PEST

Term-Propagation over Structured Data using Eigenvector Computation

Fabian Kneißl

Final Diploma Report
Advisors: Klara Weiand, Dr. Tim Furche
Supervisor: Prof. Dr. François Bry

Institute for Informatics
University of Munich

21st October 2010

Outline

Outline

Motivation

Some relevant pages are not included in Top 100 search results

- Search for *Bart* on `simpsons.wikia.com`
  $\Rightarrow$ *Homer* is not in Top 100
- Search on `Google`: the same

Why?

- Only content / page itself matters for determining a match
- PageRank only modifies the ranking
- PageRank is query-independent

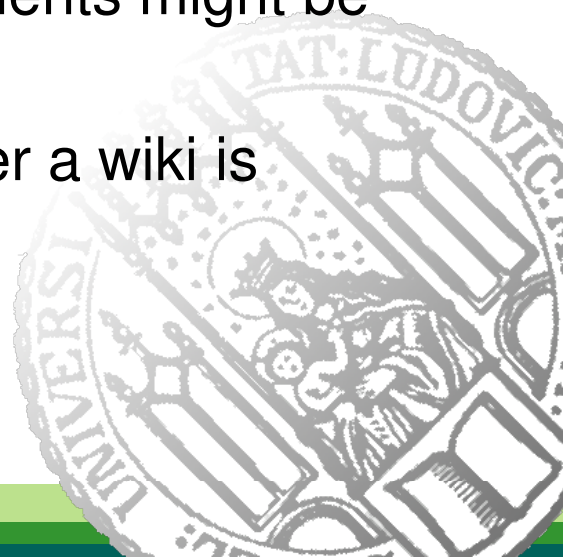
What is the problem?

Area of interest

Search in structured data (wikis, social networks, linked data)

Problems

- Search for a keyword on structured data only yields documents directly containing it
- Documents linked to by several relevant documents might be (more) relevant but are not regarded
- Problems become even more relevant the richer a wiki is structured



What do we want to achieve?

Goal: Fuzzy matching

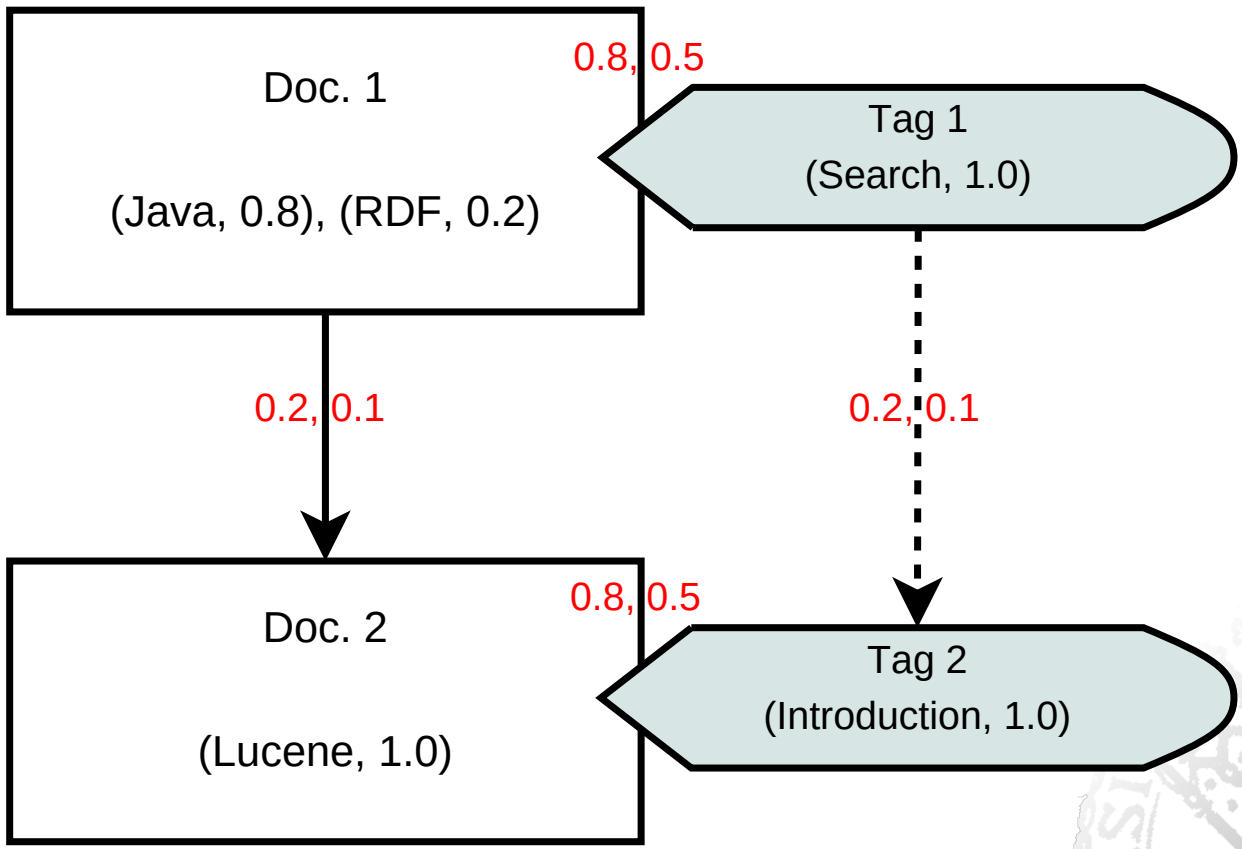
- Include results that are relevant but were not regarded before
- Produce a better search result ranking
- Applicability for all kinds of keyword search problems

