Leveraging Microblogs for Resource Ranking

Tomáš Majer, Marián Šimko

tomasmajer@gmail.com, simko@fiit.stuba.sk

23.01.2012, SOFSEM 2012

Institute of Informatics and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava

PeWe@FIIT
personalized web group

Microblog

- a brief form of a blog with limited size of a post, typically 140 characters of length
- sharing experiences, opinions, comments, links
- Twitter, Identi.ca, Jaiku, (Google+, Facebook)

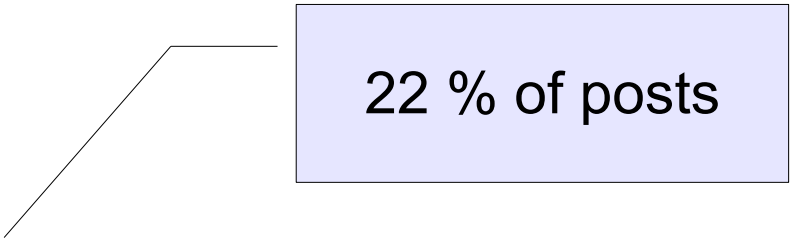
- Twitter:
  - over 300 millions of users (100 mil. at the time of writing paper)
  - over 300 millions of tweets/day (100 mil. --||--)

Microblog – why important?

- „Read-Write Web“ vision
- user-generated data
  - unbiased
  - not moderated
- huge amount of data – text, links
- valuable source of information
  - data mining

Microblog – why important?

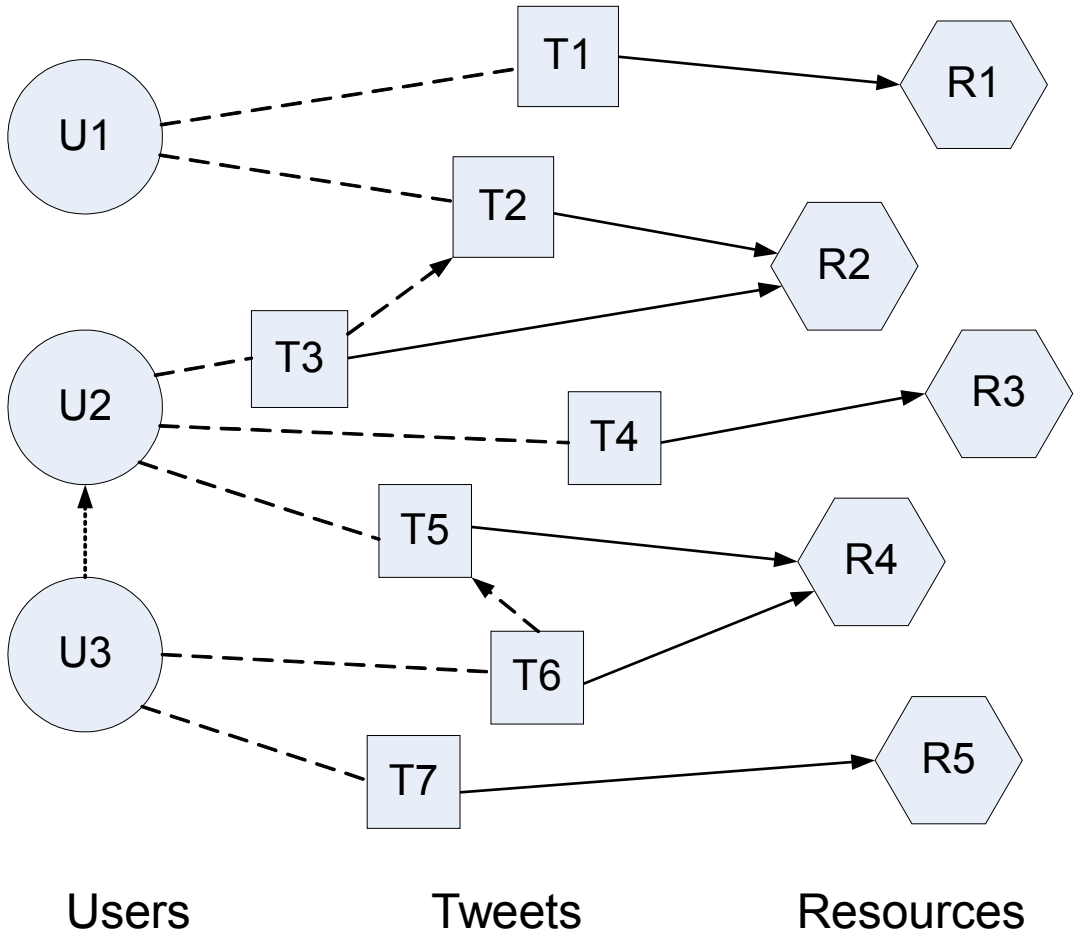
- „Read-Write Web“ vision
- user-generated data
  - unbiased
  - not moderated
- huge amount of data – text, links
- valuable source of information
  - data mining



