Using Language Modeling for Spam Detection in Social Reference Manager Websites

Toine Bogers and Antal van den Bosch

DIR 2009

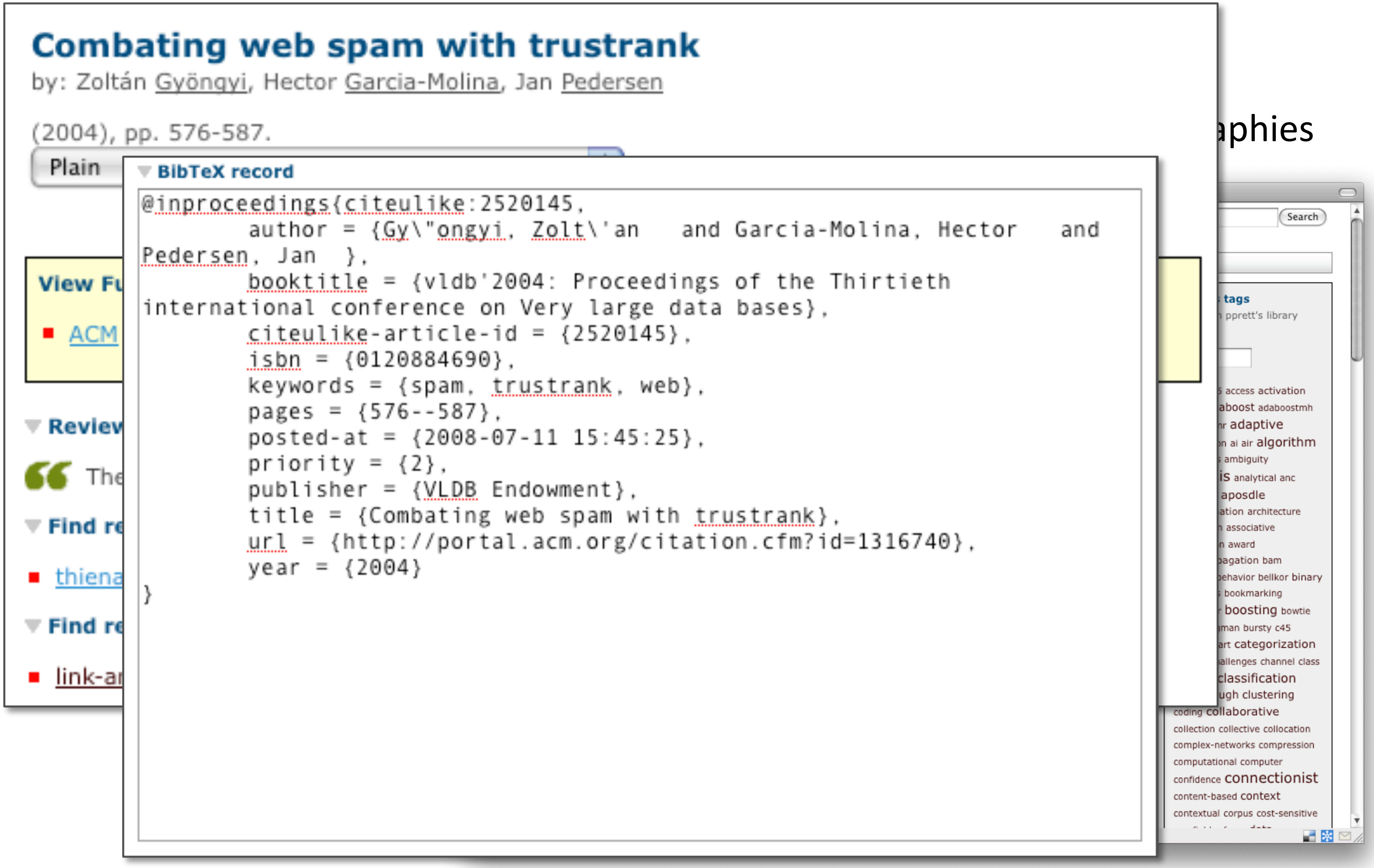
February 3, 2009

Tilburg University

Outline

- Introduction
- Methodology
- Our approach
- Results
- Discussion

