



So how can I ask for it?

Prof. Wolfgang Nejdl

L3S and University of Hannover, Germany

<http://ww.l3s.de/>

<http://ww.kbs.uni-hannover.de/~nejdl>



Overview

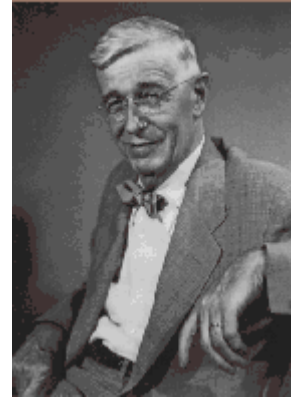
Searching for something - beyond documents ...

1. Searching Personal Collections
2. Keyword Search with Structural Feedback
3. Exploiting Tags for Music Recommendation

1. Searching Personal Collections

Memex

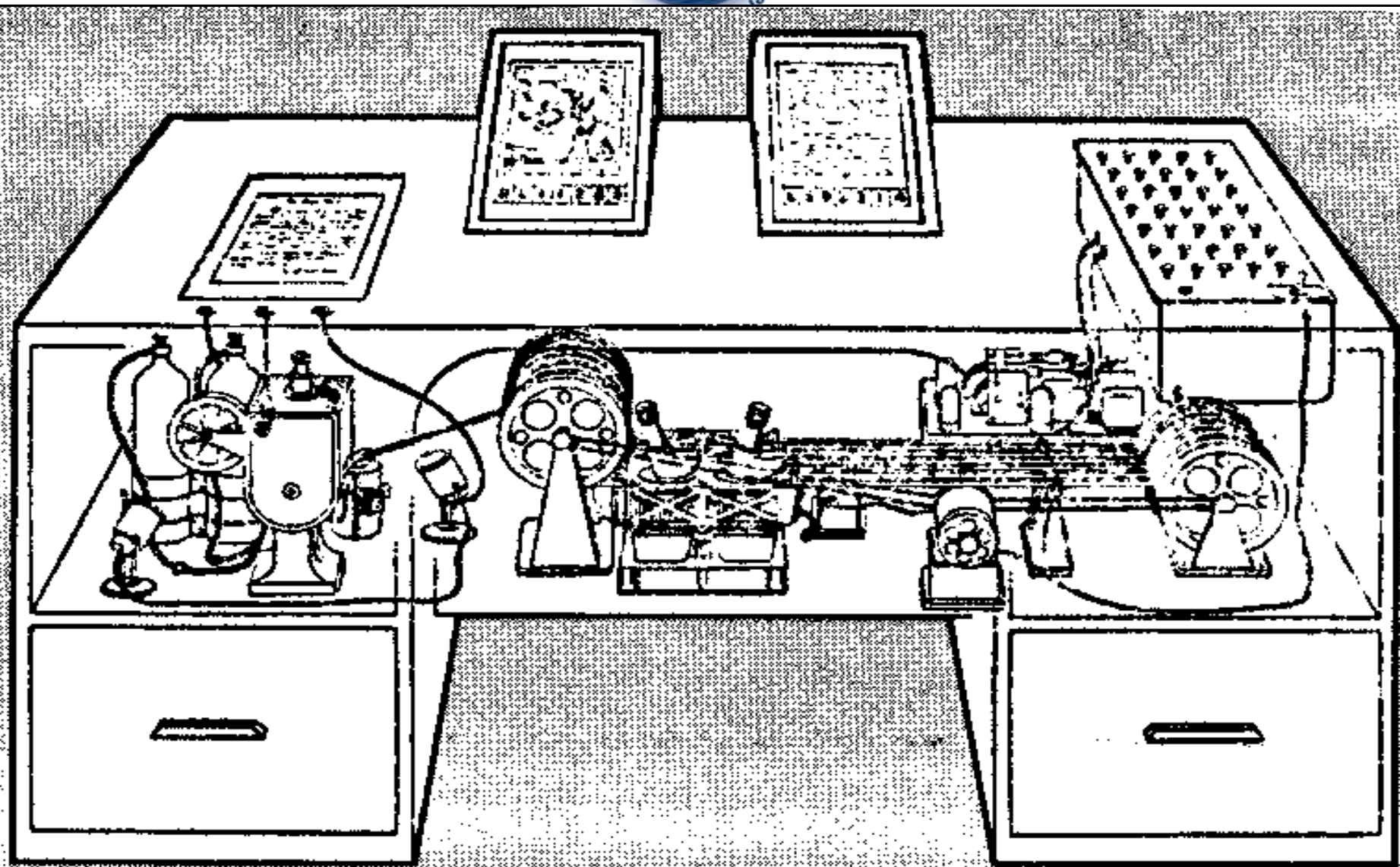
*Posited by Vannevar Bush in "As We May Think"
The Atlantic Monthly, July 1945*



“A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility”

Supports: Annotations, links between documents, and “trails” through the documents

“yet if the user inserted 5000 pages of material a day it would take him hundreds of years to fill the repository, so that he can be profligate and enter material freely”



The 1 TB Life (Gordon Bell)

1TB gives you 65+ years of:

- 100 email messages a day (5KB each)
- 100 web pages a day (50KB each)
- 5 scanned pages a day (100KB each)
- 1 book every 10 days (1 MB each)
- 10 photos per day (400 KB JPEG each)
- 8 hours per day of sound - e.g. telephone, voice annotations, and meeting recordings (8 Kb/s)
- 1 new music CD every 10 days (45 min each at 128 Kb/s)

