Department of Computational Perception JKU JOHANNES KEPLER UNIVERSITY LINZ

Part III: Music Context-based Similarity

Markus Schedl
Peter Knees

{markus.schedl, peter.knees}@jku.at

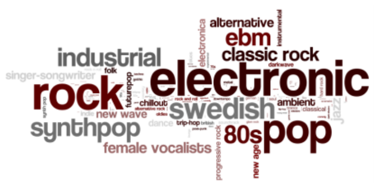
Department of Computational Perception
Johannes Kepler University (JKU)
Linz, Austria

Music Content vs. Music Context

Advantages of Content Analysis

- Features can be extracted from any audio file
- No other data or community necessary
- No cultural biases (i.e., no popularity bias, no subjective ratings etc.)

Advantages of Context Analysis



- Captures aspects beyond pure audio signal
- No audio file necessary
- Usually, user-based features are closer to what users want

Music Content vs. Music Context

Challenges for Context-Based Feature Extractors

- Dependence on availability of sources (Web pages, tags, playlists, ...)
- Popularity of artists may distort results
- Cold start problem of community-based systems (newly added entities do not have any information associated, e.g. user tags, users' playing behavior)
- Hacking and vandalism (cf. last.fm tag "brutal death metal")
- Bias towards specific user/listener groups (e.g., young, Internet-prone, metal listeners in last.fm)
- (Reliable) data often only available on artist level

Challenge for both Content and Context Analysis

- Extraction of relevant features from *noisy signal*

Context- and Web-Based Methods

In the following, contextual data refers to extended meta-data, usually

- Generated by users
- Unstructured data-sources
- Accessible via the Web

Two main classes of approaches covered in the following

- Text processing
- Co-occurrence analysis

As for content-based methods, similarity is the central concept for retrieval

Text-Based Approaches

Data sources:

- Web pages retrieved via Web search engines
- microblogs on *Twitter*
- product reviews
- semantic tags
- lyrics

Department of Computational Perception

Text-Based Similarity and Genre Classification

Use Web data to transform the music similarity task into a text similarity task

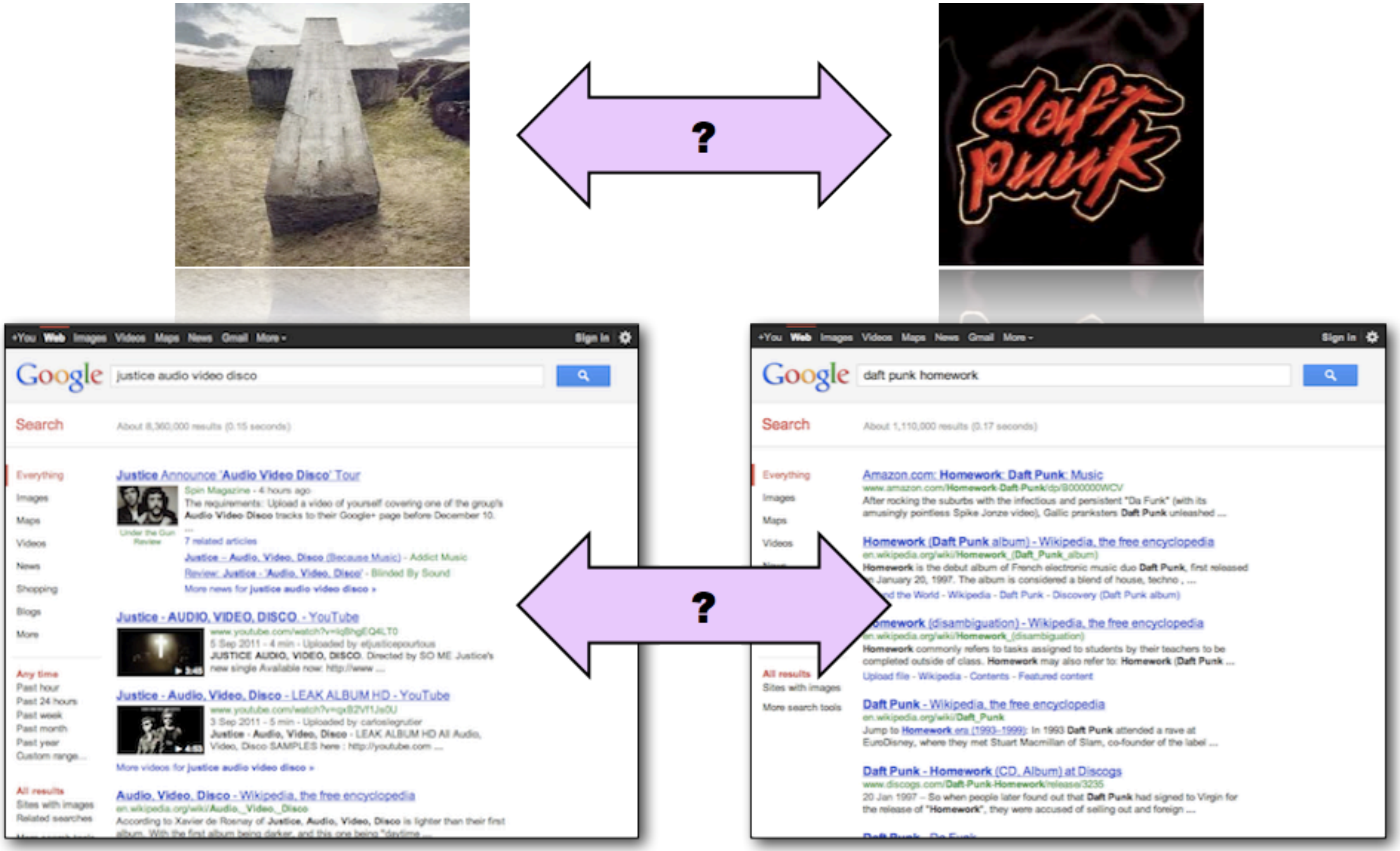
Allows to use the full armory of IR methods, typically…

