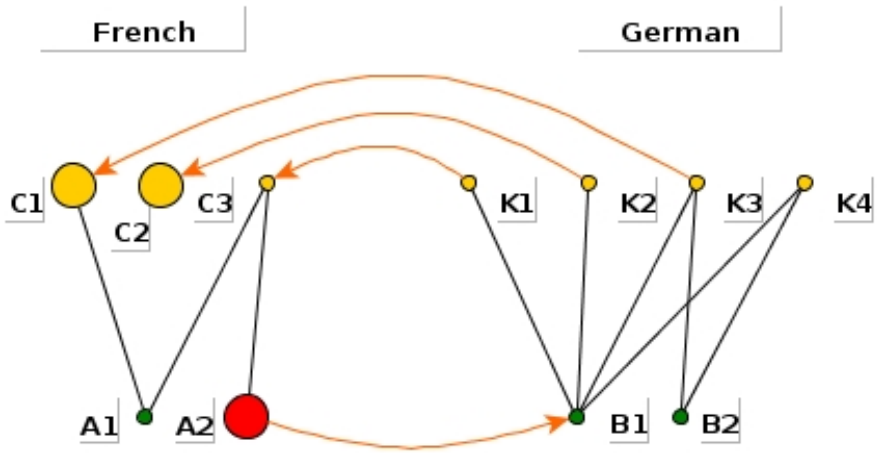
Given German and French Wikipedia and we want to find a new category for article A_2



1. Take B_1, the equivalent article in German
2. Take the categories of B_1 but discard trivial ones (K_1's equivalent is already the category of A_2, K_4 doesn't have a pair in French)
3. The candidates are their French equivalents: C_1, C_3

Category recommendation for an article

Given German and French Wikipedia and we want to find a new category for article A_2



1. Take B_1, the equivalent article in German
2. Take the categories of B_1 but discard trivial ones (K_1's equivalent is already the category of A_2, K_4 doesn't have a pair in French)
3. The candidates are their French equivalents: C_1, C_3
4. Rank candidates

- Weighted Jaccard (details were skipped here)
- SimRank
- Novelty:

$$Nov(x) = 1 - SimRank(c_1, \ldots, c_n, x)$$

where x is a candidate category for article a, and the current categories of a are $c_1, \ldots, c_n$

Similarity of several nodes:

$$s(v_1, \ldots, v_k) = \frac{C}{|I(v_1)| \cdot \ldots \cdot |I(v_k)|} \sum_{u_1 \in I(v_1)} \ldots \sum_{u_k \in I(v_k)} s(u_1, \ldots, u_k)$$

Evaluation

- In each experiment 10 % of the respective edges were deleted (Interwiki links: 13000, Categories: 1 914 000, related articles: 8.5 Mill.)
- For interwiki links: one ground truth for each input
- For categories and related articles: several ground truth instances
- Measures for the output quality:
  - MRR (mean reciprocial rank)
  - nDCG (standard measure for IR - problems)
  - Recall
  - Precision
- Manual assessment for type-2 and type-3

This was a joint work with MPII, Saarbrücken (N. Prytkova, M.Spaniol, G.Weikum)

Section 2

Temporal Wikipedia search by edits and linkage

Motivation

- Wikipedia has the great virtue of being **utterly up-to-date**
- A significant event usually has an immediate trace
- Considering a **chain of events**, we are often interested in the **causes and effects**, naturally represented by citations and links.
- If we want to know **how a story evolved** in time, we also need the information about the **time of appearance of pages and links**

Change measure

		Sep ↓ Oct	Oct ↓ Nov
Muammar Gaddafi	content	0.044	0.18
	inlink	0.55	0.12
	outlink	0.033	0.04
	total	0.63	0.34
Death of Muammar Gaddafi	content	0	7.71
	inlink	0	4.21
	outlink	0	4.64
	total	0	16.6
Battle of Sirte (2011)	content	7.78	0.79
	inlink	4.78	0.21
	outlink	4.9	0.14
	total	17.5	1.1
National Transitional Council	content	0.15	0.08
	inlink	0.91	0.13
	outlink	5.68	0.29
	total	6.7	0.5

We measure change as the sum of

- Difference between the logarithm of the **in-degree** between the two dates;
- Same for **out-degree**;
- Absolute difference between the **number of words** in the article between the two dates.

