



The Semantic Growbag – Automatically Organizing Topic Facets

Jörg Diederich, Uwe Thaden, Wolf-Tilo Balke,

*Faceted Search Workshop at SIGIR'06
Seattle, Washington, USA*



Outline

- Motivation
- Semantic Growbag
 - Algorithm
- Experimental Setup
- Demonstrator
- Summary and Future Work



Motivation

- Faceted search groups objects (Web pages, documents, etc.) into independent categories for navigational access
 - Users might focus on or remember different aspects when trying to locate information
 - Keywords of categories do not always exist in documents
 - Example: author, publication year, journal,...

- Problem in faceted search:
 - How to organize highly dynamic facets with many instances?
 - Example: “Topic” in a faceted browser for publications

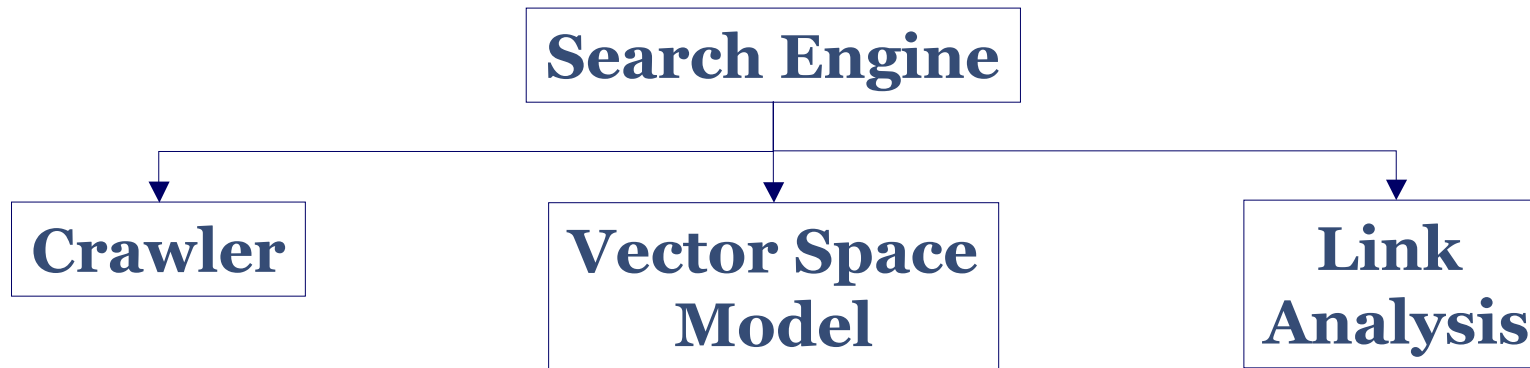


Motivation

- Hierarchies in individual facets
 - Taxonomies
 - But: facet-specific, user-specific, time-variant,...
- Idea: Semantic GrowBag Algorithm
 - Organize a facet comprising the tags from a folksonomy-based corpus of objects
 - Example: Publications annotated with keywords
 - Derive community-driven topic hierarchies automatically
 - Use them to better organize the topic hierarchy
 - Organization dependent on *when* the tagging took place
 - Show development over time (if tagging associated with a date)



Example: Topic Hierarchy



Legacy topic facet:

- Search Engine (10)
- Vector Space Model (3)
- Link Analysis(4)
- Crawler (3)
- [+ 100's more....]

Hierarchically organized topic facet:

- **Search Engine (10)**
 - Vector Space Model (3)
 - Link Analysis (4)
 - Crawler (3)

Problem: How to find such hierarchies automatically?