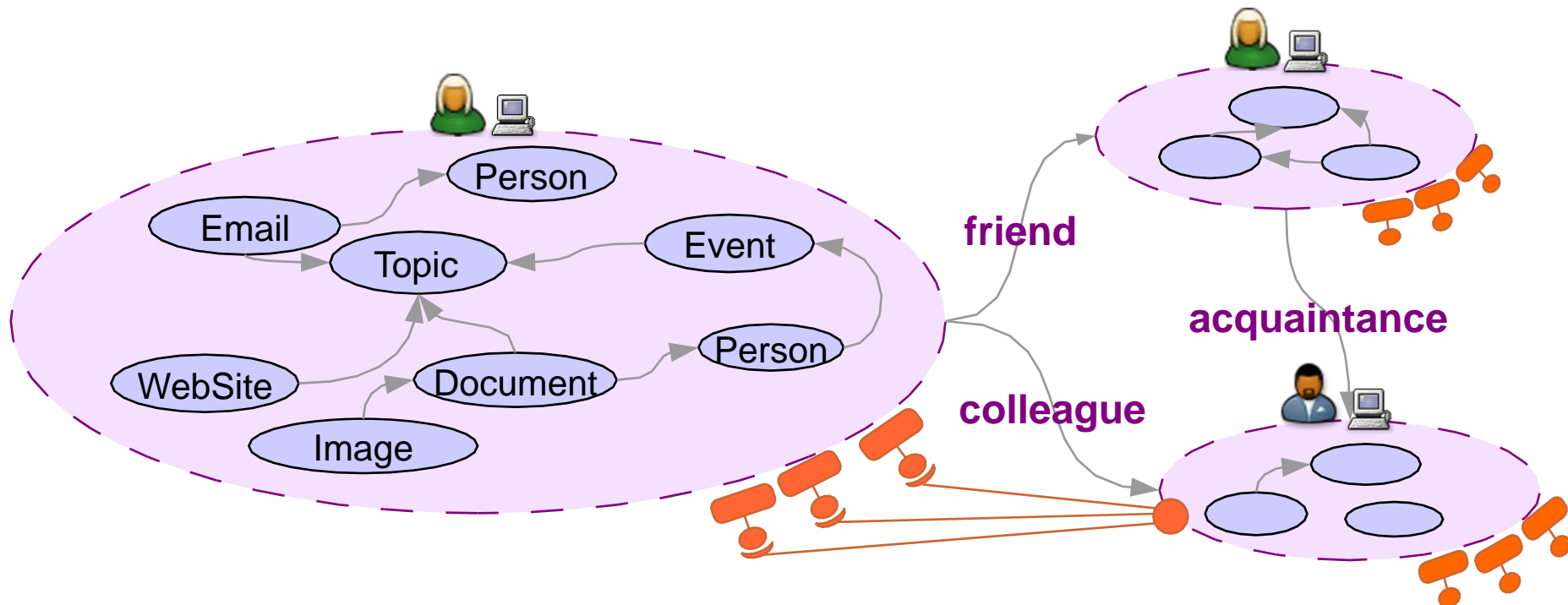


It will take you 10 years to fill up your 160 GB drive

Want video? Buy more cheap drives (1 TB/year lets you record 4 hours/day of 1.5 Mb/s video)



- NEPOMUK - Desktop:** Help individuals in managing information on their PC
- Semantic:** Make content available to automated processing
- Social:** Enable exchange across individual boundaries