Change measure

		Sep ↓ Oct	Oct ↓ Nov
Muammar Gaddafi	content	0.044	0.18
	inlink	0.55	0.12
	outlink	0.033	0.04
	total	0.63	0.34
Death of Muammar Gaddafi	content	0	7.71
	inlink	0	4.21
	outlink	0	4.64
	total	0	16.6
Battle of Sirte (2011)	content	7.78	0.79
	inlink	4.78	0.21
	outlink	4.9	0.14
	total	17.5	1.1
National Transitional Council	content	0.15	0.08
	inlink	0.91	0.13
	outlink	5.68	0.29
	total	6.7	0.5

We measure change as the sum of

- Difference between the logarithm of the **in-degree** between the two dates;
- Same for **out-degree**;
- Absolute difference between the **number of words** in the article between the two dates.

- The change of a node is interesting, if the neighborhood of the node has changed as well
- E.g. Learning to rank vs. Occupy movement

Goal:
Given a query Q and we want to find a subgraph that consists of relevant articles respective to the topic, this graph changes with time and this change explains the considered events related to Q.

- 3 snapshots (2011. september, october, november)
- 35 queries with ground truth answers
- Evaluation:
  - recall, NDCG, MRR
  - graph density
- Visualizing the graphs with our in-house built tool *(subjective evaluation)*

Selecting temporal changing subgraph

Main steps of our algorithm:

1. Composing the seed set from the search results
2. Expanding of seed set and consider the induced subgraph
3. Computing personalization vector
4. Assinging scores to the nodes
5. Selecting the top-15 (and their induced subgraph in each snapshot)

- Wikipedia content is indexed by a search engine
- top results usually don't form a connected graph
- we expand this set of nodes in order **to get a possibly connected component**
- new nodes are expected to be **relevant or recently edited**
- candidates: 1-step neighborhood of the seed set

$$\text{score}(v) = \max_{u \in \text{seed}} \left(IR(u) + \frac{change(u) + change(v)}{2} \right)$$

- expand with the nodes with highest rank

- both IR score and change are relevant:
- before combination both value need to be scaled
- α: trade-off between change and relevance
- T: depends on the distribution of change values ($T = 10$ worked fine)

$$p(u) = \alpha \cdot \frac{\text{IR(u)}}{\text{maxIR}} + (1 - \alpha) \cdot \frac{\text{change}(u)}{(\text{change}(u) + T)}$$

Random surfer model

- Browsing the web, following hyperlinks
- Sometimes she gets bored and teleports
- When teleporting, take distribution d (instead of uniform distribution)

$$PPR_d^{(k+1)T} = PPR_d^{(k)T}((1-\alpha)M + \alpha \cdot D)$$
$$= PPR_d^{(1)T}((1-\alpha)M + \alpha \cdot D)^k$$

where D has all rows equal to the personalization vector $\mathbf{d} = (d_1, \ldots, d_n)$

$$D = \begin{pmatrix} \mathbf{d} \\ \mathbf{d} \\ \ldots \\ \mathbf{d} \end{pmatrix}$$

Idea behind HITS:

- A good hub is a page that pointed to many other pages,
- A good authority is a page that was linked by many different hubs

$$\hat{a}(v) = \sum_{vu \in E} w(vu) \cdot h(u), \quad a = \hat{a}/\|a\|_\infty$$

$$\hat{h}(v) = \sum_{uv \in E} w(uv) \cdot a(u), \quad h = \hat{h}/\|h\|_\infty$$