Outline

PEST Deciphered

- Term-**P**ropagation
- using **E**igenvector Computation
- over **ST**ructured Data

Simple Example of a Structured Data Graph



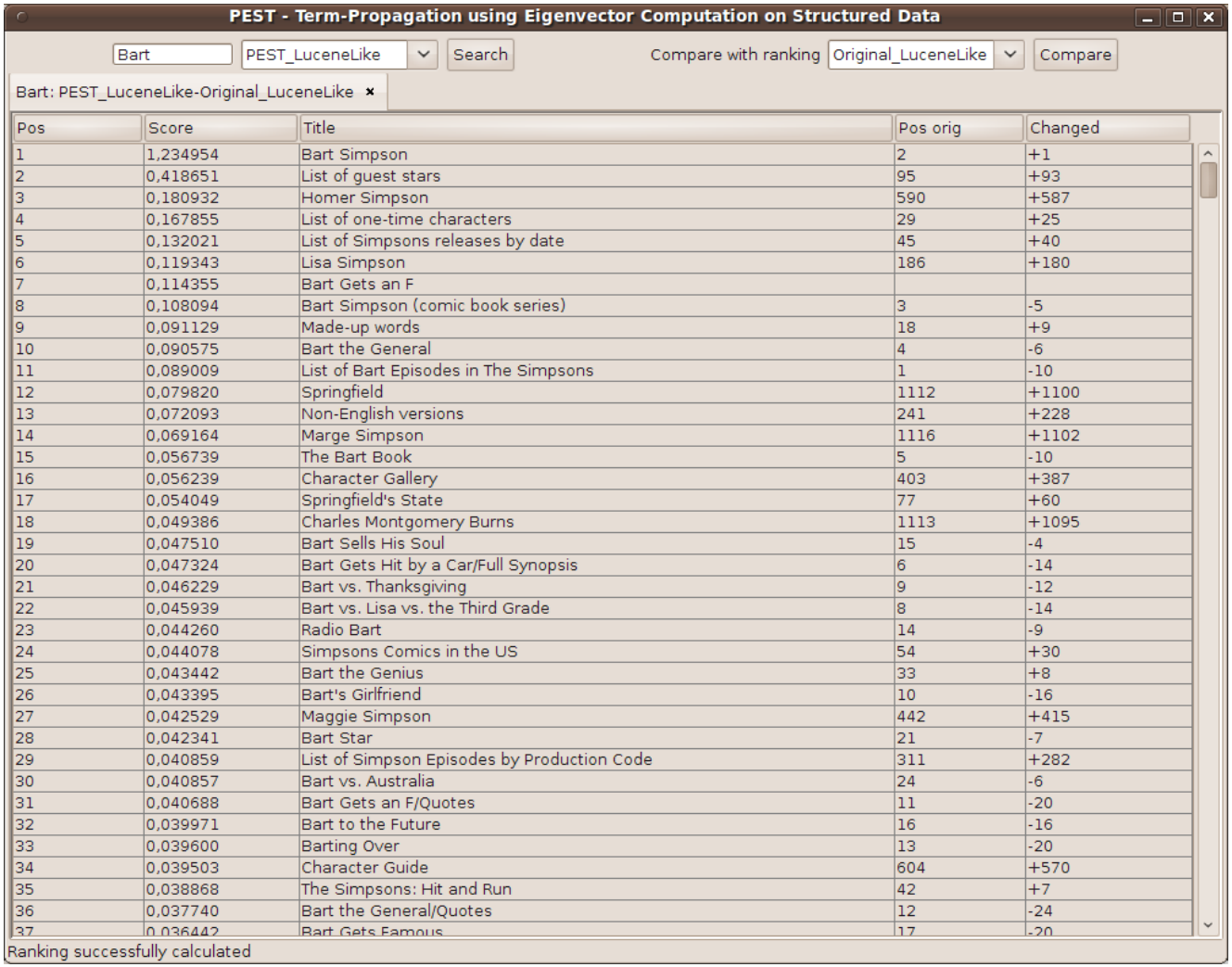
Algorithm - How It Works

1. Weighted propagation graph
   - Convert datasource into graph structure
   - Assign appropriate edge weights
2. Normalization of the adjacency matrix (term-independent)
   - Compute adjacency matrix **H**
   - Normalize each column: divide by # outgoing edges
3. Compute PEST matrix $\mathbf{P}_\tau = (1-\alpha)\mathbf{H} + \mathbf{L}_\tau$
   - Compute leap matrix $\mathbf{L}_\tau$ (term-dependent)
   - $\mathbf{L}_\tau$ consists of **informed leap** and random leap
4. Eigenvector computation
   - Apply power method to $\mathbf{P}_\tau$
   - Resulting eigenvector $\mathbf{p}_\tau$ gives the new term weights for the vertices in the content graph

Implementation

- Java
- Lucene (vector space model search engine)
- Data structures for the dataset:
  - DOT language input format
  - MySQL as backend
- Commandline application + Graphical user interface
- Parameters as commandline arguments and in a custom settings file
- Available at `http://www.pms.ifi.lmu.de/pest`

Screenshot



Pos	Score	Title	Pos orig	Changed
1	1,234954	Bart Simpson	2	+1
2	0,418651	List of guest stars	95	+93
3	0,180932	Homer Simpson	590	+587
4	0,167855	List of one-time characters	29	+25
5	0,132021	List of Simpsons releases by date	45	+40
6	0,119343	Lisa Simpson	186	+180
7	0,114355	Bart Gets an F		
8	0,108094	Bart Simpson (comic book series)	3	-5
9	0,091129	Made-up words	18	+9
10	0,090575	Bart the General	4	-6
11	0,089009	List of Bart Episodes in The Simpsons	1	-10
12	0,079820	Springfield	1112	+1100
13	0,072093	Non-English versions	241	+228
14	0,069164	Marge Simpson	1116	+1102
15	0,056739	The Bart Book	5	-10
16	0,056239	Character Gallery	403	+387
17	0,054049	Springfield's State	77	+60
18	0,049386	Charles Montgomery Burns	1113	+1095
19	0,047510	Bart Sells His Soul	15	-4
20	0,047324	Bart Gets Hit by a Car/Full Synopsis	6	-14
21	0,046229	Bart vs. Thanksgiving	9	-12
22	0,045939	Bart vs. Lisa vs. the Third Grade	8	-14
23	0,044260	Radio Bart	14	-9
24	0,044078	Simpsons Comics in the US	54	+30
25	0,043442	Bart the Genius	33	+8
26	0,043395	Bart's Girlfriend	10	-16
27	0,042529	Maggie Simpson	442	+415
28	0,042341	Bart Star	21	-7
29	0,040859	List of Simpson Episodes by Production Code	311	+282
30	0,040857	Bart vs. Australia	24	-6
31	0,040688	Bart Gets an F/Quotes	11	-20
32	0,039971	Bart to the Future	16	-16
33	0,039600	Barting Over	13	-20
34	0,039503	Character Guide	604	+570
35	0,038868	The Simpsons: Hit and Run	42	+7
36	0,037740	Bart the General/Quotes	12	-24
37	0,036442	Bart Gets Famous	17	-20