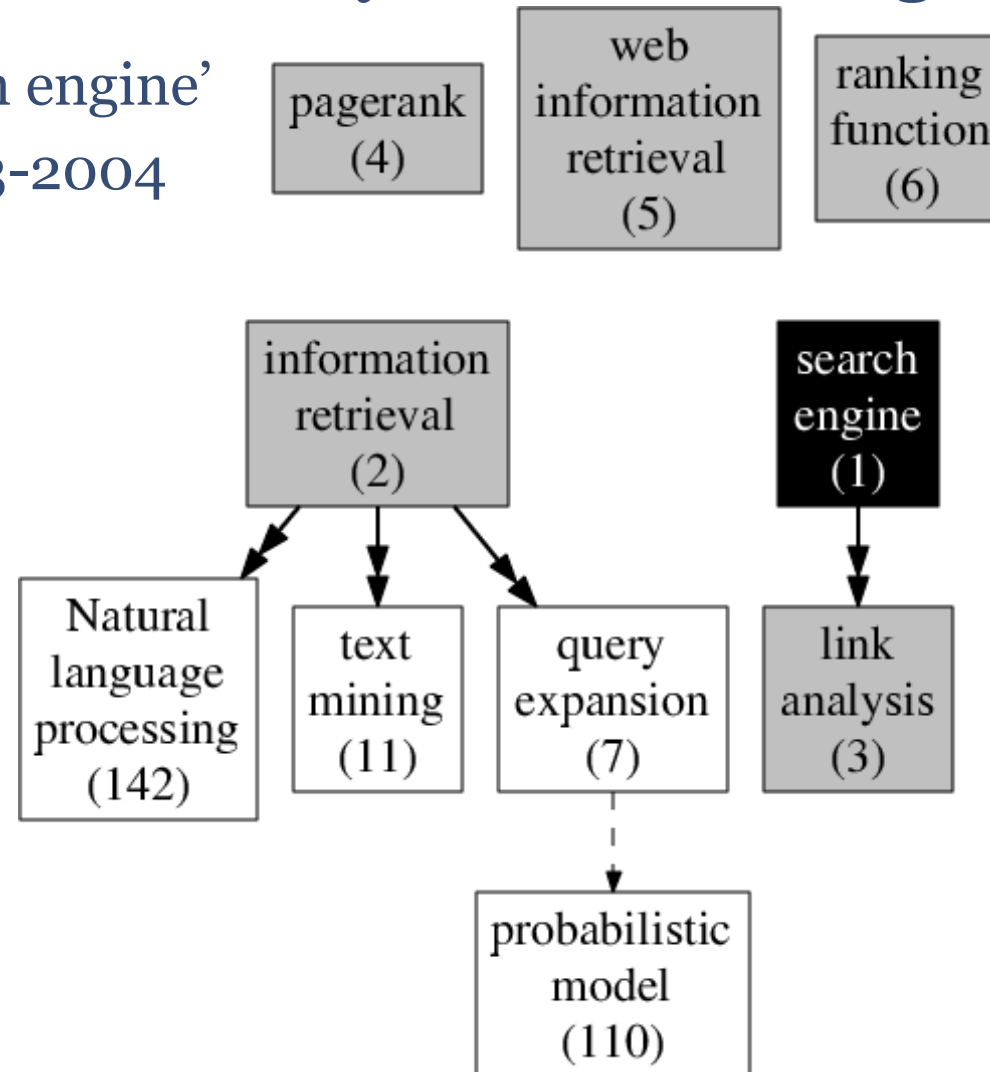
Main Idea

- Example for tagged folksonomy corpus
 - Publications manually tagged with keywords (ACM classification)
 - Social tagging activities (flickr, del.icio.us,...)
- Input data: (<nodeid>, <keyword>) & (<nodeid>, <year>)
- Query:
 - Topic hierarchy of <keyword k> for <startyear> - <endyear>?
- Semantic GrowBag Output: Graph with
 - Keywords as nodes (tagged with their rank)
 - Edges denoting the subsumption hierarchy
 - Different weights for the confidence
 - Graphically displayed in different classes
 - Currently two: 'dashed' vs. 'double-headed'



Example Topic Hierarchy from GrowBag

- Keyword: 'search engine'
- Time span: 2003-2004





The Algorithm: Overview

- 3 basic steps
 1. Determine most closely related keywords for a given keyword k (top- X keywords of k) based on co-occurrences + compute a ranking
 - Using tag-co-occurrence matrix, 'TF-IDF' and Biased PageRank
 2. Find super-topic / sub-topic relations between k and its top- X keywords
 3. Combine the relations and the top- X tags into a single graph, the topic facet of k



Part I: Determine ranking of keywords

- Given query: keyword k , time period p
- 1. Compute a list of top- X related keywords
 - Based on weighted co-occurrences with other keywords during p
 - Take the set of keywords that accounts for 20% of TF-IDF mass
 - Called 'main keywords of k '
- 2. Compute a ranking using Biased PageRank, biasing on these main keywords of k
- 3. Output: Ranked list of keywords related to k (L_k)
- 4. → Do this for all keywords