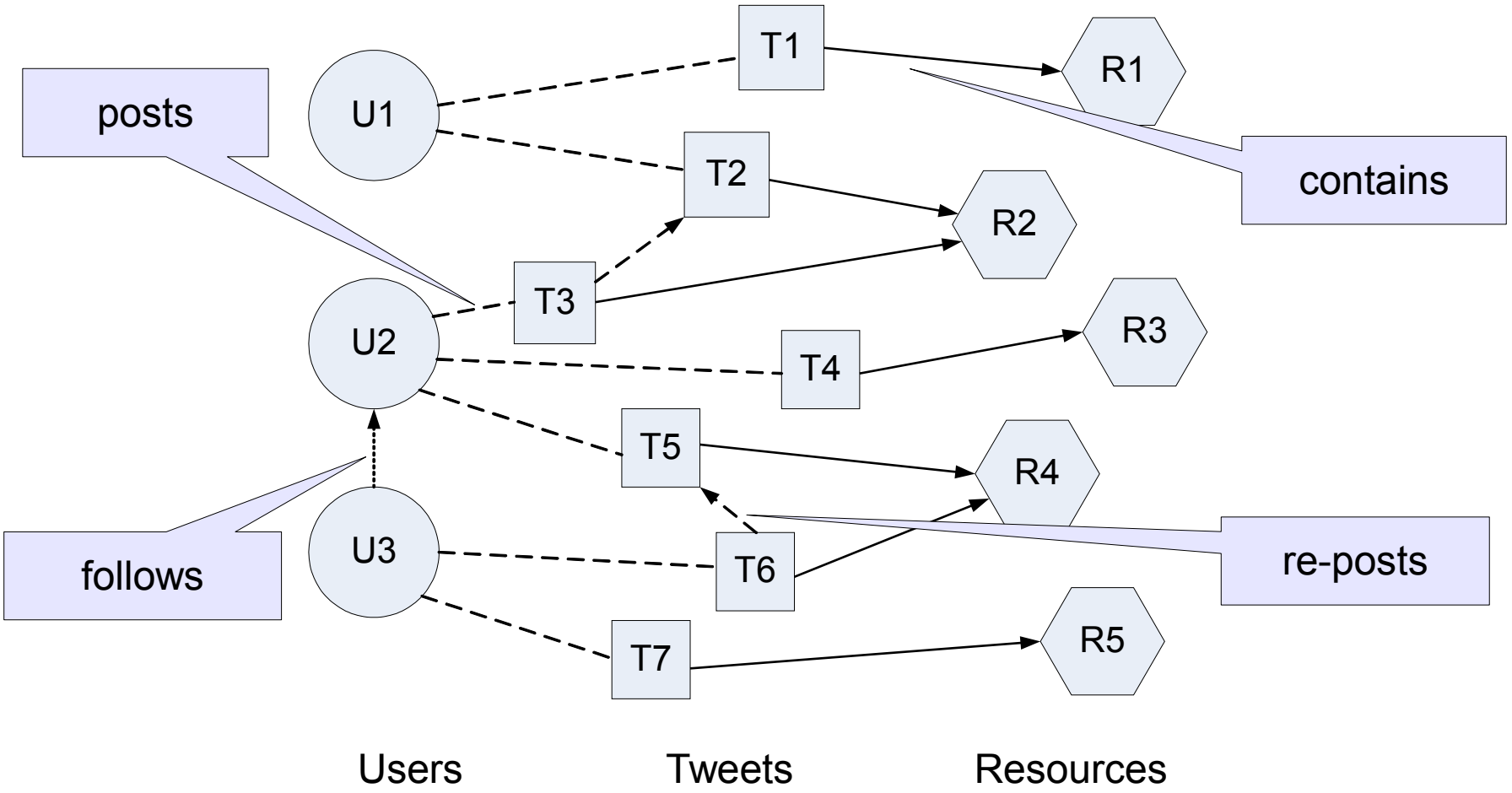
State-of-the-Art

- topic identification/keyword extraction (Ramage et al., 2010)
- opinion mining (Pendey, Iyer, 2009)
- twitter search, tweets ranking (Teevan et al., 2011)
- user ranking (Gayo-Avello, 2010)
- user modeling (Abel et al., 2011)
- …
- challenge: resource ranking

Twitter Graph



Twitter Graph



Resource Ranking: Overview

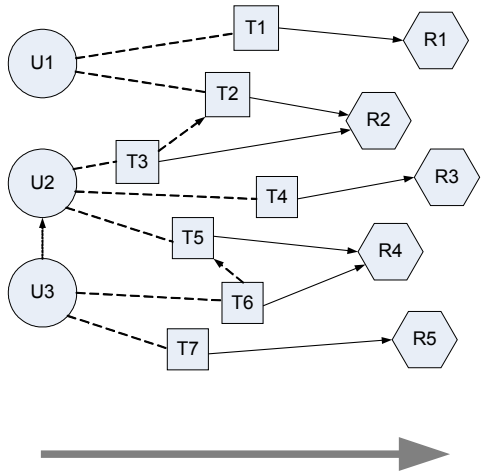
- TweetRank
  - ranking of a resource based on Twitter graph analysis

- Computation
  1. UserRank
  2. TweetRelevance
  3. TweetRank

Resource Ranking: Overview

- TweetRank
  - ranking of a resource based on Twitter graph analysis

- Computation
  1. UserRank
  2. TweetRelevance
  3. TweetRank



UserRank

$$UserRank(u) = \sum_{f \in followers(u)} \frac{1 + \gamma(u) UserRank(f)}{|followers(f)|}$$

UserRank

$$\gamma(u) = \frac{|followers(u)|}{|tweets(u)|}$$

$$UserRank(u) = \sum_{f \in followers(u)} \frac{1 + \gamma(u)\, UserRank(f)}{|followers(f)|}$$

TweetRelevance, TweetRank

$$TweetRelevance(t) = \frac{UserRank(Author(t))}{|tweets(Author(t))|}$$

TweetRelevance, TweetRank

$$TweetRelevance(t) = \frac{UserRank(Author(t))}{|tweets(Author(t))|} = TR(t)$$

$$TweetRank(r) = \sum_{t \in tweets(r)} \left(TR(t) + \sum_{rt \in retweets(t)} TR(t)\,TR(rt) \right)$$