Social reference managers



Spam

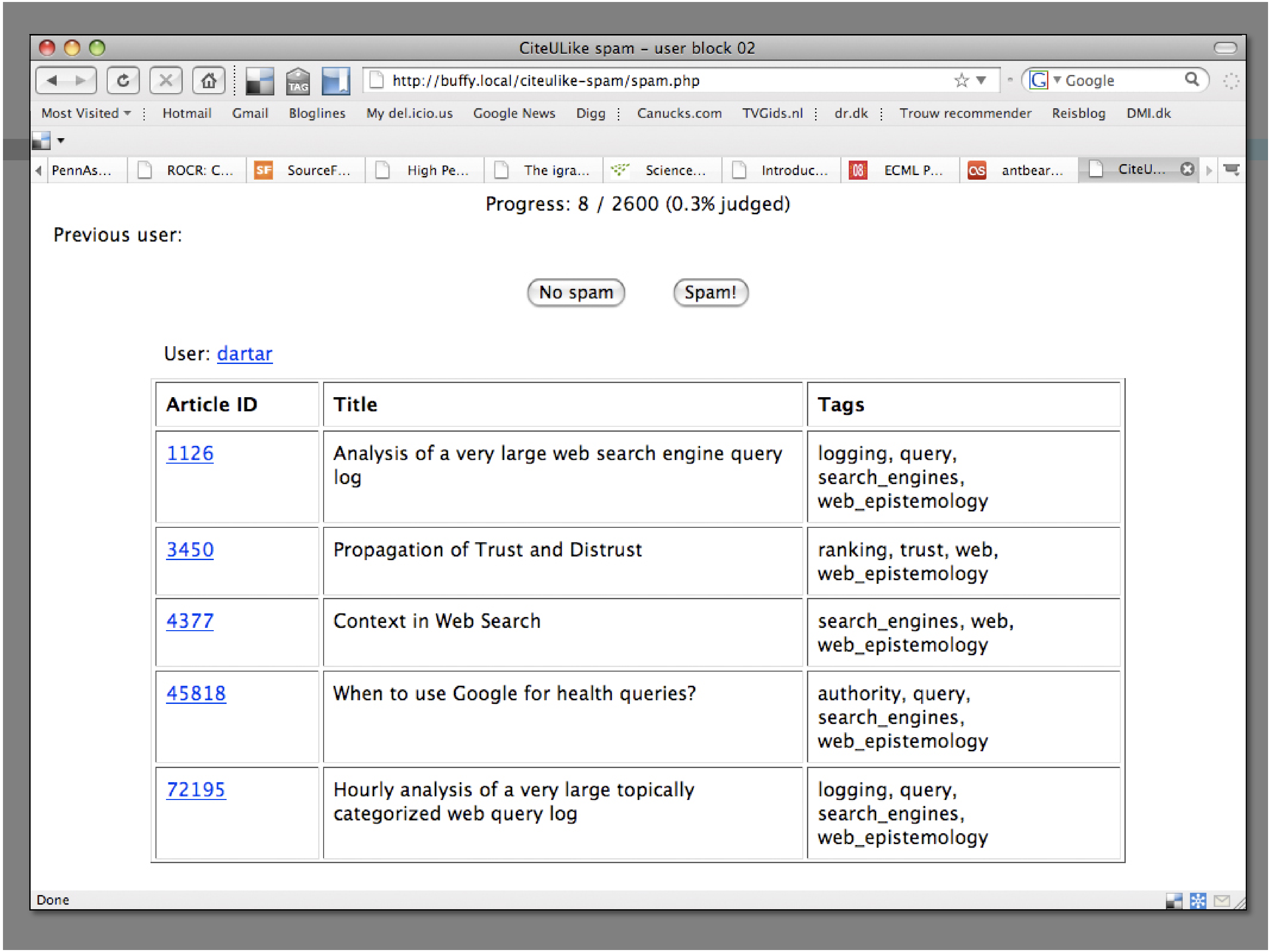
- In a social bookmarking context:
  - Users posting content and tags designed to mislead others
- Open questions
  - How big of a problem is it?
  - How harmful to which task?
  - How can we deal with it?
  - Little research done