- **supersource:** a new node of the graph which is connected with each node of the original graph, and the weight of an edge corresponds to the importance of the respective node in the personalization
- **The supersource distributes a fixed amount of score in each iteration**

$$\hat{a}(v) = \sum_{vu \in E} w(vu) \cdot h(u) + c \cdot p(v), \quad a = \hat{a}/\|a\|_\infty$$

$$\hat{h}(v) = \sum_{uv \in E} w(uv) \cdot a(u) + c \cdot p(v), \quad h = \hat{h}/\|h\|_\infty$$

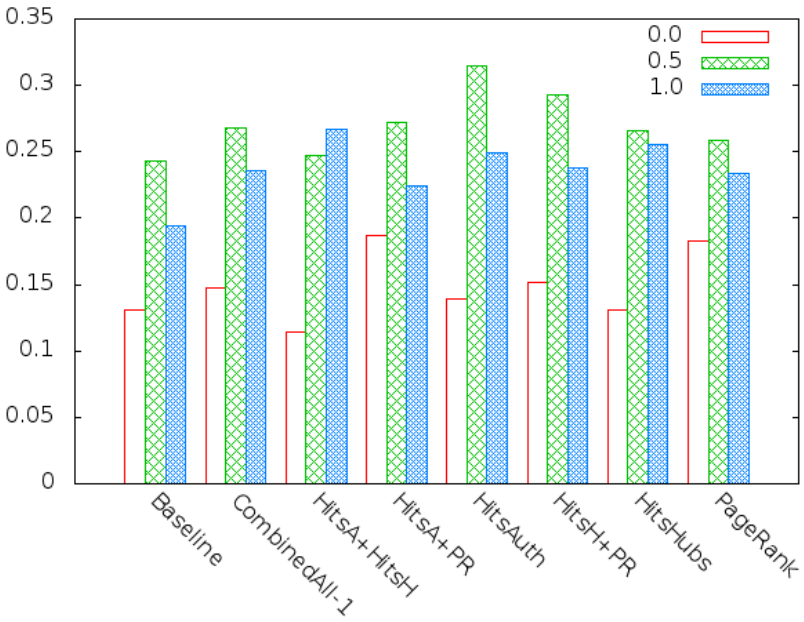
Node scoring

Multiple versions:

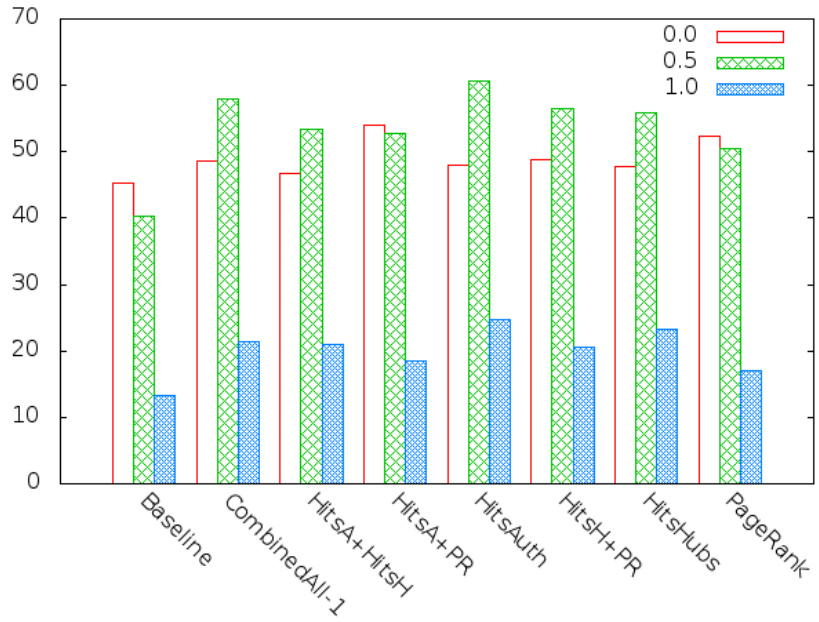
- Scores from Hubs
- Scores from Authorities
- combination of Hub and Authority vector

Other combinations are possible as well:

