

# Managing implicit feedback to enhance search engine results

Research Day

08.11.2006

Ralph Weires

University of Luxembourg, Campus Kirchberg  
Interdisciplinary Lab for Intelligent and Adaptive Systems  
Management of Information and Net-Centric Computing Group  
6, Rue Richard Coudenhove-Kalergi, L-1359 Luxembourg

e-mail: [ralph.weires@uni.lu](mailto:ralph.weires@uni.lu)

phone: +352-466644-5412

# Overview

## Introduction

## Implicit Feedback

- Capturing
- Preprocessing
- Storage
- Usage
- Evaluation

## Possible Problems

## Next Steps

# Overview

## Introduction

### Implicit Feedback

- Capturing
- Preprocessing
- Storage
- Usage
- Evaluation

### Possible Problems

### Next Steps

## Topic & Goal

- Enhancing results of web search engines
  - Enhance the average quality of results
  - Work against „Search engine optimization“
  - Give hints (e.g., additional search terms)
  
- Include user feedback for enhancement
  
- Query types (web)
  - informational ←
  - navigational
  - transactional

## Explicit vs. implicit feedback

- Explicit feedback
  - Ask users about their opinion (e.g., ratings)
  - Attributes
    - + Rather reliable information
    - Requires user cooperation & effort
  
- Implicit feedback
  - Observe the users behaviour
  - Draw implicit conclusions
  - Attributes
    - + Easy to capture
    - + No user effort
    - Error-prone

# Overview

Introduction

## Implicit Feedback

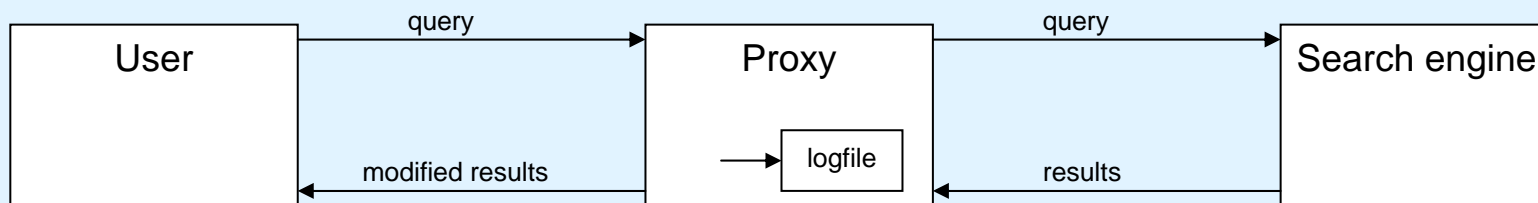
- Capturing
- Preprocessing
- Storage
- Usage
- Evaluation

Possible Problems

Next Steps

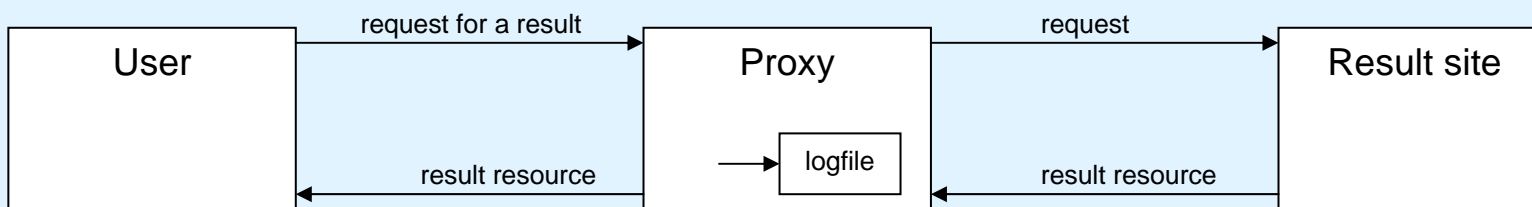
## Capturing implicit feedback (1)

- Search engine-like website
  - Proxy in between a "real" search engine and user
- Capture
  - queries
  - clicked results
  - timestamps
- Search query



## Capturing implicit feedback (2)

- Result query



## Preprocessing

- Divide raw data into search sessions, containing
  - one or more queries (query chains)
  - results (partly)
  - clicked results
  - duration of result visits
  - additional information (e.g., synonyms)