Outline

- Introduction
- Methodology
- Our approach
- Results
- Discussion

Task

- Task definition take from the 2008 Discovery Challenge
  - Annually organized data mining competitions
  - Two tasks in 2008
    - Tag recommendation
    - Spam detection
- Spam detection task
  - Learn a model that predicts spam at the user level
  - Equal to detecting spam users
  - Organizers provided a pre-labeled data set
  - All of a spam user's posts are labeled as spam



Progress: 8 / 2600 (0.3% judged)

Previous user:

No spam Spam!

User: dartar

Article ID	Title	Tags
1126	Analysis of a very large web search engine query log	logging, query, search_engines, web_epistemology
3450	Propagation of Trust and Distrust	ranking, trust, web, web_epistemology
4377	Context in Web Search	search_engines, web, web_epistemology
45818	When to use Google for health queries?	authority, query, search_engines, web_epistemology
72195	Hourly analysis of a very large topically categorized web query log	logging, query, search_engines, web_epistemology

Data representation

- BibSonomy
  - Treated bookmarks and BibTeX the same
  - Divide the metadata into 4 different fields: **TITLE**, **DESCRIPTION**, **TAGS**, and **URL**
  - Normalized the URL (tokenization, removal of common prefixes/suffixes)
- CiteULike
  - Clean posts had metadata, but most spam posts did not
  - Used only **TAGS** metadata for a fair comparison

Example of a clean post

booktitle

author

```
<DOC>
  <DOCNO> 694792 </DOCNO>
  <TITLE>
    When Can We Call a System Self-Organizing
  </TITLE>
  <DESCRIPTION>
    ECAL Carlos Gershenson and Francis Heylighen
  </DESCRIPTION>
  <TAGS>
    search agents ir todo
  </TAGS>
  <URL>
    springerlink metapress openurl asp genre article issn
    0302 9743 volume 2801 spage 606
  </URL>
</DOC>
```

Experimental setup & evalution

- Experimental setup
  - BibSonomy: pre-defined split in training and test material
    - Official training material divided in 80-20 split on users (38,920 users)
    - 80% training set (25,372 users)
    - 20% validation set for parameter optimization (6,343 users)
    - Official test set (7,205 users)
  - CiteULike
    - 60% training set (4,160 users)
    - 20% validation set for parameter optimization (520 users)
    - 20% test set (520 users)
- Evaluation metric
  - AUC (Area Under the ROC Curve)