Example: Step 1 for keyword 'IR'

Rank	Tag	TFxIDF	TF
1	IR	669.0	200
2	Search Engine	67.0	15
3	Language Model	60.2	12
4	WWW	51.1	16
5	Web search	41.4	8
6	Query expansion	39.9	7
7	Text mining	36.2	8
8	Indexing	29.1	7
9	NLP	28.3	6
10	Question Answer	27.0	5
...
22	Inf. Extraction	18.7	4

Co-occurrence analysis

Rank	Tag	Score
1	IR	148.3
2	WWW	103.8
3	Machine Learning	94.3
4	Ontology	91.1
5	Search Engine	90.4
6	Semantic Web	88.9
7	Digital Library	85.6
8	Web search	79.2
9	Knowledge Management	77.3
10	Query expansion	76.2

Biased PageRank: $L_k('IR')$



Step 1 for keyword 'Query expansion'

Rank	Tag	TFxIDF	TF
1	Query expansion	130.0	23
2	IR	23.4	7
3	Probabilistic Model	16.9	3
4	Web search	15.5	3
5	Search Engine	13.4	3
6	Page Segmentation	12.6	2
7	Web information retrieval	12.2	3
8

Co-occurrence analysis

Rank	Tag	Score
1	IR	541.3
2	Query expansion	490.5
3	Probabilistic Models	476.4
4	Search Engines	74.6
5	Web Search	45.3
6	WWW	41.4
7	Data mining	37.5
8

Biased PageRank
(L_k('Query expansion'))



Part II: Find super-topic / sub-topic relations

- Given: keyword k , ranked lists L_k for all keywords k' (for period p)
 1. Extract the top- X keywords from L_k (the ranked_list of k)
 - Top- X the same as in Part I
 2. For all keywords k^* in top- X keywords
 1. Get the scores of k and k^* in L_k and L_{k^*}
 2. If both scores of k are larger than the ones of k^* : k is a super-topic of k^*
 3. Analogously for “smaller than” \rightarrow “sub-topic”
 3. Do it for all keywords k

\rightarrow Output: all super-topic / sub-topic relations

\rightarrow Example: ‘IR’ super-topic of ‘query expansion’, ‘query expansion’ sub-topic of ‘IR’



Part III: Create the topic hierarchy

1. Use the main keywords as found in step 1 as ‘seed’
2. Get all keywords k^* which are sub-topics of the main keywords (or of which the main keywords are super-topics) (output of step 2)
 - Repeat recursively until no additional sub-topics found
 - ‘Growing’ the set of nodes N of the topic hierarchy
3. Add the immediate super-topics of the main keywords
4. Find all super-topic/sub-topic relations where nodes in N are involved in
 - The set of links L in the topic hierarchy
5. Determine the ‘strength’ of a relation k_1 to k_2 :
 - k_1 super-topic of k_2 and k_2 sub-topic of k_1 : high
 - k_2 sub-topic of k_1 : low

