Managing Collaborative Feedback Information for Distributed Retrieval

Pascal Felber, Toan Luu, Martin Rajman, Étienne Rivière
Université de Neuchâtel
Ecole Polytechnique Fédérale de Lausanne
Outline

• Motivation
• Collaborative Feedback based Retrieval System
  – General approach
  – System architecture
• Challenges and Ongoing works
• Related works
• Conclusion
Motivation

• Many research efforts on P2P Web search
  – ~18,800 results on Google Scholar with keywords “P2P web search” in Oct 2008
• No P2P system has reached the level of quality and efficiency of centralized search engines (bootstrap problem)
• Our argument for P2P Web search:
  – Do not try to replace centralized search engines, but complement them with additional functionalities!
Our goals

Design a Collaborative (relevance) Feedback-based Retrieval System that:

• Provides a dedicated P2P storage system for aggregating the search results obtained by a community of users from a given (or possibly several) centralized search engine(s)
  – Collaborative (P2P) storage and aggregation
  – Balanced load
  – Decentralized and dynamic settings

• Uses semantic profiling
  – User profiles and aggregated (relevance) feedback information are used to better target the results (re-ranking)
General approach

- Search engine (web search, P2P file sharing, ...)
  - Raw results
  - Query
  - Query + Interest Profile
- Collaborative ranking service
  - Results

Rank according to structural information and popularity:
1. item a
2. item b
   ...
n. item n

Rank according to user-perceived pertinence:
- 50 item a'
- 43.5 item b'
- 38 item i'
- 10.5 item m
System architecture

Local:
- User Interest Profiling
  - (query, user profile)
  - Queries
  - User Interface
  - Relevance Feedback Tracker

Distributed:
- Routing
- Storage
  - Existing Search Mechanism
  - Results
  - Results
  - inserts relevance tracking information
  
(query, doc profile, doc ref)
User/Document profiles

- **Document profile**
  - Set of most representative keywords extracted from the document or from the document summary (snippet)
  - Document profiles are used to preserve the informational diversity of the relevance feedback information stored by the collaborative system.

- **User profile (for a query)**
  - Set of the most relevant expansion keywords generated for the query by the local query expansion system
  - User profiles are used by the collaborative system to filter the results generated for a given query before retrieving them

- **User/document profiles represented by Bloom filters**
  - Compact and encoded (privacy)
  - Adequate for computing the Jaccard similarity: \[
  \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}
  \]
Profile maintenance/usage

- Each time a user selects a document in the result list obtained from the collaborative system for a given query
  - the most representative keywords are extracted from the document
  - the query and the selected keywords are provided to the local query expansion system
  - the selected keywords are stored in a Bloom filter (document profile) that is added to the (query, document reference) relevance feedback transmitted to the collaborative system

- Each time a user submits a query to the collaborative system
  - the most relevant query expansion keywords are retrieved from the local query expansion system
  - the selected keywords (user profile) are stored in a Bloom filter that is associated to the query when submitted to the collaborative system.
Routing layer

- Structured P2P overlay
  - \(O(\log N)\) hops to reach the destination
  - Resilient to dynamic settings
  - Each peer holds a balanced number of query terms

- 2 calls for the application layer:
  - Request: (query, user profile)
  - Feedback: (query, docRef, document profile)

- Skewness of accesses leads to load unbalance
  - Specific, adaptive load balancing mechanisms
Routing layer: Delegation mechanism

- Links Used
- Links Not Used

Querying Application

Insert (Relevance) or Query (Q)

Storage Layer

Send (msg)

KBR

Transit (msg)

Send (msg)

KBR

P(Q) deg. level 1

Transit (msg)

Send (msg)

KBR

P(Q)

Deliver (msg)

master's copy

(master)

(updated)

(master)

(updated)

(master)

(updated)

Periodic Updates

delegate's copy

(updated)

(updated)

(updated)

delta = 2

delta = 3

delta = 3

delta = 3

Query or Relevance Tracking information Propagation

"Normal" (no caching) propagation
Storage layer

• Manages of relevance feedback provided by users
  – Queue: temporal storage for the identification of popular RFItems
  – POPstore: stores the popular-”not yet” popular RFItems.
  – Archive: store past popular Rfitems.

• Generates the result list for the submitted queries
  – Send back $k$-most similar RFitem w.r.t. users’ profile
Storage layer: Algorithm

- New RFItem
- Insert or increment freq of RFItem

- Queue:
  - Decrement freq of oldest RFItem
  - Oldest RFItem

- POPstore:
  - Oldest RFItem with freq = 0 and maxFreq ≥ Freq_{min}

- Archive:
  - Closest RFItem
  - Oldest RFItem

- Algorithm flowchart with connections and conditions.
Challenges

• User profiling
  – Deciding on the set of representative keywords.
  – Bloom filter dimensioning

• Routing layer
  – Latency and Throughput
  – Scalability and Load Balancing
  – Fault Tolerance

• Storage layer
  – Replace queue by using a probabilistic modeling for arrival time
Ongoing work

• Query log analysis (AOL)
  – 36 M lines of data
  – 10 M unique (normalized) queries
  – 19 M user click-through events
  – 0.6 M unique user ID’s

• Prototype implementation
  – Not a simulator!
  – Running on a cluster and on PlanetLab

• Evaluation
  – Retrieval quality
  – Load balance of P2P layer
Related works

• P2P Web search:
  – Mainly concentrates on comparing with centralized systems (scalability, bandwidth consumption, retrieval quality...)

• Meta search engine:
  – Do not take into account relevance feedback and user profiles.

• Search techniques based on user interest profiles:
  – Do not benefit from collaboration in user communities.

• Collaborative (social) annotations:
  – Requires annotation efforts
  – Centralized management
Conclusion

• Using a collaborative approach to complement centralized search

• Customized search result based on:
  – Users’ interest profiling
  – Popularity of users’ feedback (click through)
  – Diversity of search results

• Specially designed P2P system:
  – Leverages properties of key-based routing
  – Adaptive load balancing mechanisms