Outline

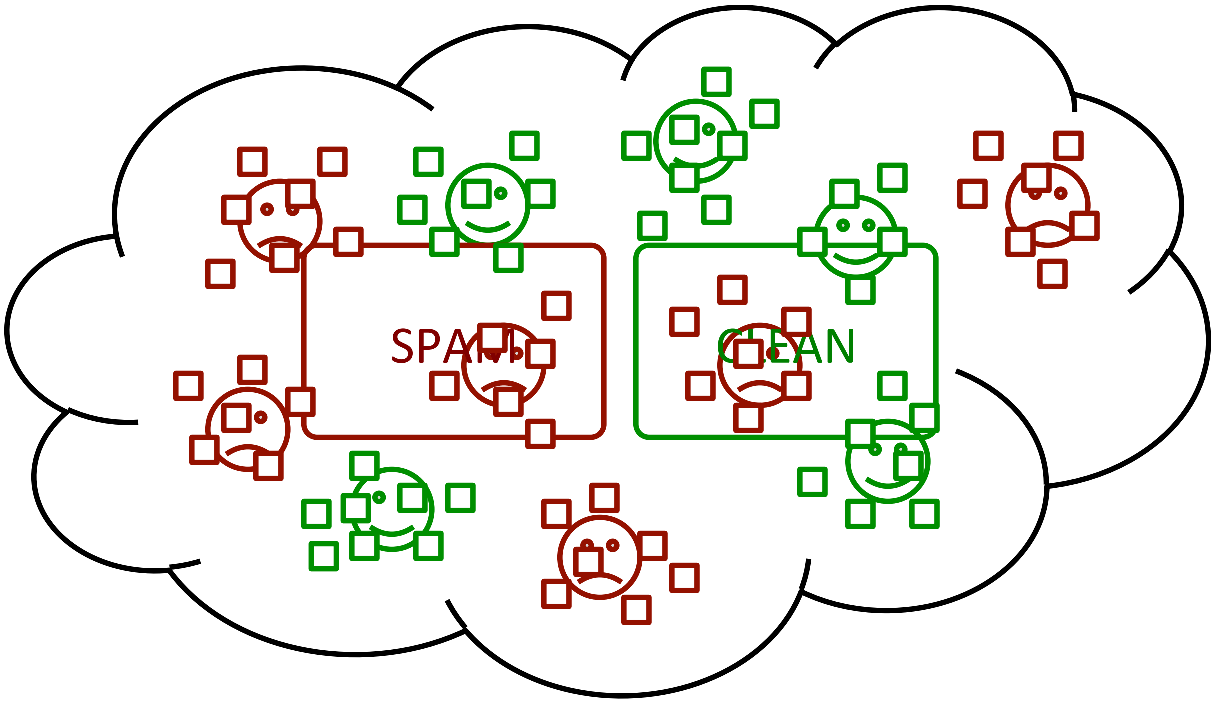
- Introduction
- Methodology
- Our approach
- Results
- Discussion

Our approach

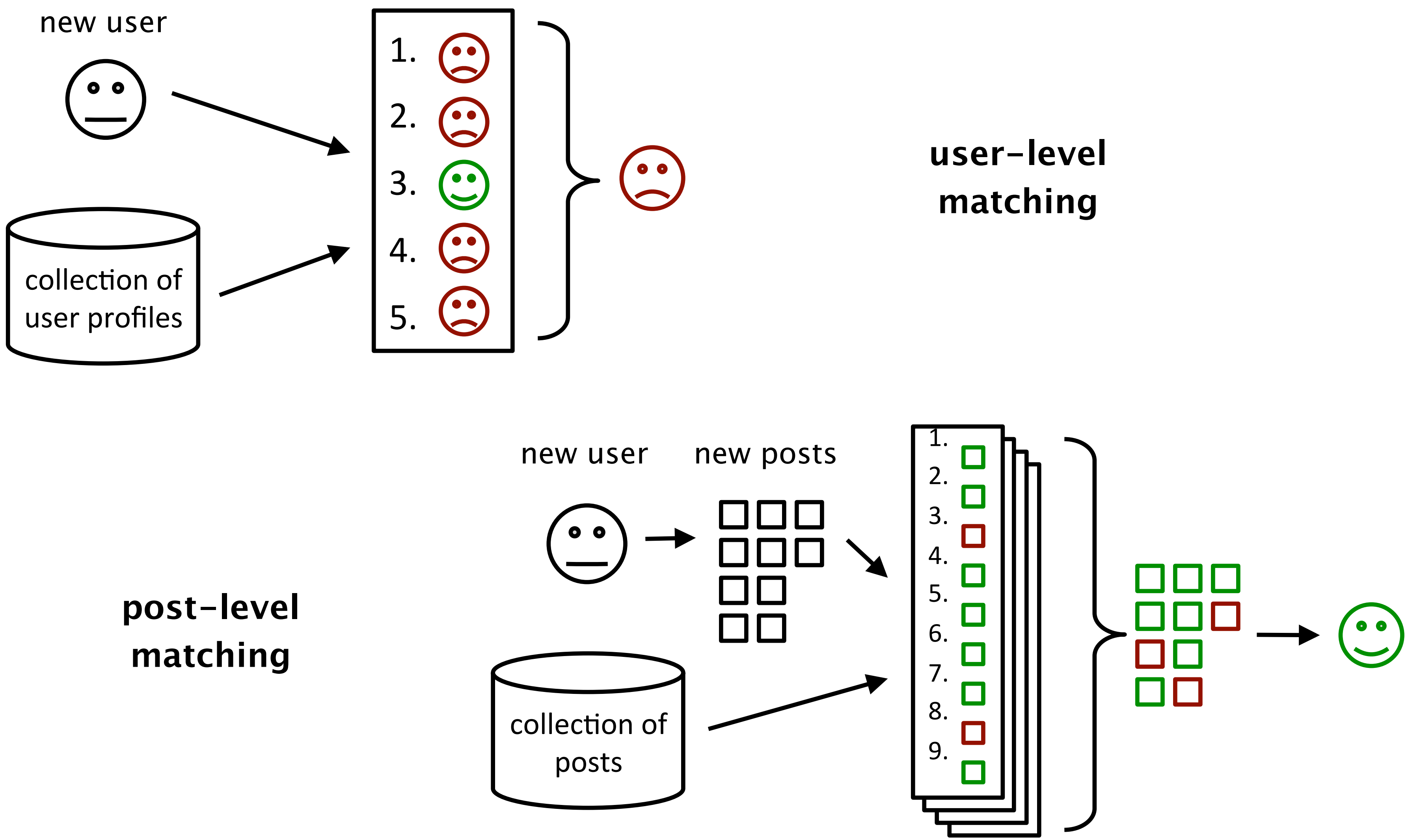
- Inspired by Mishne et al. (2005) for blog spam
- Approach based on similar language use of similar users
  - We compare language models of spam and ‘genuine’ content
- Two-stage approach
  - Determining most similar matching content using language models
  - Let the most similar matches determine the spam label