- Bag-of-words, Vector Space Model
- Stopword removal, dictionaries, term selection
- TF · IDF
- Latent Semantic Indexing
- Part-of-Speech tagging
- Named Entity Detection
- Sentiment analysis

Large range of possible similarity measures

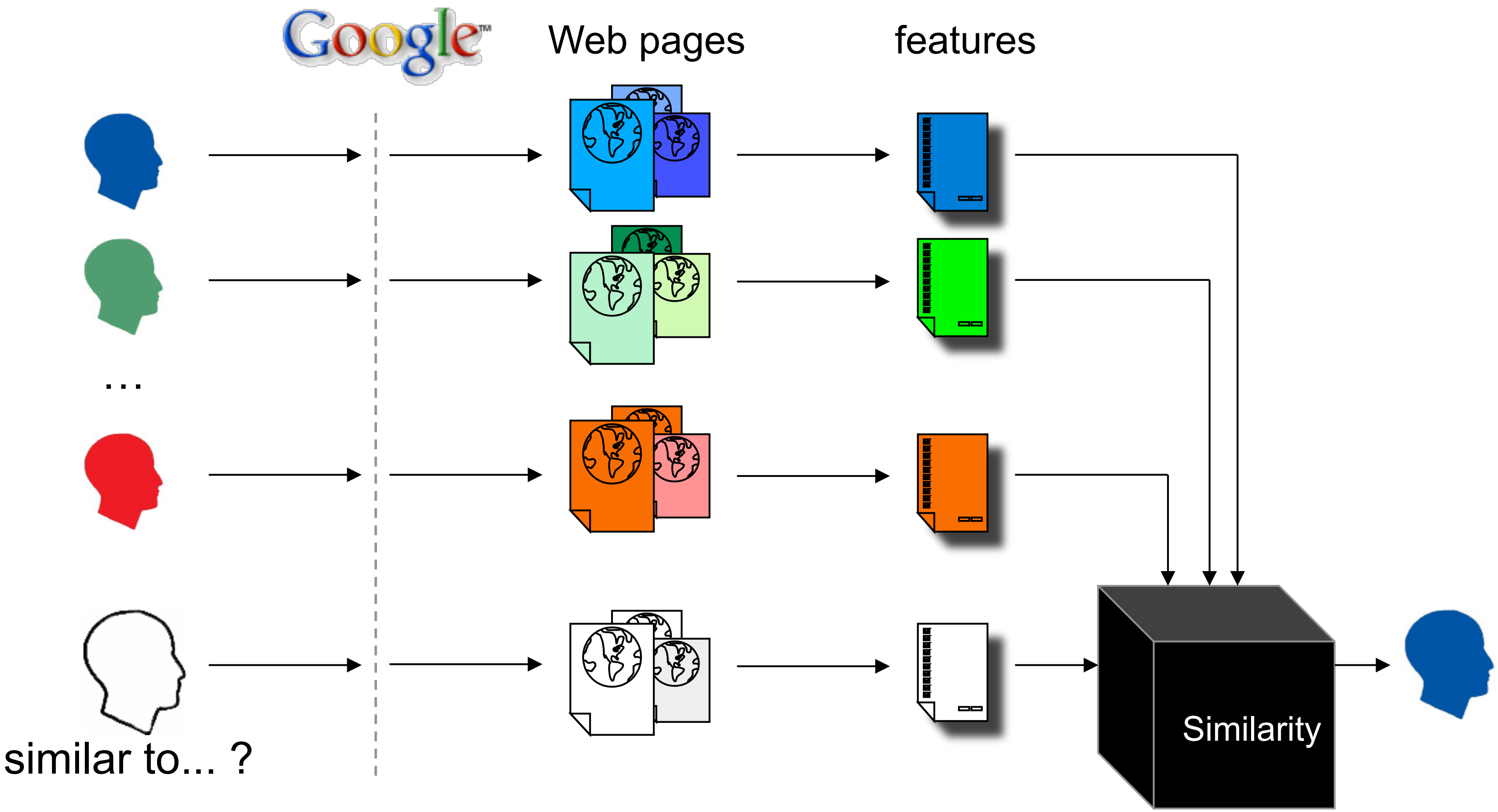
- Overlap, Manhattan, Euclidean, Cosine, etc.