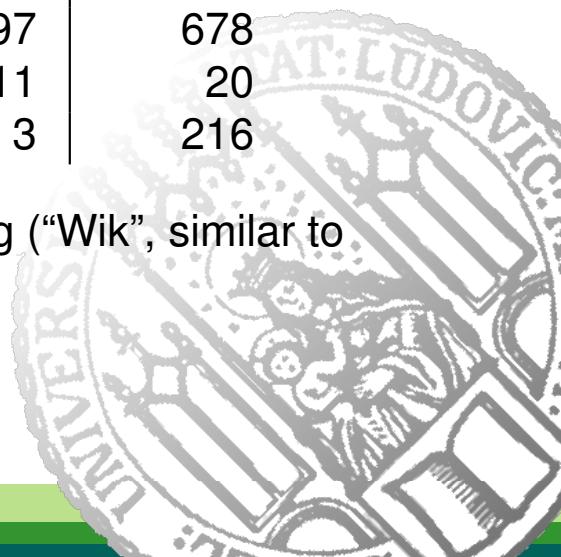
Outline

Ranking the Simpsons

Top 10 ranking for a search for 'Bart'

	PEST score	**Page title**	**Wik**	**Google**
1	0.1348	Bart Simpson	4	1
2	0.0340	Homer Simpson	980	36
3	0.0245	Lisa Simpson	281	181
4	0.0183	Bart the Genius	2	4
5	0.0180	Marge Simpson	1321	548
6	0.0148	Bart Gets an F	19	3
7	0.0115	Bart's Bike	1	-
8	0.0112	Maggie Simpson	497	678
9	0.0105	Bart the General	11	20
10	0.0089	List of Bart Episodes in The Simpsons	3	216

Ranks of the PEST algorithm compared to an enhanced tf-idf ranking ("Wik", similar to Wikipedia's search) and the Google rank.