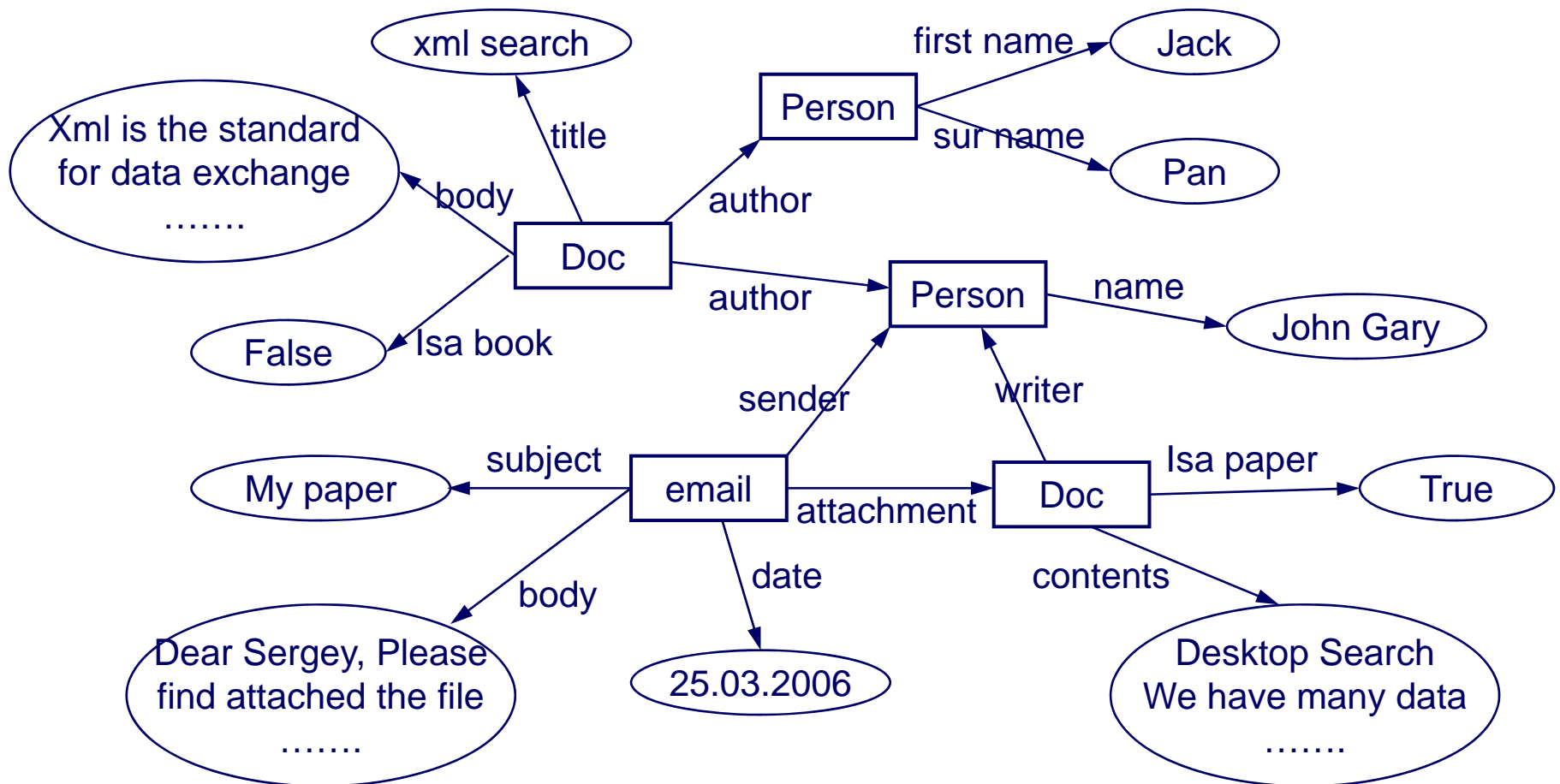


Personal Semantic Web: a semantically enlarged intimate supplement to memory

Social protocols and distributed search

NEPOMUK enabled peers

2. Keyword Search with Structural Feedback





Database Usability [Jagadish SIGMOD 2007]



Challenge: Unknown Schema

Conceptually simple real-world objects are normalized into a number of tables



○ for \$a
 \$s
let \$b
where
 \$s/co
 \$b/a
return {



amazon" and

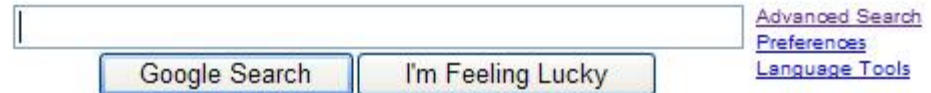
[Jagadish SIGMOD 2007]

Information Retrieval Approach

i.e. Keyword Search

Advantages:

- Intuitive
- Highly Flexible
- Easy to Use



However:

- Ignoring structured information
- Limited expressiveness → Limited ability in catching users' intents
- Example: "essay George Bush"
Essay *about* George Bush **or** Essay *written by* George Bush ?

Querying the IMDB Database

•User Intent

- Movie „Hot Fuzz“, directed by Wright. Action takes place in London.



•Query

- „Fuzz London Wright“

•Ambiguous Result Examples:

- “Run Ronnie Run”
 - Actor Wright and a character „Fuzz“
- “One life to live”
 - Actors: Mary Wright, Cindy London
- ...



...

•81 combinations in the IMDB

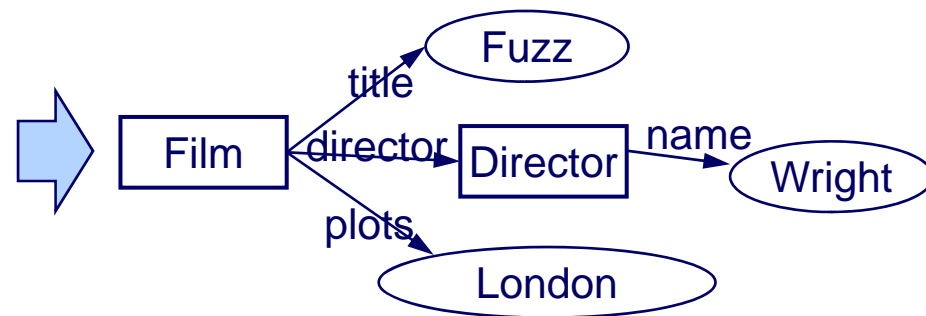
The SUITS Approach

Let user issue a keywords query in the beginning
 “Fuzz Wright London”

Help user construct a structured query through interactions

System feedback:

- | | |
|------------------------------|-----|
| Is Fuzz in movie title? | Yes |
| Is Wright in movie title ? | No |
| Is Wright a name? | Yes |
| Is Wright a director’s name? | Yes |
| Are you looking for a film? | Yes |
| Should London be in plots? | Yes |