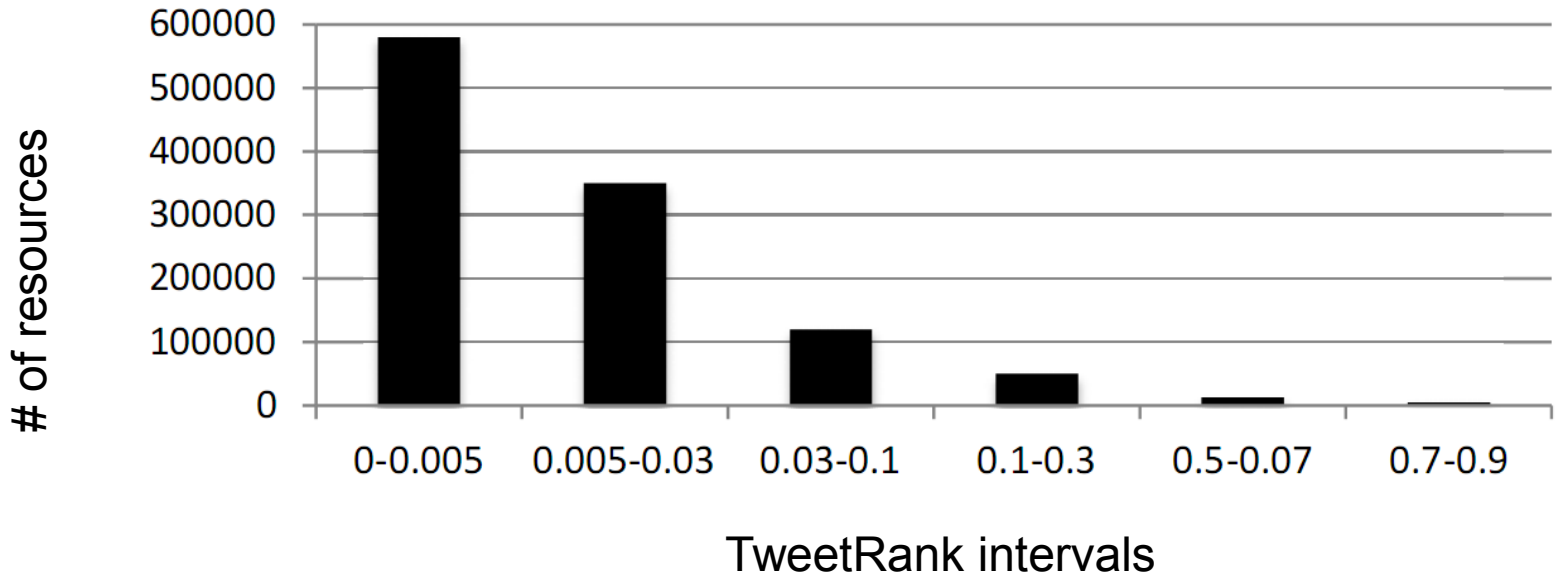
Evaluation

1. TweetRank ranking vs. explicit user ranking (*YouTube*)
2. Search results ranking study (*Search)*

- Data:
  - 1,997,466 tweets from 367,824 users
    - 85 % in English
  - 1,468,365,182 connections between 40,103,281 users
  - 1,150,168 unique web links
    - 3 % of them: YouTube

Computing TweetRank

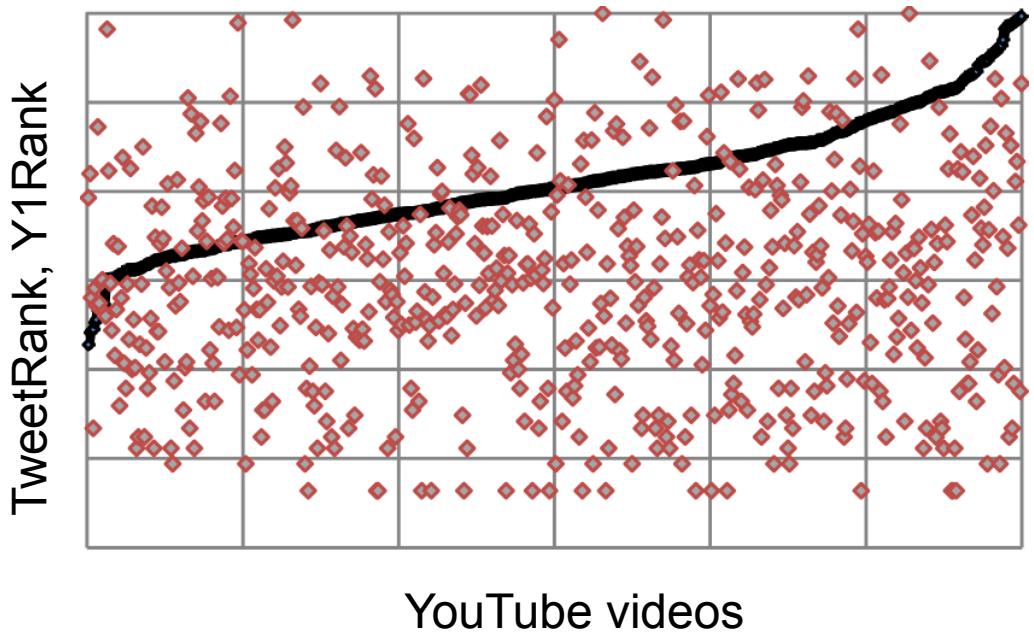
- TweetRank computed for each resource
  - power-law distribution