Simpsons User Study

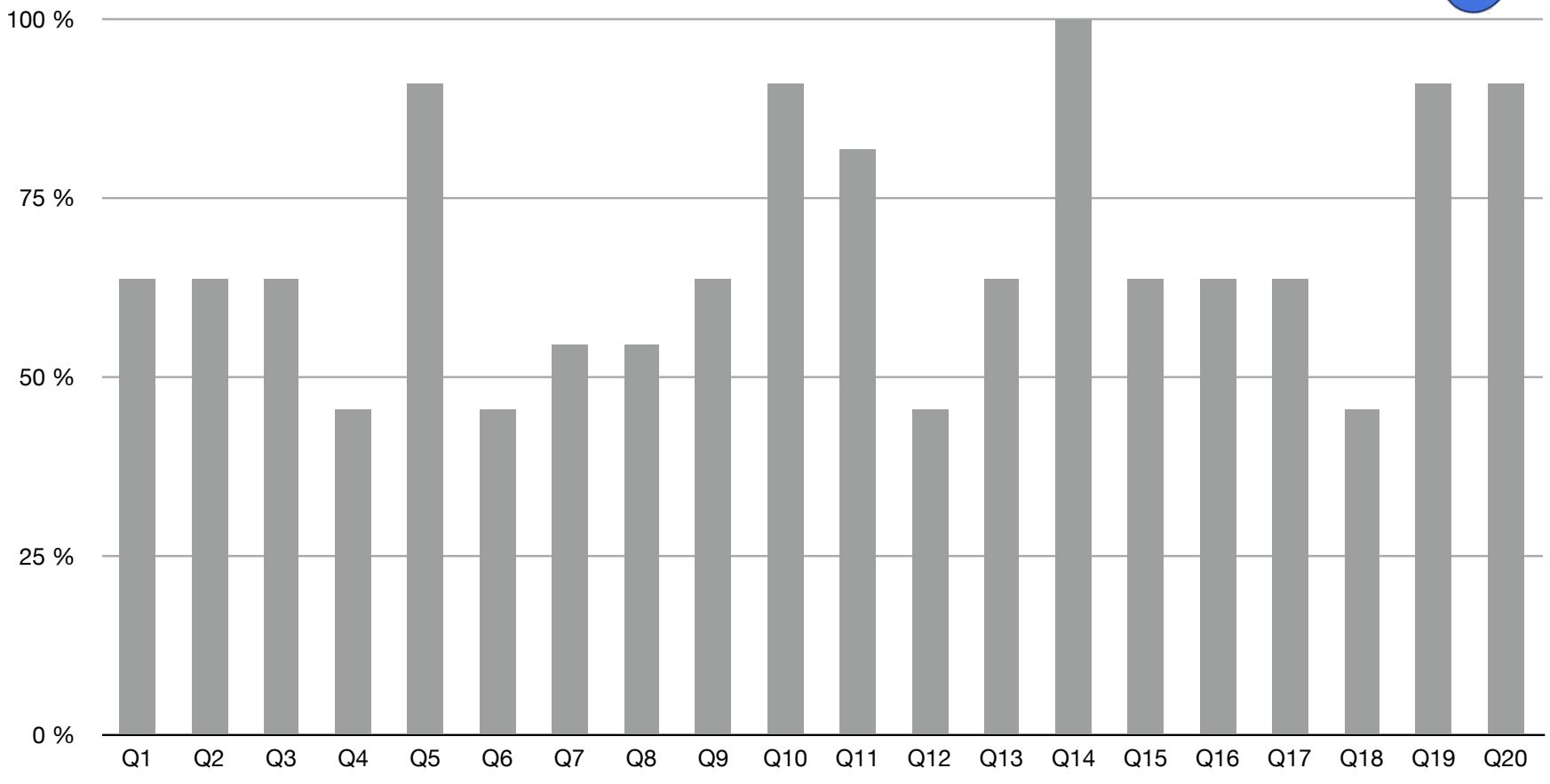
Setup

- 20 keyword-queries
- Comparison of PEST with enhanced tf-idf
- Users decide which ranking they like better
- Small-size user study (11 participants)

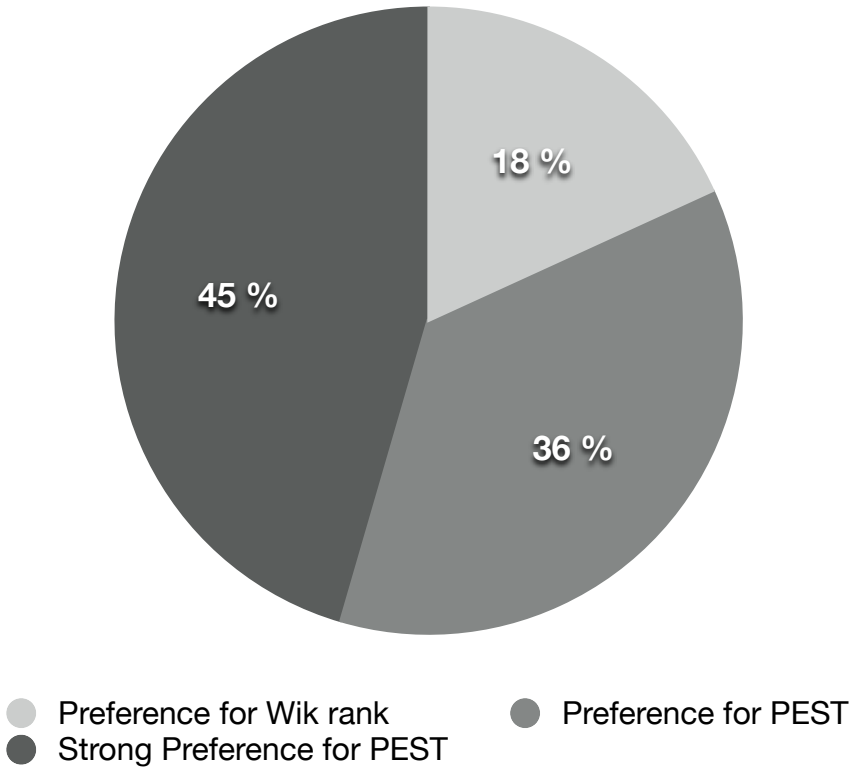
Results

- Across all queries and users: PEST 67% of the time preferred
- 14 queries: PEST better with acceptance of 63 – 100% of all users
- 6 queries: weak preference for or against PEST

Simpsons User Study II - Per Query



Simpsons User Study III - Per User



Percentage of users who preferred the PEST-enhanced ranking for

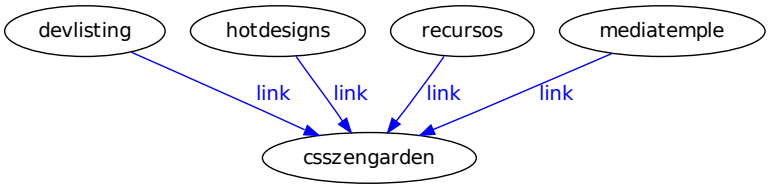
- less than half of the queries ("Preference for Wik rank")
- 50-70% of the queries ("Preference for PEST")
- and 75% and more of the queries ("Strong Preference for PEST")