Suits Interface

SUITS Database Search

Terms Top 10

1

```

graph LR
    plot3["plot: 3  
plottext: LONDON"]
    movie2["movie: 2  
title: FUZZ"]
    director0["director: 0  
name: WRIGHT"]
    directs1["directs: 1"]

    plot3 --> movie2
    movie2 --> director0
    movie2 --> directs1
    director0 --> directs1
    
```

director: 0
name: WRIGHT → directs: 1 ← movie: 2
title: FUZZ

FUZZ

imdb_plots.plottext imdb_movies.title

imdb_movies2actors.as_character imdb_actors.name

WRIGHT

imdb_plots.plottext imdb_actors.name

imdb_directors.name imdb_movies.title

imdb_movies2actors.as_character

LONDON

imdb_plots.plottext imdb_movies.title

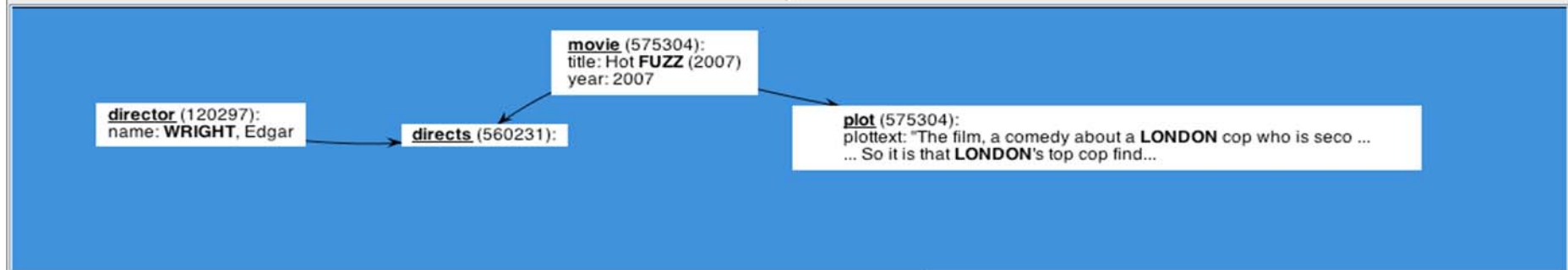
imdb_actors.name imdb_movies2actors.as_character

imdb_directors.name

Partial Queries

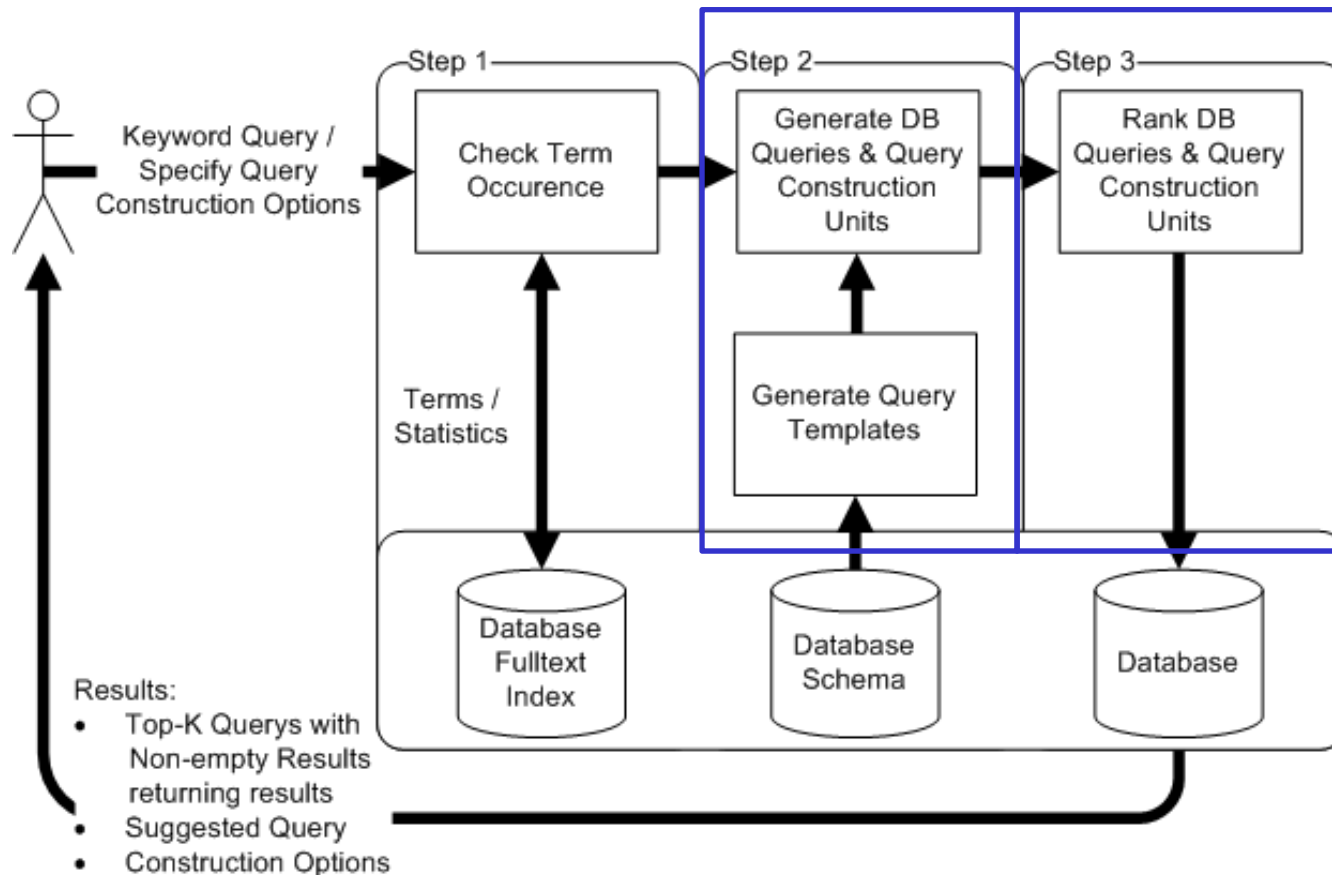
directors(WRIGHT).directs.movies(FUZZ).acts.actors

directors(WRIGHT).directs.movies(FUZZ)