Experiment YouTube

- YouTube – explicit user rating (Y1Rank)
  - positive/negative vote, normalized

- TweetRank vs. Y1Rank
  - correlation coefficient
    r = 0.02

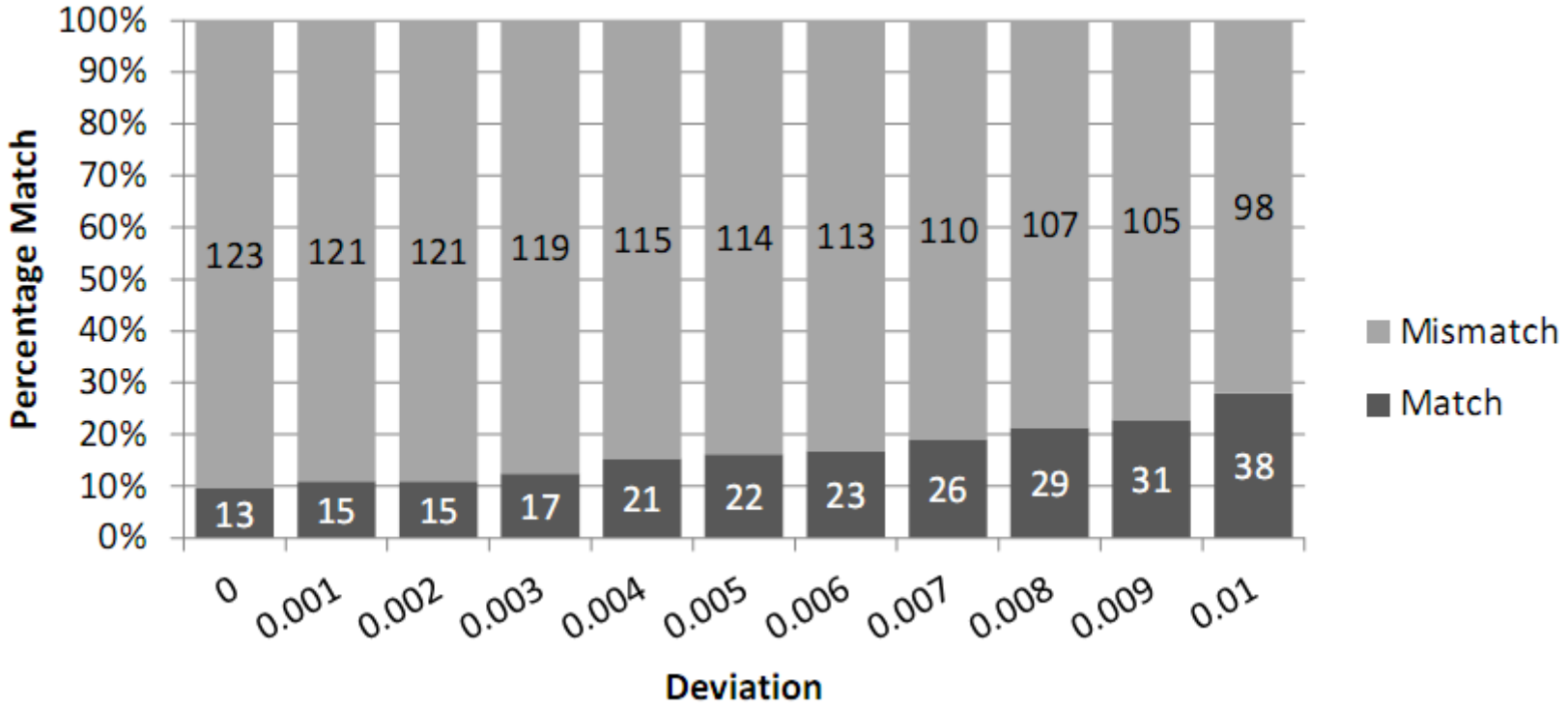


Experiment YouTube 2

- YouTube – application-collected user rating (Y2Rank)
  - „how do you like the video“?
  - 5 degree scale (1-best, 5-worst), 70 participants
- TweetRank vs. Y2Rank
  - Kendall rank correlation coefficient $\tau = 0.125$
- Relative video ranking