Matching language models

- At what level should we compare our language models?



Matching language models



Matching language models

- (Dis)similarity between LMs calculated using KL-divergence
  - Used Indri Toolkit for experiments
- Experimented with all fields combined and all 4 fields separately
  - 9 different matchings

collection
(training set)

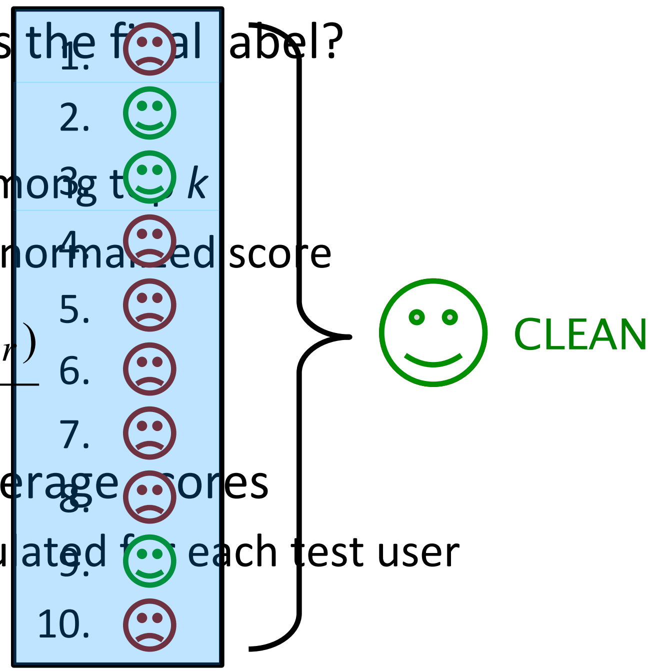
new
users/posts

Spam classification

- After the matching phase we get a normalized ranking
  – Each user/post has a score between 0 and 1 and a binary spam label
- Questions
  – How many of the top *k* matches help determine the final label?
    - Optimized on AUC, from k = 1 to k = 1000
  – How do the top *k* matches contribute towards the final label?
    - Simplest: take top label
    - A bit more sophisticated: take average label among top *k*
    - What we did: take average label, weighted by normalized score

$$score(u_i) = \frac{\sum_{r=1,r\neq i}^{k} sim(u_i, u_r) \cdot label(u_r)}{k}$$

  – At the post level we get per-post weighted average scores
    - Simple average of per-post scores is then calculated for each test user