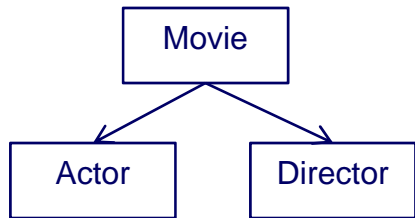
SUITS Architecture

Key Issues

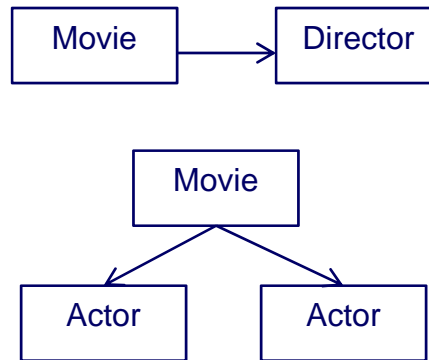


Constructing Structured Queries

Keywords: **Tom hanks**

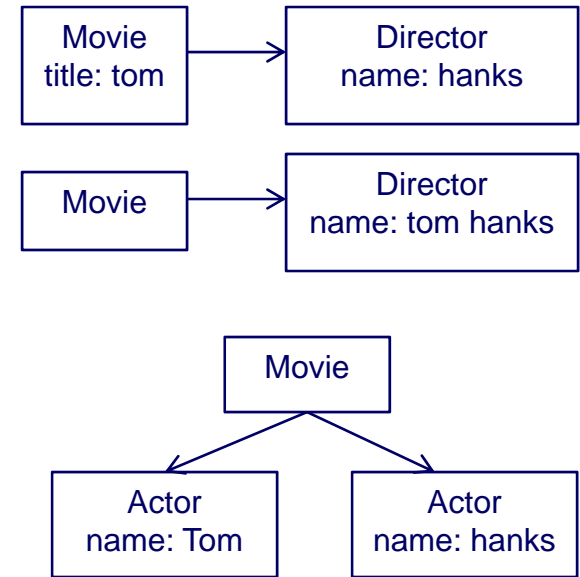
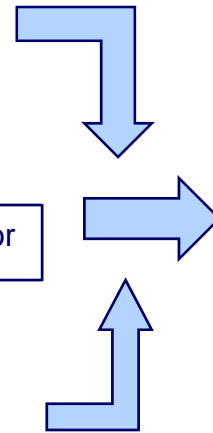


Schema



...

Query Templates



...

Queries

Ranking Structured Queries

Rank structured queries base on the likelihood of matching user's intent

Top-k Queries

return the first k structured queries with non-empty results

Ranking function: $SER \times AC \times TC$

- SER: expected number of results
- AC: attribute completeness
- TC: term completeness

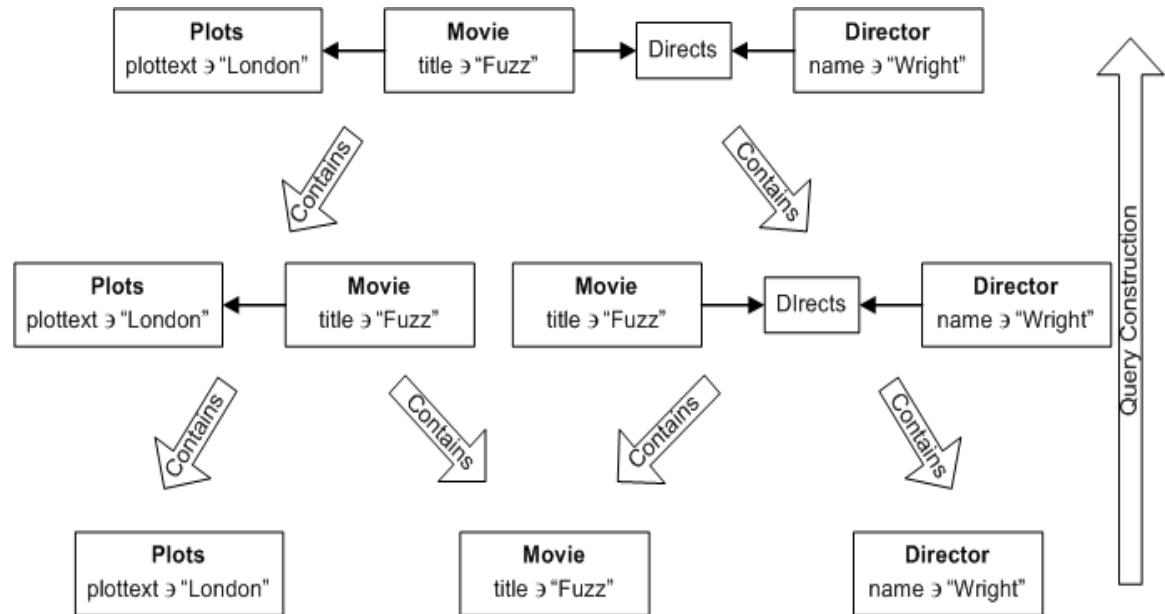
Query Construction Options = Partial Queries

The smallest partial queries are keyword attribute pairs:

e.g. Fuzz: movie.title OR movie.plots OR actor.name

Wright: movie.title OR director.name

User can construct his structured query by climbing the hierarchy of partial queries.