- Hubs & PageRank
- Authority & Pagerank
- Hubs & Authority & Pagerank

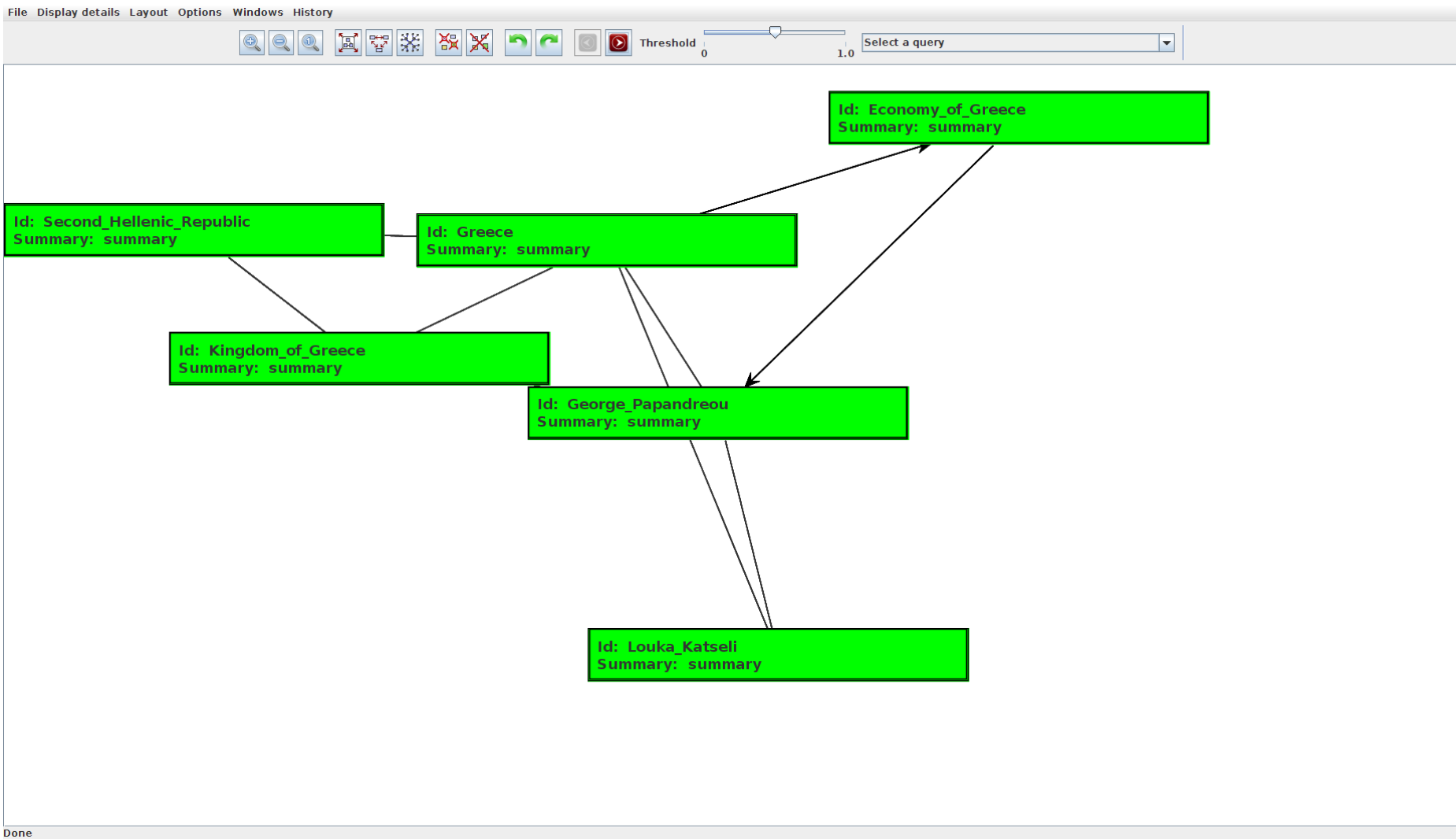


nDCG values



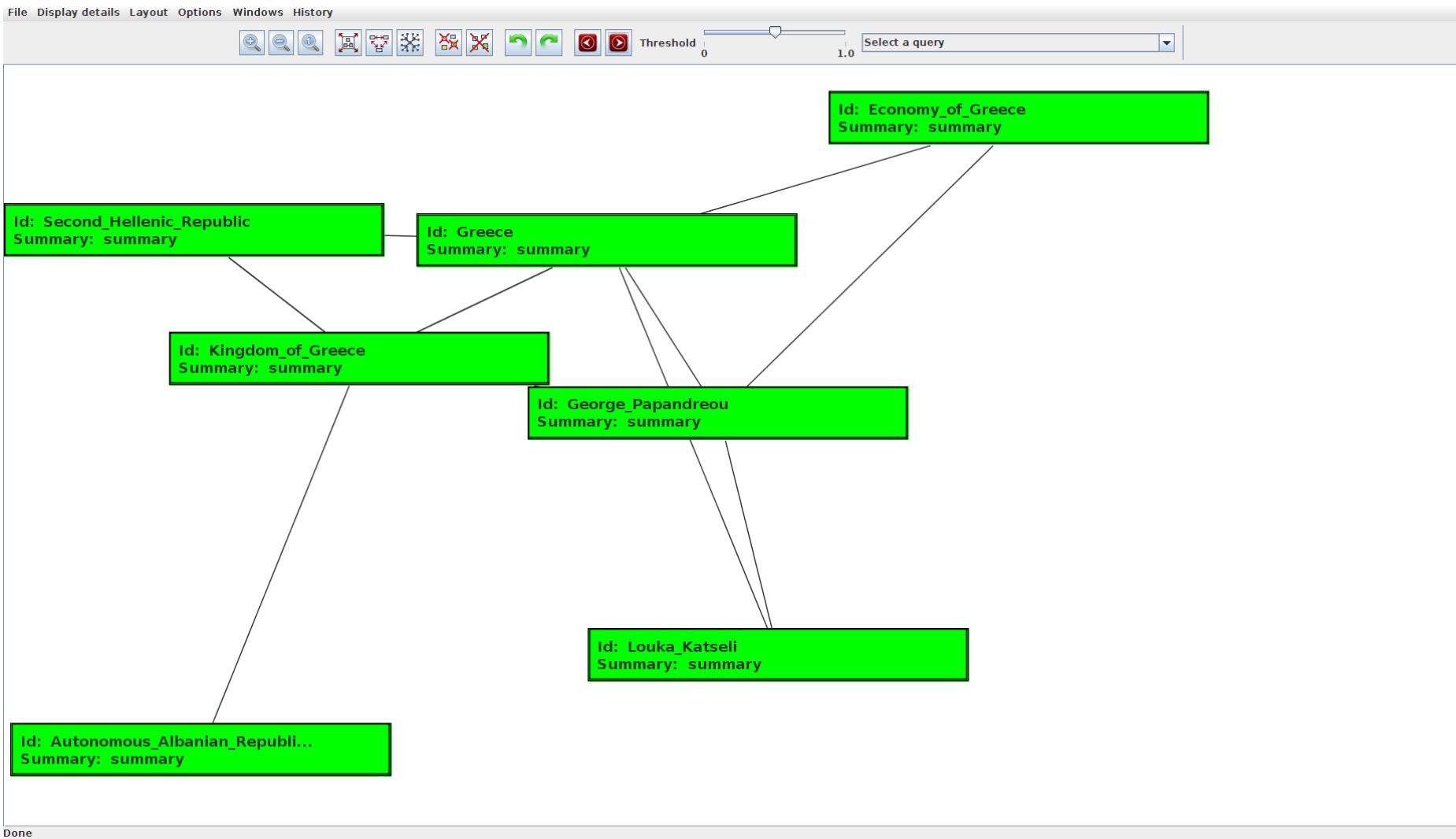