Related Web Pages as Text Source



Department of
Computational
Perception

Related Web Pages as Text Source



Department of
Computational
Perception

Related Web Pages as Text Source

- Using search engines and queries such as

  ```
  "artist" +music
  "artist" +music +review
  ```

(Whitman, Lawrence; 2002) (Baumann, Hummel; 2003) (Knees et al.; 2004)

- Analyze
  - result page directly or
  - download up to top 100 Web pages (combine into one "virtual document" or analyze separately)
- Apply "IR magic"
- Applicable for similarity estimation, classification, retrieval, annotation

(NB: Most discriminating terms between genres are artist names and album/track titles)

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments):

(Schedl et al.; 2011)

- term frequency

Abbr.	Description	Formulation
TF_A	Formulation used for binary match SB = b	$r_{d,t} = \begin{cases} 1 & \text{if } t \in \mathcal{T}_d \\ 0 & \text{otherwise} \end{cases}$
TF_B	Standard formulation SB = t	$r_{d,t} = f_{d,t}$
TF_C	Logarithmic formulation	$r_{d,t} = 1 + \log_e f_{d,t}$
TF_C2	Alternative logarithmic formulation suited for $f_{d,t} < 1$	$r_{d,t} = \log_e(1 + f_{d,t})$
TF_C3	Alternative logarithmic formulation as used in *ltc* variant	$r_{d,t} = 1 + \log_2 f_{d,t}$
TF_D	Normalized formulation	$r_{d,t} = \frac{f_{d,t}}{f_d^m}$
TF_E	Alternative normalized formulation. Similar to [55] we use $K = 0.5$. SB = n	$r_{d,t} = K + (1 - K) \cdot \frac{f_{d,t}}{f_d^m}$
TF_F	Okapi formulation, according to [55, 36]. For W we use the vector space formulation, i.e., the Euclidean length.	$r_{d,t} = \frac{f_{d,t}}{f_{d,t} + W_d / av_{d \in D}(W_d)}$
TF_G	Okapi BM25 formulation, according to [35].	$r_{d,t} = \frac{(k_1+1) \cdot f_{d,t}}{f_{d,t} + k_1 \cdot \left[(1-b) + b \cdot \frac{W_d}{av_{d \in D}(W_d)}\right]}$ $k_1 = 1.2, b = 0.75$

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- term frequency

- inverse document frequency

Abbr.	Description	Formulation
IDF_A	Formulation used for binary match SB = x	$w_t = 1$
IDF_B	Logarithmic formulation SB = f	$w_t = \log_e \left(1 + \frac{N}{f_t}\right)$
IDF_B2	Logarithmic formulation used in *ltc* variant	$w_t = \log_e \left(\frac{N}{f_t}\right)$
IDF_C	Hyperbolic formulation	$w_t = \frac{1}{f_t}$
IDF_D	Normalized formulation	$w_t = \log_e \left(1 + \frac{f_m}{f_t}\right)$
IDF_E	Another normalized formulation SB = p	$w_t = \log_e \frac{N - f_t}{f_t}$
	The following definitions are based on the term's noise n_t and signal s_t.	$n_t = \sum_{d \in \mathcal{D}_t} \left(-\frac{f_{d,t}}{F_t} \log_2 \frac{f_{d,t}}{F_t}\right)$ $s_t = \log_2(F_t - n_t)$
IDF_F	Signal	$w_t = s_t$
IDF_G	Signal-to-Noise ratio	$w_t = \frac{s_t}{n_t}$
IDF_H		$w_t = \left(\max_{t' \in \mathcal{T}} n_{t'}\right) - n_t$
IDF_I	Entropy measure	$w_t = 1 - \frac{n_t}{\log_2 N}$
IDF_J	Okapi BM25 IDF formulation, according to [35, 31]	$w_t = \log \frac{N - f_t + 0.5}{f_t + 0.5}$

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- term frequency

- inverse document frequency

- virtual document modeling: *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- term frequency

- inverse document frequency

- virtual document modeling: *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.

- normalization with respect to document length

Abbr.	Description	Formulation
NORM_NO	No normalization.	
NORM_SUM	Normalize sum of each virtual document's term feature vector to 1.	$\sum_{t \in T_d} r_{d,t} = 1$
NORM_MAX	Normalize maximum of each virtual document's term feature vector to 1.	$\max_{t \in T_d} r_{d,t} = 1$

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- **term frequency**
- **inverse document frequency**
- **virtual document modeling:** *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.
- **normalization with respect to document length**
- **similarity measure**