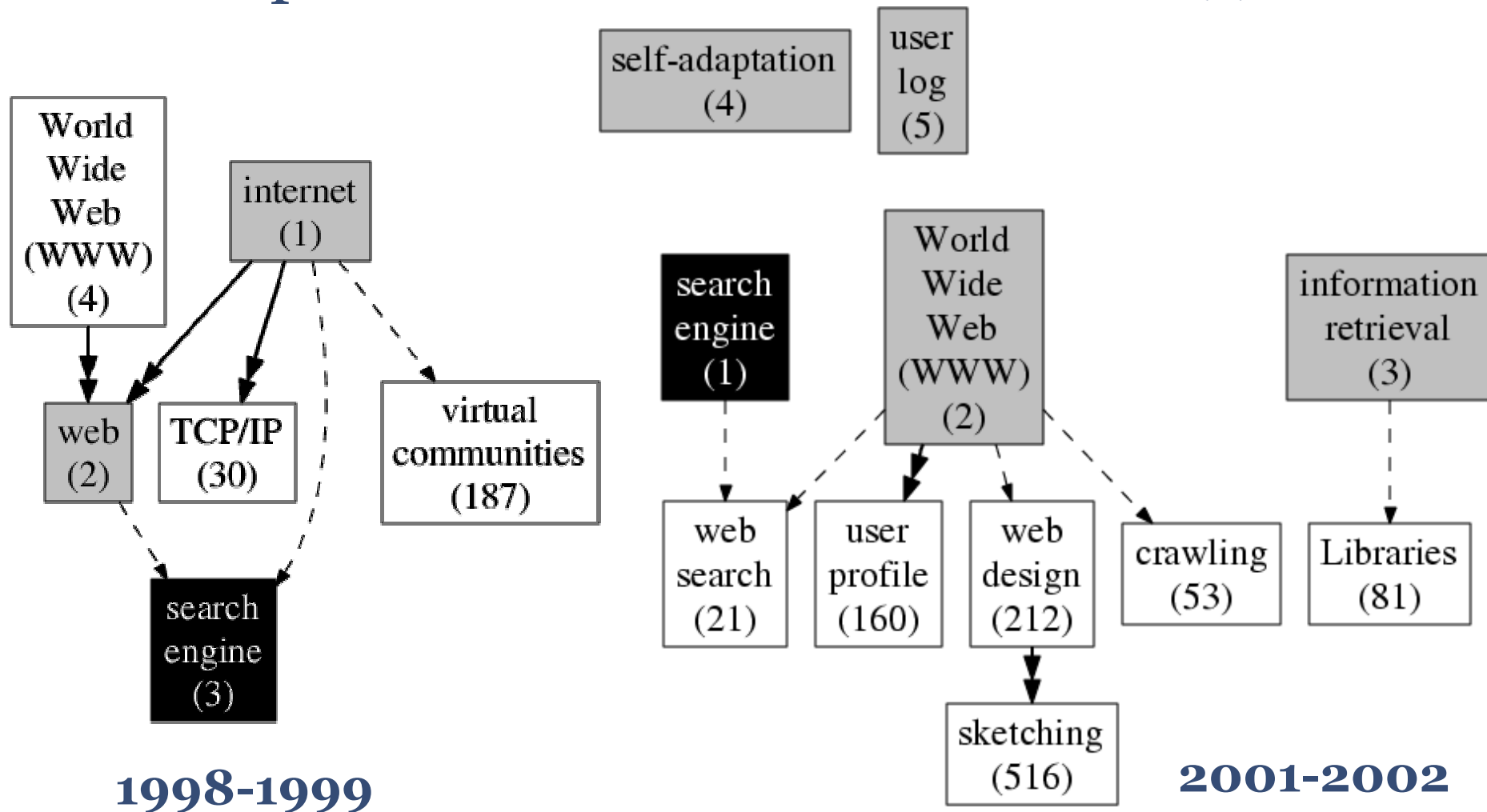


Experimental Setup

- DBLP data (as of July/August 2005)
- Extracted manually added keywords + abstracts using the links provided in DBLP (limited to Springer, ACM DL, IEEE DL)
 - About 53.000 documents stored in mysql database
 - About 180.000 distinct tags
- Postprocessing:
 - Replacing well-known acronyms (e.g. XML,) by the most popular full version in the database
 - Stemming of keywords
- Creating a list of most popular keywords (8622, about 60%)
 - Actually 'key phrases...', not only keywords...