Ranking Partial Queries

When the database schema is big, there will be a big number of query construction options.

return the k options user like most

Ranking function: $SEL \times AC$

- SEL: selectivity
- AC: attribute completeness

Experiment

Datasets:

IMDB

Relation Schema	# Tuples
movies(mID, title, year)	858,967
directs(mID, dID)	572,638
directors(dID, name)	123,178
acts(aID, mID, character)	6,727,186
actors(aID, name)	1,199,918
plots(mID, plottext)	91,565
genres(mID, genre)	637,976
Total number of tuples	10,211,428

Lyrics

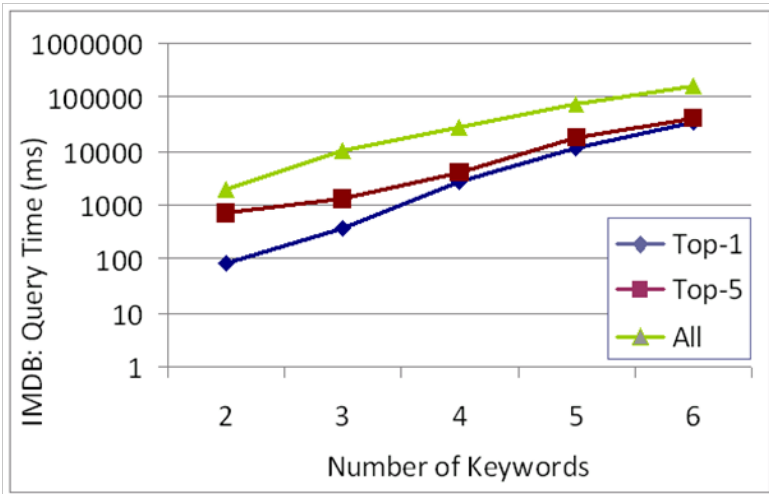
Relation schema	# Tuples
artist(aID, name)	3,691
articalbum(aID, bID)	15,160
album(bID, title)	15,160
albumsong(bID, sID)	177,231
song(sID, title, lyrics)	177,231
Total number of tuples	388,473

Query sets: the query log of a real search engine

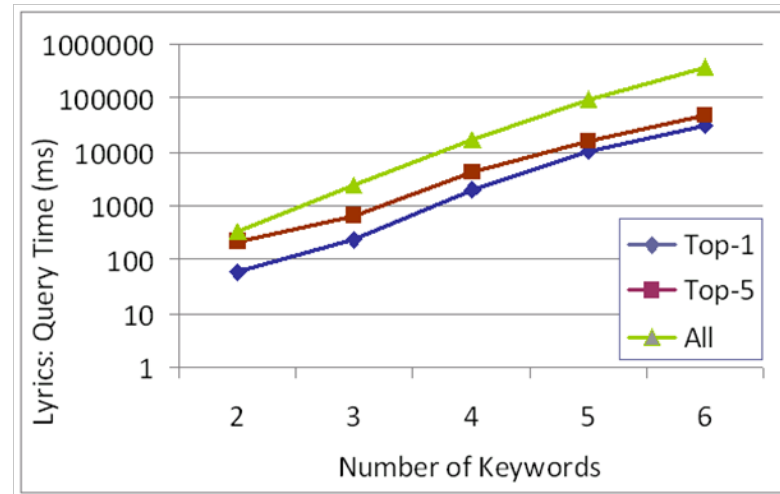
SUITS implemented in Java

Relational Database: MySQL

Performance



IMDB



Lyrics

Observation: I/O dominant, good performance for short and median keyword queries

SUITS - Constructing Structured Queries Using Keywords: Xuan Zhou, Elena Demidova, Gideon Zenz, Wolfgang Nejdl. Technical Report, submitted for publication.

3. Exploring Tags for Music Recommendation

Tags are:

- Written chaotically
- Not verified
- Unstructured
- Heterogeneous
- Unreliable

But if many,
the **correct ones** arise
“Wisdom of the masses”