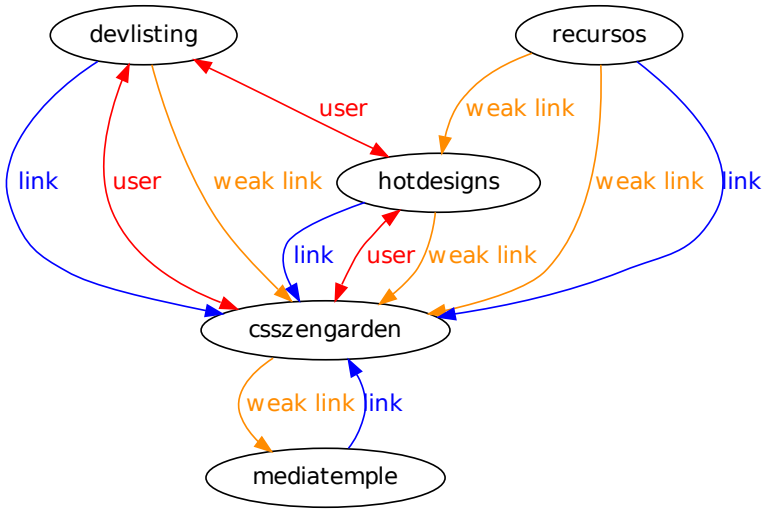
⇒ Paper submitted to Information Systems Journal

Delicious as Social Network

delicious
social bookmarking

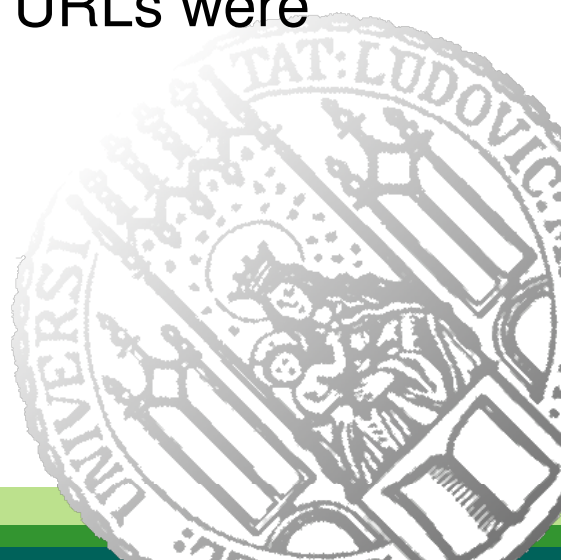
Sample graphs





Delicious

- Social network for sharing URLs
- URLs can be tagged by users
- 8 different kinds of connecting URLs were tested



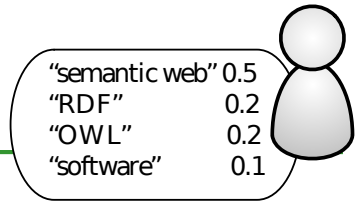
Delicious Evaluation Results

delicious
social bookmarking

Evaluation results

- Employing web-graph links improves the search result
- “Social connections” often do not lead to a better result
- Properties for datasets suitable for `PEST` were gained
  - Sensible identification of **data items**, **links** and **terms**
  - Connectivity between data items
  - Choice and manipulation of terms

PEST for Personalization (PESTP)



- Collaboration with Frederico Durao, Aalborg University (KiWi)
- Technique
  - User preferences: propagation of terms from movies to the user, based on his edited/commented/tagged movies
  - Movie search: terms get propagated between linked movies and from a user to movies
- Evaluation on a movie database
  - Improves precision and recall over PEST and tf-idf
  - Decreased runtime performance
- Paper submitted to DASFAA 2011 (Database Systems for Advanced Applications)

Outline

Conclusion

- `PEST` shows an impressive improvement of search results compared to tf-idf as well as Google
- Evaluation on several real-world datasets
- Guidance to what kinds of data are specifically suited for `PEST`
- Already the straightforward implementation yields good performance; possible improvements described in thesis