Abbr.	Description	Formulation
SIM_INN	Inner Product	$S_{d_1,d_2} = \sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)$
SIM_COS	Cosine Measure	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1} \cdot W_{d_2}}$
SIM_DIC	Dice Formulation	$S_{d_1,d_2} = \frac{2 \sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1}^2 + W_{d_2}^2}$
SIM_JAC	Jaccard Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)}{W_{d_1}^2 + W_{d_2}^2 - \sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)}$
SIM_OVL	Overlap Formulation	$S_{d_1,d_2} = \frac{\sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} \cdot w_{d_2,t}\right)}{\min(W_{d_1}^2, W_{d_2}^2)}$
SIM_EUC	Euclidean Similarity	$D_{d_1,d_2} = \sqrt{\sum_{t \in \mathcal{T}_{d_1,d_2}} \left(w_{d_1,t} - w_{d_2,t}\right)^2}$ $S_{d_1,d_2} = \left(\max_{d'_1,d'_2}\left(D_{d'_1,d'_2}\right)\right) - D_{d_1,d_2}$
SIM_JEF	Jeffrey Divergence-based Similarity	$S_{d_1,d_2} = \left(\max_{d'_1,d'_2}\left(D_{d'_1,d'_2}\right)\right) - D_{d_1,d_2}$ $D(F,G) = \sum_i \left(f_i \log \frac{f_i}{m_i} + g_i \log \frac{g_i}{m_i}\right)$ $m_i = \frac{f_i+g_i}{}$

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- term frequency
- inverse document frequency
- virtual document modeling: *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.
- normalization with respect to document length
- similarity measure
- index term set

Abbr. / Term Set	Cardinality	Description
TS_A - all_terms	C224a, QS_A: 38,133 C224a, QS_M: 19,133 C3ka, QS_A: 1,489,459 C3ka, QS_M: 437,014	All terms (stemmed) that occur in the corpus of the retrieved Twitter posts.
TS_S - scowl_dict	698,812	All terms that occur in the entire SCOWL dictionary.
TS_N - artist_names	224 / 3,000	Names of the artists for which data was retrieved.
TS_D - dictionary	1,398	Manually created dictionary of musically relevant terms.
TS_L - last.fm_toptags	250	Overall top-ranked tags returned by last.fm's *Tags.getTopTags* function.
TS_F - freebase	3,628	Music-related terms extracted from Freebase (genres, instruments, emotions).

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- **term frequency**
- **inverse document frequency**
- **virtual document modeling:** *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.
- **normalization with respect to document length**
- **similarity measure**
- **index term set**
- **query scheme**

Abbr.	Query Scheme
QS_A	"artist name"
QS_M	"artist name"+music

Large-Scale Study

Investigating different aspects in modeling artist term profiles from Web pages (9,200 experiments): (Schedl et al.; 2011)

- term frequency
- inverse document frequency
- virtual document modeling: *concatenate* all Web pages/posts of the artist or perform *aggregation* via mean, max, etc.
- normalization with respect to document length
- similarity measure
- index term set
- query scheme

implemented in our ***CoMIRVA*** framework available from http://www.cp.jku.at/comirva

Interesting Findings

(Schedl et al.; 2011)

- modeling artists as ***virtual documents*** is preferable
- using query scheme ***"artist" +music*** outperforms ***"artist"***
- ***normalization*** does not yield a statistically significant difference
- standard ***cosine similarity*** measure does not yield the very best results, but the most stable ones (varying other parameters)
- ***consistent results*** among the (top-ranked) variants for two collections
- ***minor change in one parameter*** can have a huge impact on performance
- overall winners in terms of term weighting functions:

  TF_C3.IDF_I

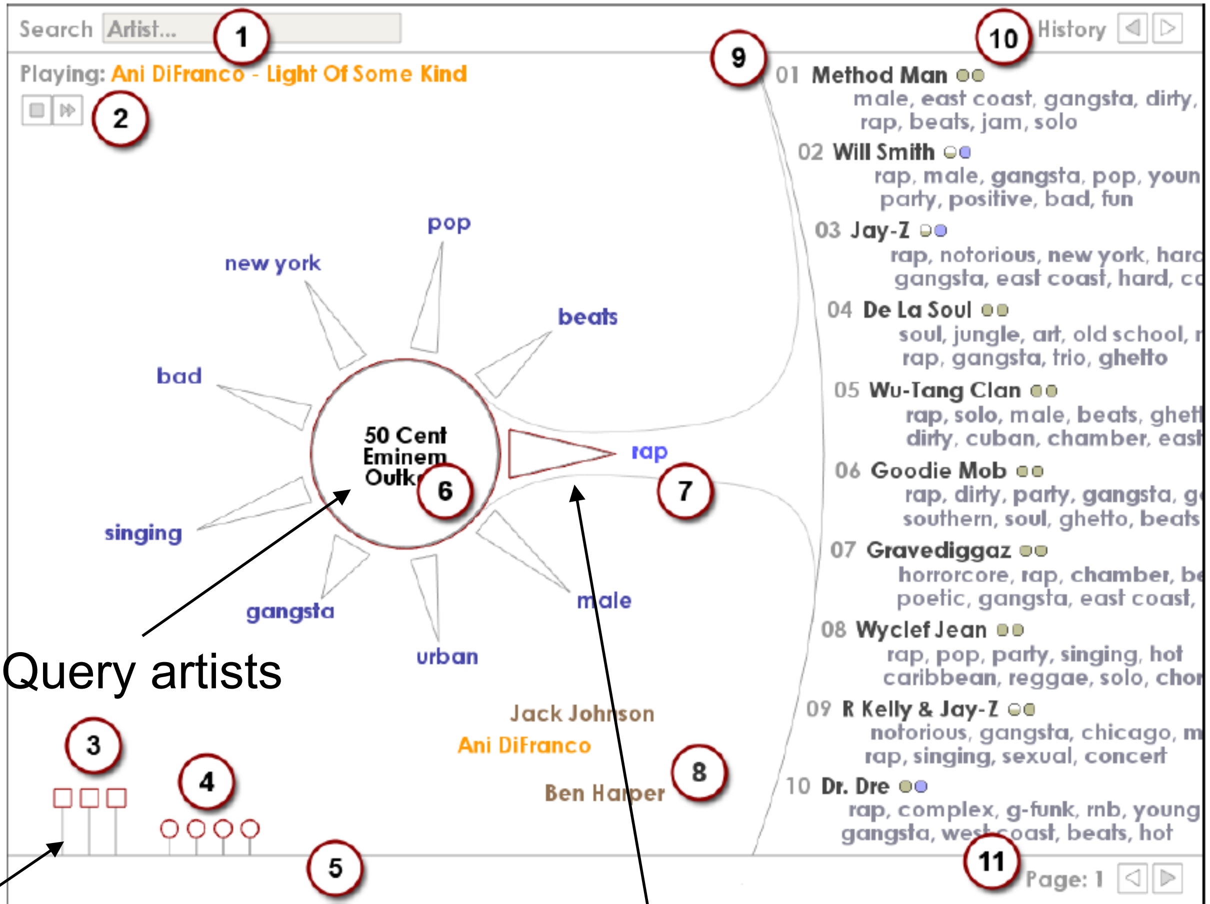
  TF_C3.IDF_H → logarithmic formulations for TF and IDF