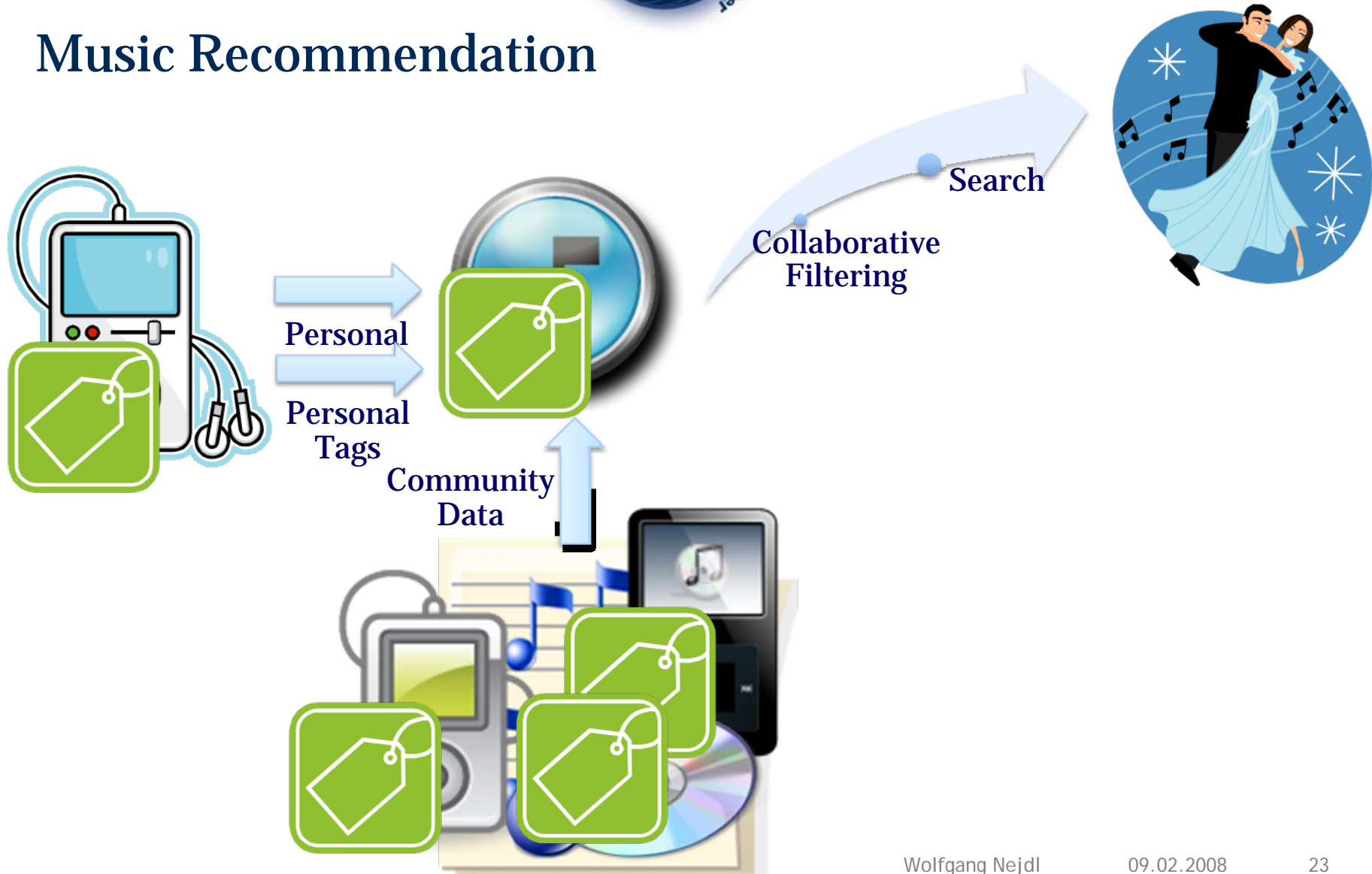


Last.fm – “The Social Music Revolution”

The screenshot shows the Last.fm interface for the track "My Happy Ending" by Avril Lavigne. The page is annotated with several blue callout boxes and arrows pointing to specific features:

- Track**: Points to the track title and album cover.
- Track Usage Info**: Points to the play count and scrobble information.
- Artist**: Points to the artist's name and profile link.
- Similar Artists**: Points to the list of artists recommended based on the track.
- Albums**: Points to the album "Under My Skin" featuring this track.
- Similar Tracks**: Points to the "Similar" tab in the navigation menu.
- Tags (with weight)**: Points to the list of user tags such as "alternative", "canadian", "female vocalist", "female vocalists", "pop", "pop rock", and "punk rock".
- User Comments**: Points to the "Shoutbox" section where users can leave comments.

Music Recommendation



Last.fm Tag Data

21,177 unique tags

- Number of times used
- Number of users who have used each tag
- Tag similarity

In total, tags have been used 18,735,549

60% of the top 100 tags describe a genre

40% : Personal Impressions, Artists, Time Period, Country of Provenance, Soundtrack, Tempo or Instruments