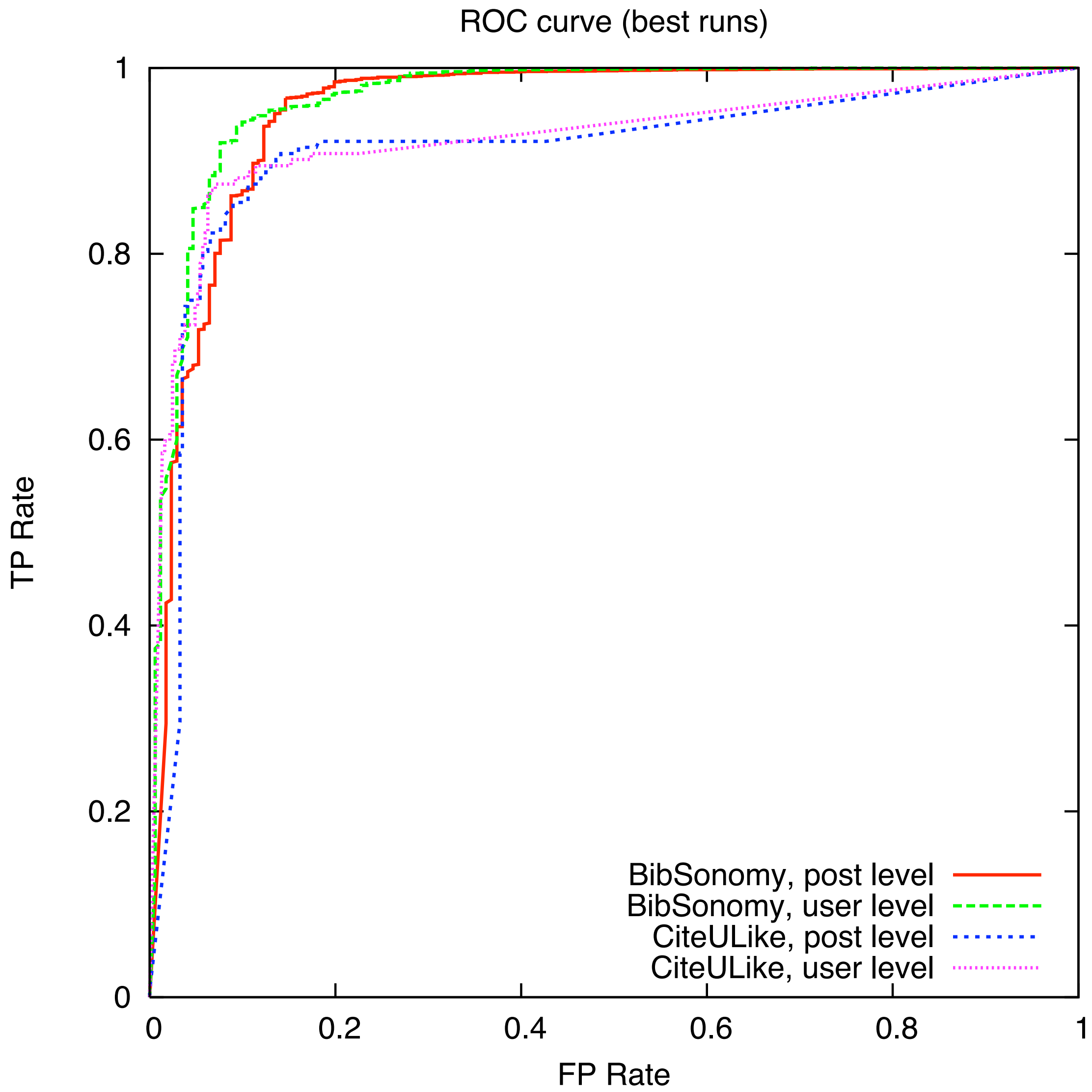
Outline

- Introduction
- Methodology
- Our approach
- Results
- Discussion

Results

		User level			Post level		
Collection	**Fields**	**Validation**	**Test**	**k**	**Validation**	**Test**	**k**
BibSonomy (matching fields)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9290	0.9450	150	0.9055	0.9287	45
	description	0.9055	0.9452	100	0.8802	0.9371	100
	tags	**0.9724**	0.9073	110	**0.9614**	0.9088	60
	URL	0.8785	0.8523	35	0.8489	0.8301	8
BibSonomy (single fields in evaluation sets)	all fields	0.9682	**0.9661**	235	0.9571	**0.9536**	50
	title	0.9300	0.9531	140	0.9147	0.9296	50
	description	0.9113	0.9497	90	0.8874	0.9430	75
	tags	**0.9690**	0.9381	65	**0.9686**	0.9251	95
	URL	0.8830	0.8628	15	0.8727	0.8369	15
CiteULike	tags	**0.9329**	**0.9240**	5	**0.9262**	**0.9079**	5

Results



ROC curve (best runs)
TP Rate
FP Rate
0
0.2
0.4
0.6
0.8
1
BibSonomy, post level
BibSonomy, user level
CiteULike, post level
CiteULike, user level