  TF_C2.IDF_I

Web-Based Descriptions for Browsing

"MusicSun"
(Pampalk, Goto; 2007)

- Interactive "Artist Recommender"
- Recommendation is influenced/directed by selecting relevant similarity dimensions
- Combines different similarity measures



3 types of similarity: audio, web-based, word
overall similarity = weighted average of ranks

Relevant word dimension

Department of Computational Perception

Web-Based Texts for Indexing and Retrieval

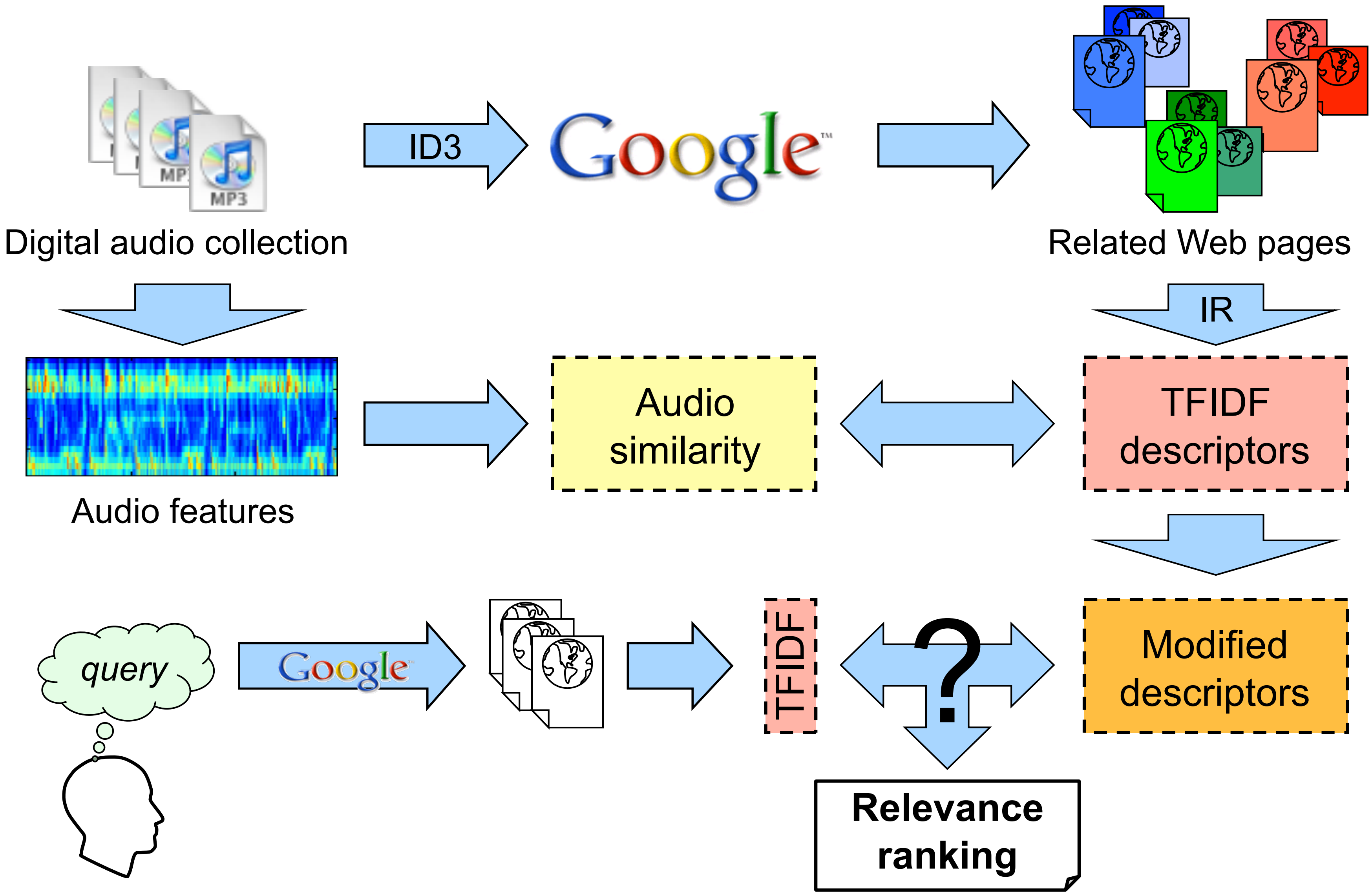
- Use Web data to transform music retrieval into a text retrieval task
- Find associated (or associable) texts and use them instead of music
- Allows for diverse and semantic queries
  (e.g, “chilled music”, “great riffs”)