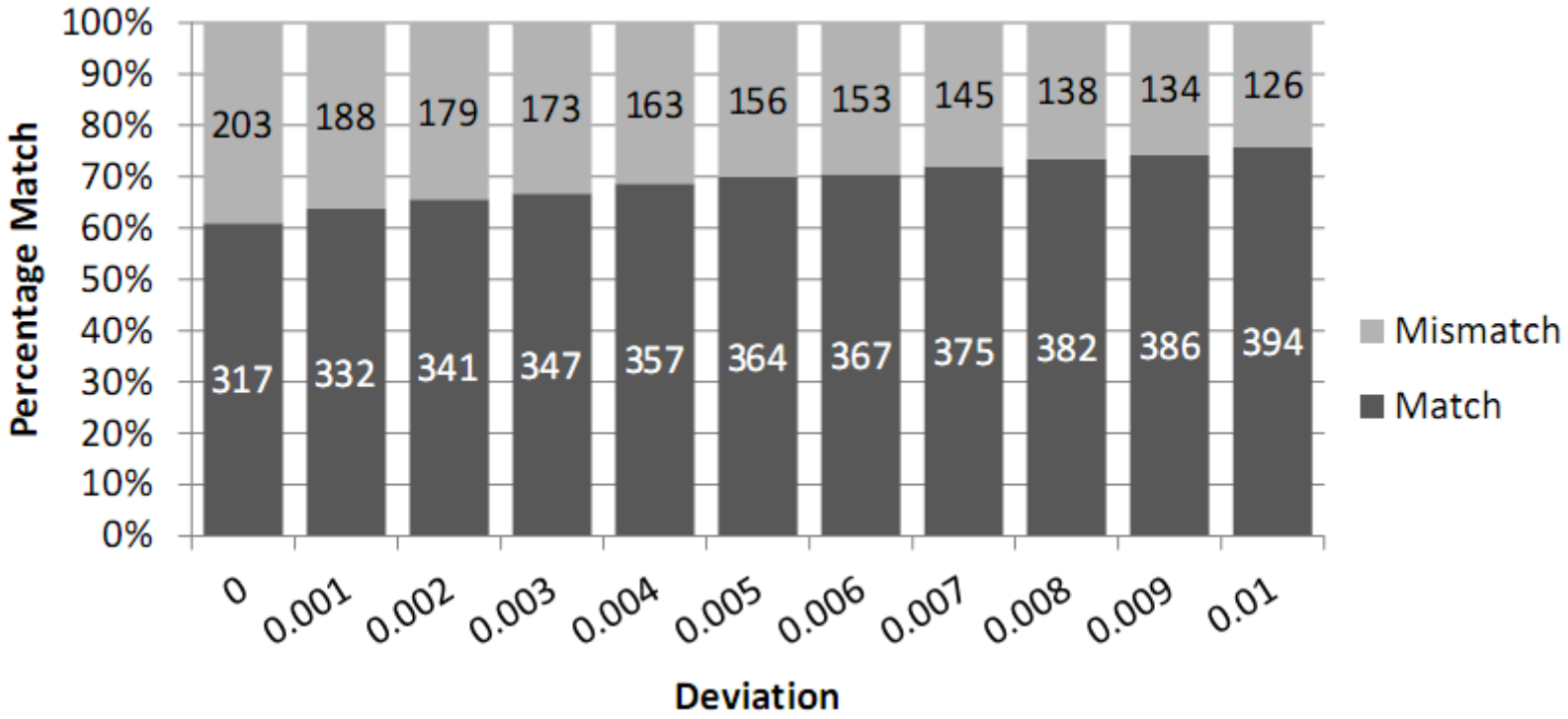
Experiment YouTube 2

- 5-tuplets



Experiment YouTube 2

- pairs



Experiment Search

- 20,000 resources
- indexing: SOLR (Apache Lucene)
- searching: resource ranking *extended* with TweetRank

- search results manual comparison
  - 20 randomly selected queries (e.g.: „apple“)
  - analyzed top-*k* results

Experiment Search

- findings:
  - in general, „newer“ resources rank better
  - ranking does not reflect chronological ordering of resources

- suitable for sorting search results within a predefined time window

Conclusions

- **microblog**
  - perspective source of data, information, knowledge
- we proposed **novel method for resource ranking** leveraging microblog network
- an important **additional knowledge** from the crowd
  - a form of indirect explicit user rating
- great **potential for search improvement**
  - reflects temporal characteristics (not linearly)
  - sorting results within a predefined time window