Improvements for the future

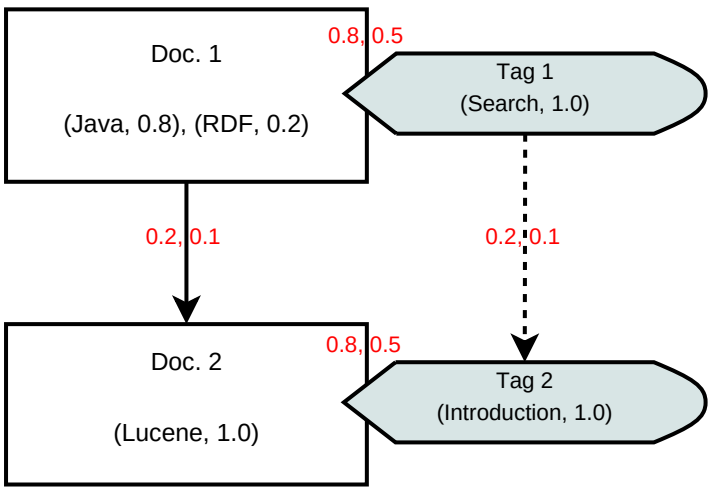
- Tuning of edge weights
- Include ontologies into the dataset
- High performance implementation

Any Questions?

Thank you for your attention!

Backup

Generation of the PEST Matrix



- Transposed adjacency matrix:

	Doc 1	Doc 2	Tag 1	Tag 2
Doc 1	0	0.1	0.8	0
Doc 2	0.2	0	0	0.8
Tag 1	0.5	0	0	0.1
Tag 2	0	0.5	0.2	0

- Normalized adjacency matrix **H**:

	Doc 1	Doc 2	Tag 1	Tag 2
Doc 1	0	0.05	0.40	0
Doc 2	0.10	0	0	0.40
Tag 1	0.25	0	0	0.05
Tag 2	0	0.25	0.10	0

- PEST matrix $\mathbf{P}_{Java}$:

	Doc 1	Doc 2	Tag 1	Tag 2
Doc 1	0.57	0.66	0.80	0.50
Doc 2	0.13	0.04	0.04	0.38
Tag 1	0.26	0.04	0.04	0.08
Tag 2	0.04	0.26	0.12	0.04

Computation of the PEST Matrix

$$\mathbf{P}_\tau = (1-\alpha)\cdot\mathbf{H} + \mathbf{L}_\tau$$

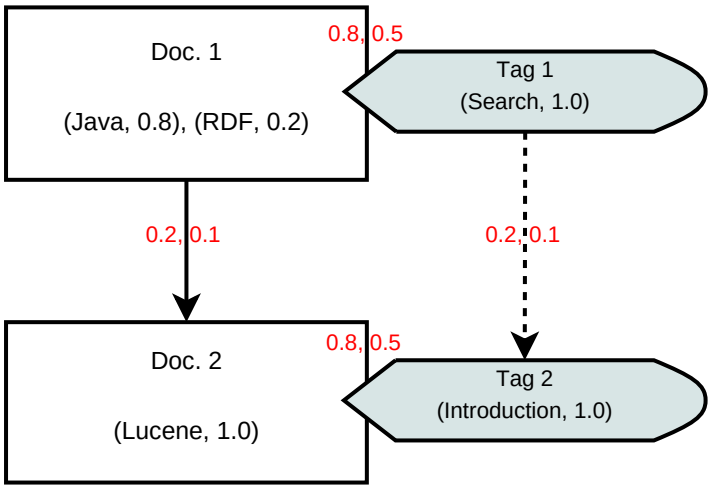
$$\mathbf{L}_\tau = \left(P(leap|j)\cdot((1-\rho)\cdot I_\tau^{inf}(i,j) + \rho\cdot I^{rnd}(i,j))\right)_{i,j}$$

$$P(leap|j) = \alpha + (1-\alpha)\cdot(1-\sum_i \mathbf{H}_{i,j})$$

$$I_\tau^{\mathrm{inf}}(i,j) = \frac{w_t(i,\tau)}{\sum_k w_t(k,\tau)}$$

$$I^{\mathrm{rnd}}(i,j) = \frac{1}{|V_d \cup V_t|}$$

Eigenvector Computation



- Resulting eigenvector after applying power method:

$$\mathbf{p}_{Java} = \begin{pmatrix} 0.62 \\ 0.12 \\ 0.12 \\ 0.08 \end{pmatrix}, \text{vs.} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- Explanation:
  Java term weight of Doc. 1 is propagated to the other nodes