Search Sounds (Celma et al.; 2006)
Crawl lists of RSS feeds and use Weblog entries to index pieces

Squiggle (Celino et al.; 2006):
Combine meta-data databases (like MusicBrainz) for rich indexing

Gedoodle (Knees et al.; SIGIR 2007):
Query Google and combine Web pages to index pieces

Gedoodle



Department of Computational Perception

Gedoodle

(Knees et al.; SIGIR 2007)

- For each track: join 100 Google results of
  - "artist" music
  - "artist" "album" music review
  - "artist" "title" music review -lyrics
- Combine all pages into one virtual document
- Create normalized TFIDF vector for each track
- Include audio similarity for vector modification and dimensionality reduction

Gedoodle (Example queries)

damon albarn | Search for Music

Results **18** - **27** of **1691** for **damon albarn**. (**0.02** seconds)

Music Is My Radar
by blur
from the album: *blur: the best of*
Genre: Alternative - 192 kBit/s - length: 5:29 min. Listen

On Your Own
by blur
from the album: *blur: the best of*
Genre: Alternative - 192 kBit/s - length: 4:27 min. Listen

Girls & Boys
by blur
from the album: *blur: the best of*
Genre: Alternative - 192 kBit/s - length: 4:19 min. Listen

There's No Other Way
by blur
from the album: *blur: the best of*
Genre: Alternative - 192 kBit/s - length: 3:14 min. Listen

Feel Good Inc.
by Gorillaz
from the album: *FM4 Soundselection 12*
Genre: Alternative - 192 kBit/s - length: 4:20 min. Listen

Say So What
by Graham Coxon
from the album: *Uncut - 2006.05*
Genre: Rock - 192 kBit/s - length: 3:05 min. Listen

Slash Dot Dash
by Fatboy Slim
from the album: *The Greatest Hits: Why Try Harder*
Genre: Electronic - 192 kBit/s - length: 2:55 min. Listen

smooth and relaxing | Search for Music

Results **1** - **10** of **1774** for **smooth and relaxing**. (**0.02** seconds)

Joy And Pain
by Count Basic
from the album: *Moving In The Right Direction*
Genre: Acid Jazz - 168 kBit/s - length: 6:25 min. Listen

Higher
by Count Basic
from the album: *Bigger & Brighter*
Genre: Acid Jazz - 192 kBit/s - length: 4:00 min. Listen

Sweet Luis
by Count Basic
from the album: *Moving In The Right Direction*
Genre: Acid Jazz - 158 kBit/s - length: 5:11 min. Listen

Got To Do
by Count Basic
from the album: *Moving In The Right Direction*
Genre: Acid Jazz - 167 kBit/s - length: 4:58 min. Listen