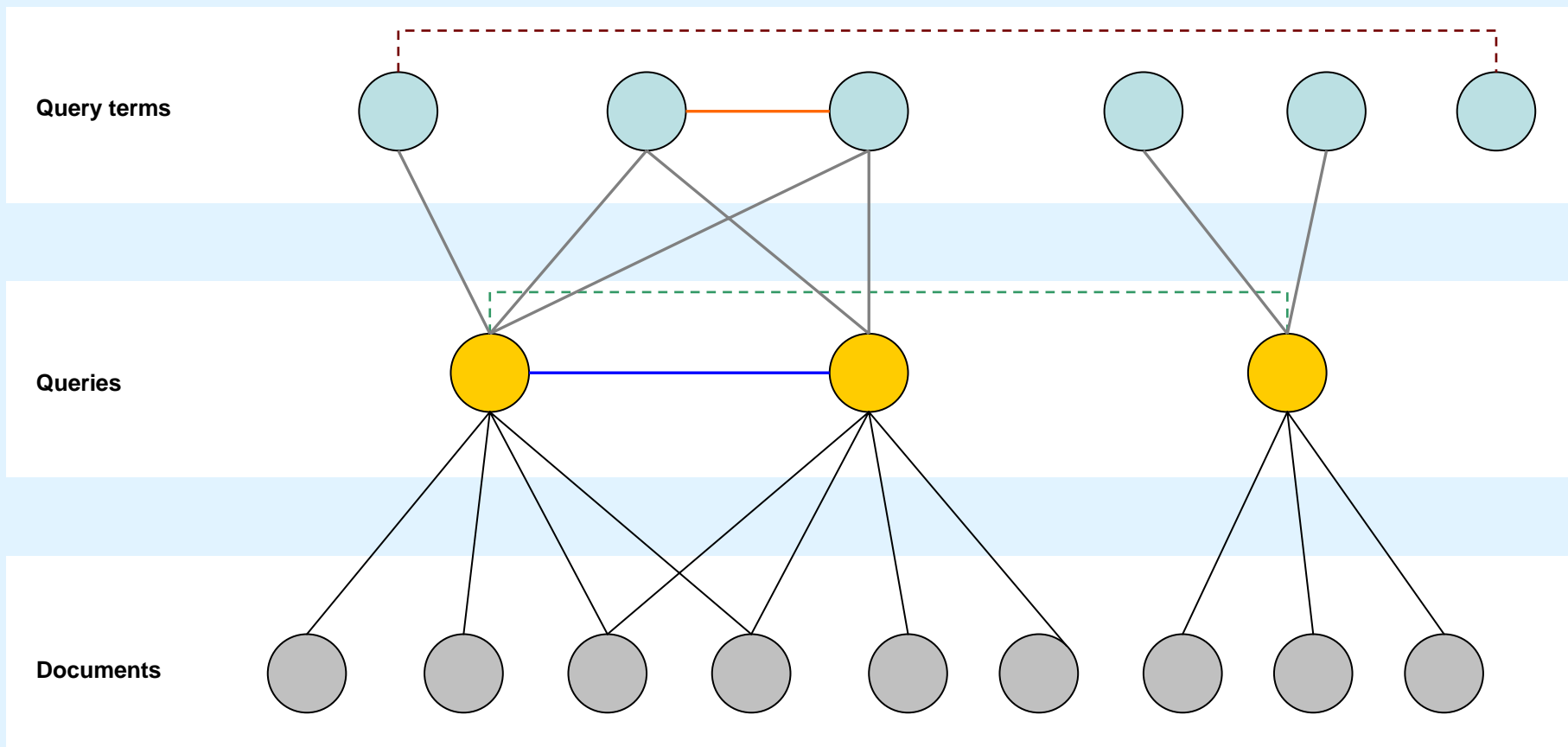
## Storage (1)

- Anima (one approach)
  - Main idea:
    - analyze streaming data
    - transaction-based
    - find characteristic patterns
    - Don't store everything, but only the "most important" information
  - Example for Anima

## Storage (2)

- Different types of nodes
  - Query terms (QT)
  - Queries (Q)
  - Documents (D)
  
- Different types of connections
  - QTxQT: co-occurrence, semantic relations (e.g. synonyms)
  - QTxQ: containment
  - QxQ: syntactic similarity (terms) [?], semantic similarity (query chains)
  - QxD: results
  
- Connections may be weighted (depending on their type)

# Storage (3)



## Usage of the stored data

- Enhance results of further queries by
  - Considering related query terms (co-occurrences, synonyms) and query chains
    - Propose additional terms
    - Automatically use additional/alternative terms
  - Including usage information (clicked links, duration of visits)
    - Rerank the results

# Evaluation

- Classical IR measure
  - Precision/Recall
  - Precision: Ratio of relevant documents in the result
  - Recall: Ratio of found relevant documents to all relevant documents
- Web IR
  - Recall is hardly important
  - Precision is what we want
- Some typical measures
  - $P@k$  ( $P@1, P@5, P@10, \dots$ )
  - AP, MAP
  - R-Precision
  - bpref

# Overview

## Introduction

## Implicit Feedback

- Capturing
- Preprocessing
- Storage
- Usage
- Evaluation

## Possible Problems

## Next Steps

## Possible Problems (1)

- Technical issues
  - How to create sessions (query chains)?
  - How to define result visiting duration?
  - Definition of appropriate storage parameters
  
- Security/Reliability
  - How to prevent misuse
  
- Using external search engines for this task
  - Limited access
    - e.g. 1000q/day
  - Legality?
    - depends on the usage terms, generally not
    - automatic querying and result processing problematic

## Possible Problems (2)

- Possible solutions
  - Search for engines which allow such usage (or build one)
  - Only use such a system in a closed environment
    - personalized, intranet
    - but probably too few data to be useful
  
- Alternative: browser plugin
  - analyze information whenever search engines are used
  - no automatic querying needed (legality still not clear, though)
  - Advantages:
    - Assignment of queries to users over multiple sessions
    - Tabbed browsing (what is a user really looking at, more reliable information)
  - Drawbacks:
    - more intrusive
    - (increased) privacy concerns
    - deployment of such software has to be secure

# Overview

## Introduction

## Implicit Feedback

- Capturing
- Preprocessing
- Storage
- Usage
- Evaluation

## Possible Problems

## Next Steps

## Next steps

- Finish the capturing software
- Get people to use the system for getting test data
- Record a (raw) data set first (completely)
  - Manual analysis
  - Try out different separation methods for the transactions
  - Make analyses with different storage parameters