Music Recommendation – Algorithms

Three types of recommendation algorithms were investigated

- Collaborative Filtering based on Tracks

Baseline algorithm

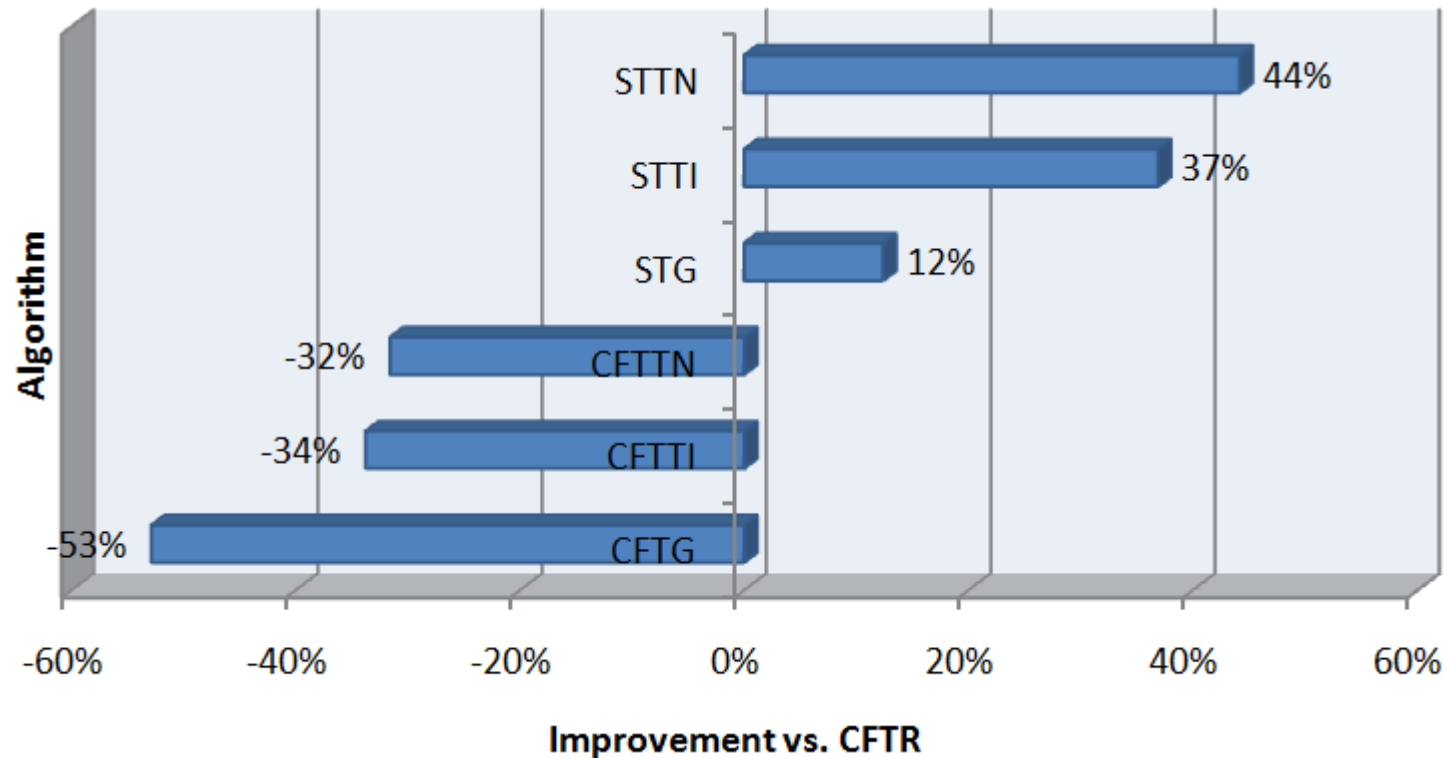
- Collaborative Filtering based on Tags

Recommended tags are used to search for tracks

- Search based on Tags

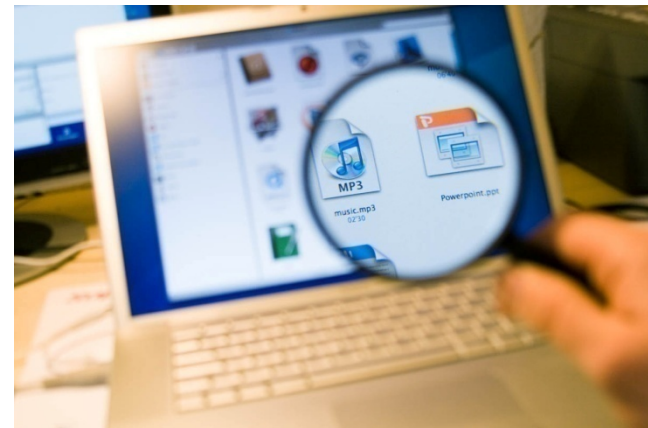
Tags in user profile are used directly to search for tracks

Gain over the Baseline (CF on Tracks)



The Benefit of Using Tag-Based Profiles. Claudiu S. Firan, Wolfgang Nejdl and Raluca Paiu. LA-WEB 2007

„Knowing better how to ask for it ...“



Avril Lavigne - My Happy Ending

44,613 plays scrubbed on Last.fm

Embed this track

More Tracks by Avril Lavigne

Track	Time	Listeners
Get Back	3:29	107,264
Complicated	3:58	88,632
Fall to Pieces	3:36	81,483

Avril Lavigne's Radio (show all 10)

Similar Artists

- Avril Lavigne
- Avril Lavigne
- Avril Lavigne
- Avril Lavigne
- Avril Lavigne

Albums featuring this track

- Under My Skin** (Released: 24 May 2004, 29 tracks, Label: RCA Records/LaFace)

User Tags (show more)

alternative canadian female vocalist female vocalists pop pop rock punk rock

Top Listeners (show more)

10,013 listeners total

Related Journals (show more)

My happy ending... by 11 July 2007

My Happy Ending... by 10 November 2007

Avril Lavigne (checked) by 10 July 2007

Avril Lavigne Showcase at Madrid, Review + impressions by 10 July 2007

Shoutbox

For Avril Lavigne - My Happy Ending

1 Leave a comment

LAST.FM LAST.FM © 2007 LAST.FM