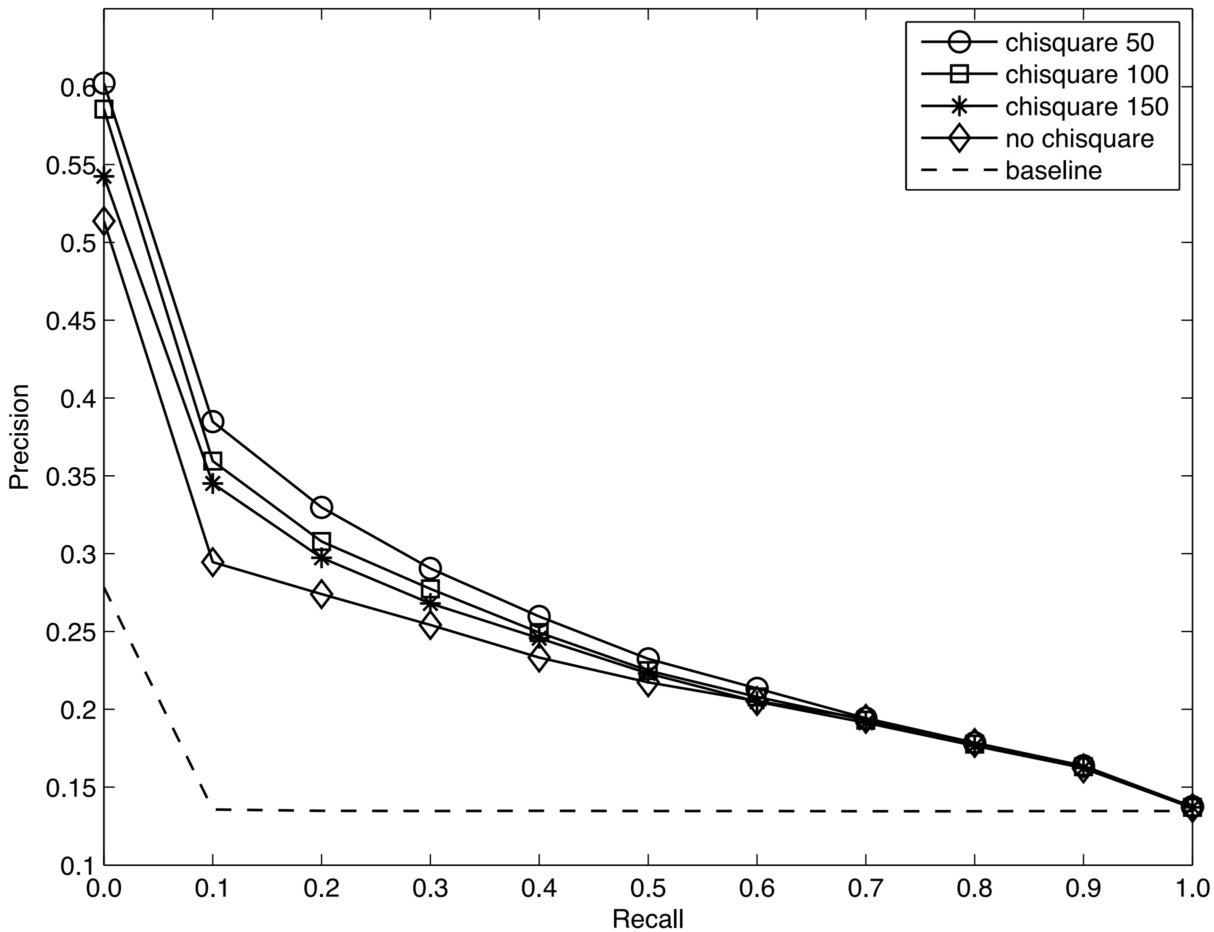
John Lee Huber
by Tosca
from the album: *J.A.C.*
Genre: Electronica/Dance - 192 kBit/s - length: 4:33 min. Listen

No More Olives
by Tosca
from the album: *J.A.C.*
Genre: Electronica/Dance - 192 kBit/s - length: 6:02 min. Listen

Naschkatze
by Tosca
from the album: *J.A.C.*
Genre: Electronica/Dance - 192 kBit/s - length: 4:34 min. Listen

Gedoodle Results

Effects of TFIDF feature space pruning using content-similarity-based χ^2-test (Knees et al.; SIGIR 2007)



Department of Computational Perception

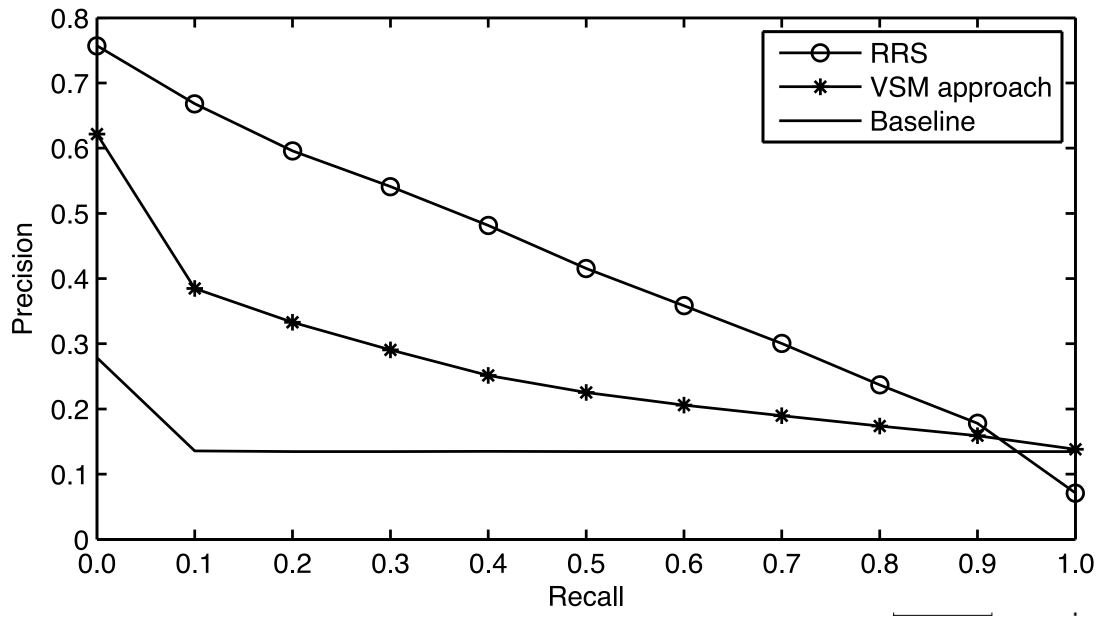
Gedoodle Results

Alternative: Document-centered ranking (Knees et al.; ECIR 2008)