Growth of number of edges

An example for changing subgraph



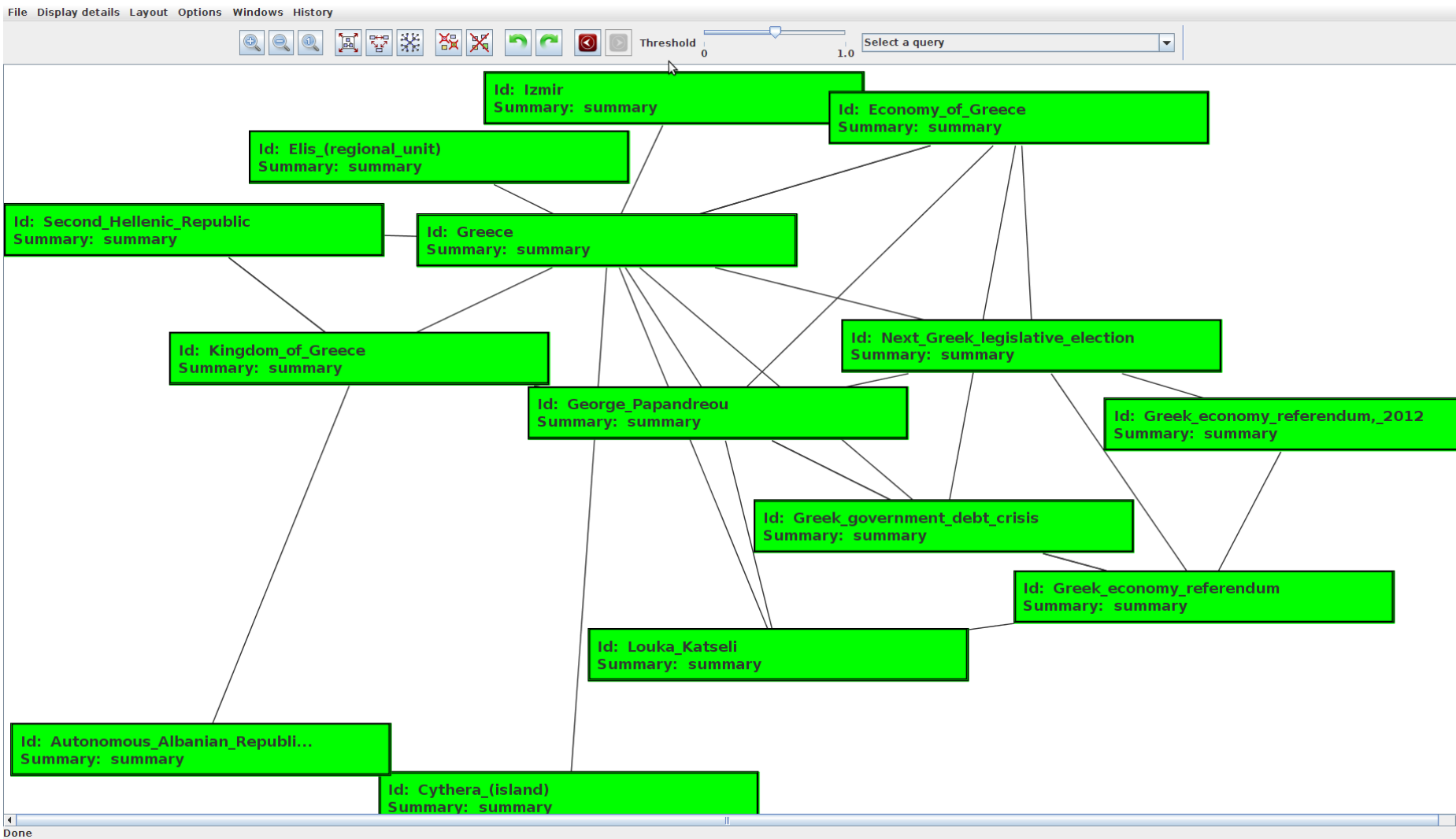
Result for "Greek economy"

An example for changing subgraph



Result for "Greek economy"

An example for changing subgraph



Result for "Greek economy"

Thank you for your attention!