Outline

- Introduction
- Methodology
- Our approach
- Results
- Discussion

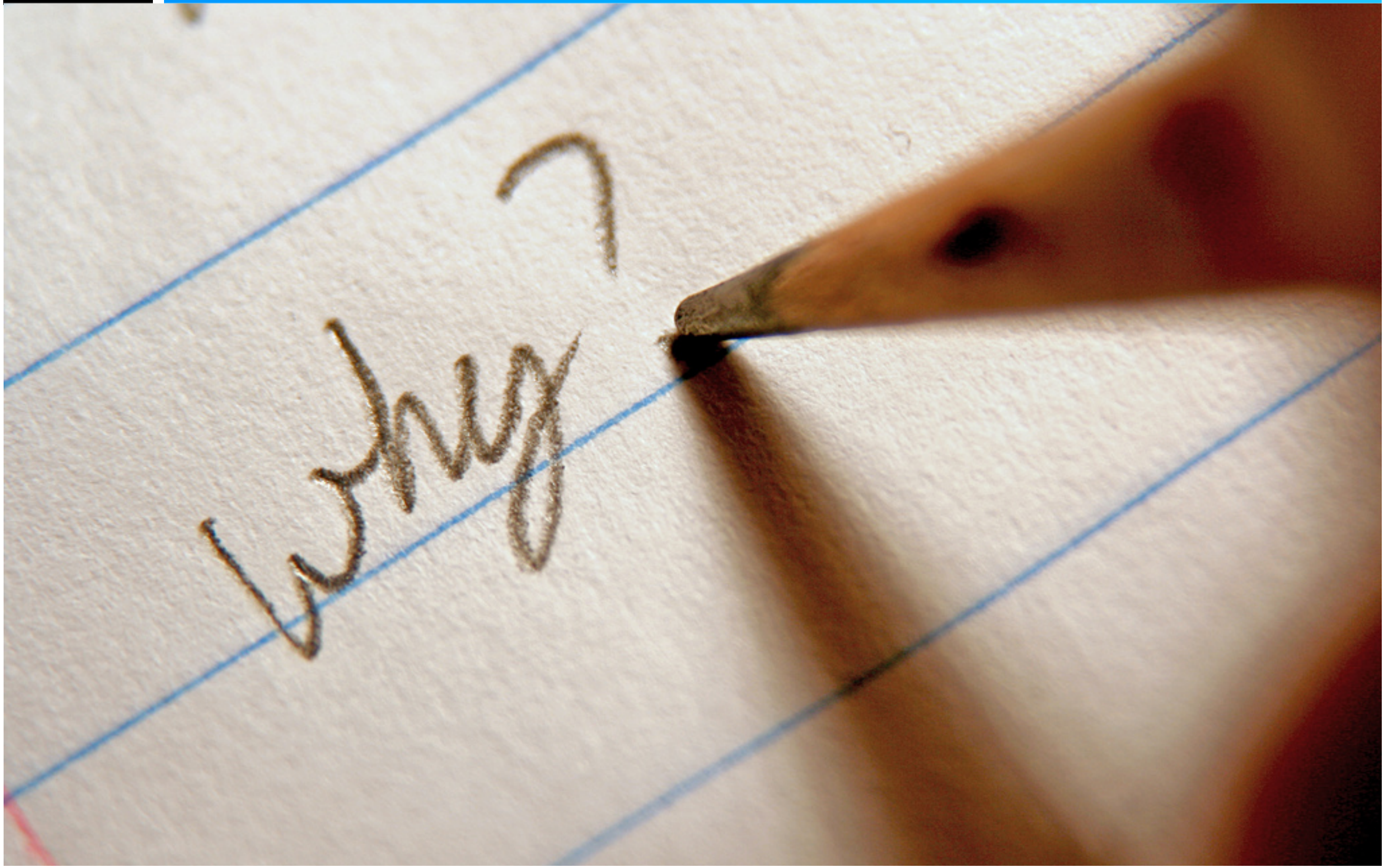
Discussion

- Straightforward approach with >90% score
- User-level detection works better than post-level detection
  - Spam labels assigned at the user-level
  - Users are a better aggegration level; less sparse
- Using only matching fields performs slightly lower than all collection fields
  - Probably because of less data
  - Using all fields is the overall best approach on (the test set)
- Approach works well on both data sets
- Easy to implement on top of existing search engine

Comparison with related work

- Comparison to other Discovery Challenge submissions
  - Eight participants scored over the baseline
  - Score of 0.9661 would have achieved third place
  - Four SVM approaches; one better then ours
  - Ridge regression approach performed better than ours
  - Naïve Bayes and five other machine learning approaches performed worse

Questions? Comments? Suggestions?



Spam classification

- Not every new user has matching users/posts
  - Missing metadata or outlier users/posts
  - Only 0.7% (44 out of 6343 validation users) had no matches
  - Default prediction is ‘clean’
    - These missing users were clean in 84% of the cases in the validation set

Data sets

	BibSonomy	CiteULike
posts	2,102,509	224,987
bookmarks, spam	1,766,334	
bookmarks, clean	177,546	
articles, spam	292	70,168
articles, clean	158,335	154,819
users	38,920	5,200
spam	36,282	1,475
clean	2,638	3,725
average posts/user	54.0	43.3
spam	48.7	47.6
clean	127.3	41.6
tags	352,542	82,121
spam	310,812	43,751
clean	64,334	45,401
average tags/post	7.9	4.6
spam	8.9	7.7
clean	2.7	3.2

Example of a spam post

```
<DOC>
  <DOCNO> 2775810 </DOCNO>
  <TITLE>
    How To Build Traffic To Your Blog
  </TITLE>
  <DESCRIPTION>
    -
  </DESCRIPTION>
  <TAGS>
    blogging directory promotion traffic
  </TAGS>
  <URL>
    webpronews ebusiness sitepromotion wpn
    3 20041210HowToBuildTrafficToYourBlog
  </URL>
</DOC>
```

Future work

- Plans for the future
  - Implement and test the class-level approach
- Other possibilities
  - Use extra features like PageRank for bookmarks
  - Direct comparison on CiteULike data set with algorithms like SVMs
  - Evaluate at the post level instead of at the user level
    - But: harder to obtain such spam labeling