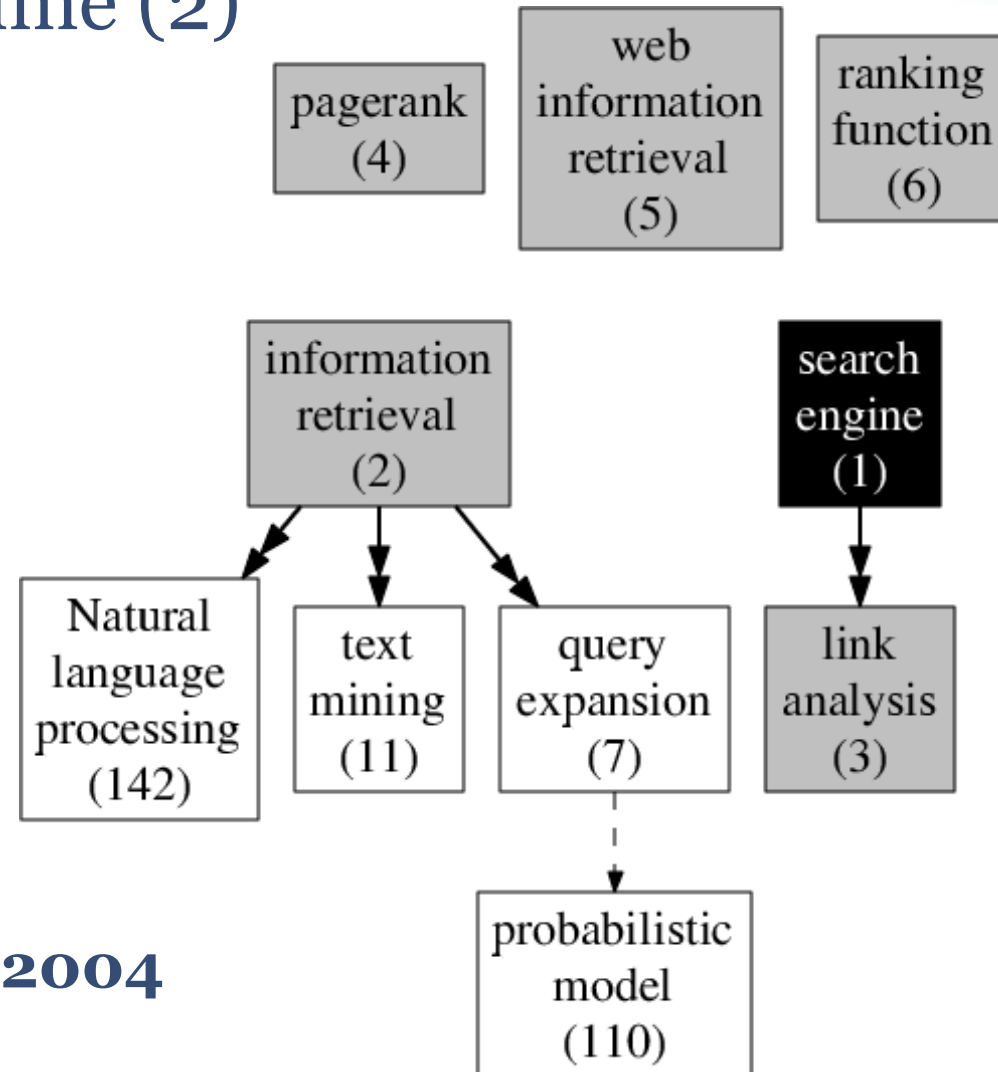


Development of Hierarchies over Time (1)





Development over Time (2)



2003-2004



The Semantic GrowBag Demonstrator for Tagged Computer Science Publications

Available Topic facets for 2003-2004 with at least one strong edge:

Graphs with no strong edge
 Graphs without edges

(In the reduced version those subgraphs, that are not connected to the graph with the start tag, are folded into the participating top-X node. Please note, that quite some graphs do not contain edges because of the power-law nature of the co-occurrence distribution of tags in our collection. The 'top-X' values are an indicator for the size of the 'community' around a tag (limited to 10 for visualization reasons). The 'nodes' and the 'edge' values are shown to have an indication of the size of the graph.

Topic	top-X	Nodes	Edges	Link to Pictures		Development over time	
abstract interpretation	4	10	6	Full version	Reduced version	Full version	Reduced version
abstraction	7	19	14	Full version	Reduced version	Full version	Reduced version
access control	8	19	14	Full version	Reduced version	Full version	Reduced version
active objects	3	3	1	Full version	Reduced version	Full version	Reduced version
adaptation	10	20	10	Full version	Reduced version	Full version	Reduced version
ad hoc	3	6	5	Full version	Reduced version	Full version	Reduced version
ad hoc networks	9	21	20	Full version	Reduced version	Full version	Reduced version



Summary

- Problem:
Sensible hierarchical organization of dynamic facets
- Idea:
Use GrowBag algorithm to automatically create topic hierarchies + organize topic facets accordingly
- Applicable to any tagged set of objects
 - medline, bibsonomy,...



Future Work

- Evaluation of hierarchies difficult...
- Use different input data sets, such as the Medline Database
- Find ‘interesting’ topics (changing over time) automatically (change rate of the top-X over time)
- Detect keywords which are used as synonyms in two non-connected communities
- Allow for more than one start tag



Questions?

balke@l3s.de