$\Rightarrow$ **Fuzzy Matching** mission accomplished

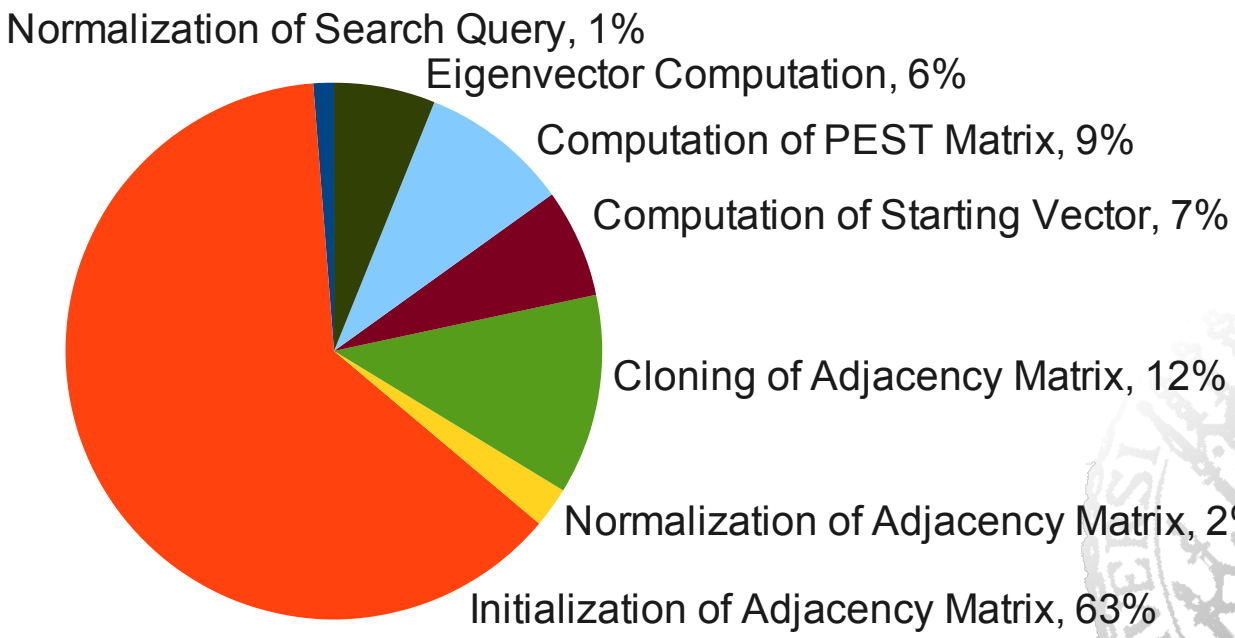
Runtime Performance in the Simpsons Wiki

In absolute terms:

Initialization of term-independent adjacency matrix: 16 sec

Total computation for one term: 8 sec

Time spent in different steps of the algorithm



Simpsons User Study - Query List

Characters

bart, homer, krusty, lisa, marge, milhouse, montgomery, ned, nelson, skinner

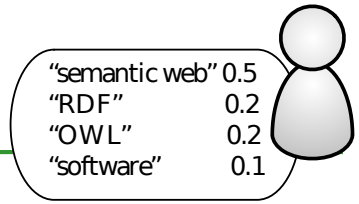
Places

brewery, lake, school, springfield, tavern

Other

beer, nuclear, pistol, retirement, skateboarding

PESTP Ranking



PEST	Movie title (gender)	Luc	PCH	Mov	PCH	PEST	PCH
1	Scary Movie 2 (comedy)	17	+16	13	+12	38	+37
5	Gremlins (comedy)	19	+14	15	+10	22	+17
25	Friday the 13th (horror)	22	-03	18	-07	20	-05
56	The Dark Knight (horror)	15	-41	24	-32	16	-41
67	Freddy vs. Jason (horror)	11	-56	40	-27	9	-54

Five movies from the PESTP ranking for query *"Scary"* issued by user with ID 4654. The ranking approaches are followed by the PESTP **ch**ange. (PCH)