- Indexing of all web documents in standard index
- Music query addresses this index
- Music ranking calculated from web doc ranking according to

$$RRS(m,q) = \sum_{p \in D_m \cap D_q} 1 + |D_q| - rank(p, D_q)$$

Comparison with vector space model



Semantic Querying via Auto-Tagging

- Use machine learning techniques to predict tags (labels) based on song features (content, context, or combination)
- Automatic description of music (browsing) and automatic generation of indexing terms for retrieval
- Mitigates "cold-start problem" in social tagging

Automatic Record Reviews (Whitman, Ellis; 2004)

Regularized least squares learning on TFIDF-Web and cepstral features

Autotagger (Bertin-Mahieux et al.; 2008)

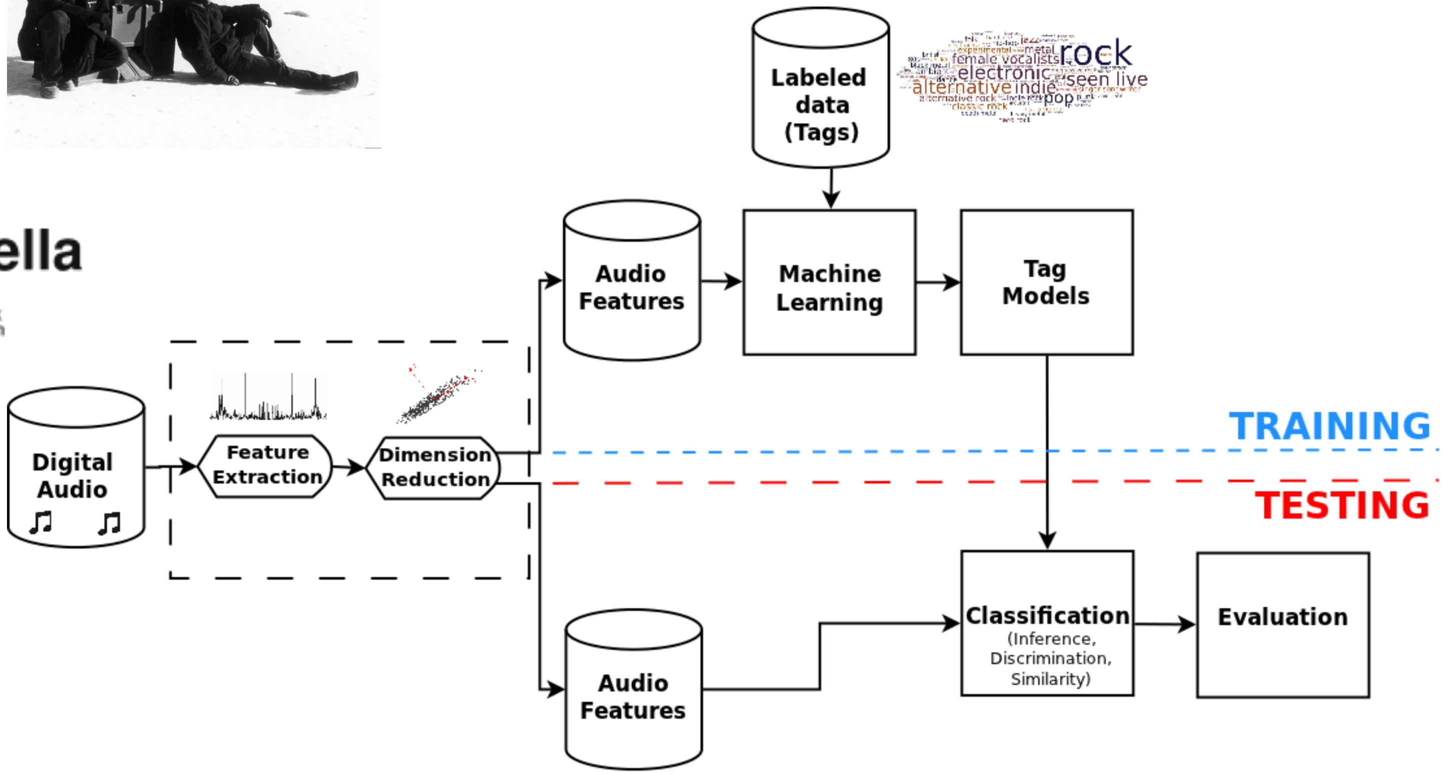
Ensemble classifier to map MFCCs, autocorrelation, Const-Q. to Web tags

Semantic Music Discovery (Turnbull et al.; SIGIR 2007, 2009):
Combines timbre, harmony, Web texts, and Web tags to predict user labels

Semantic Annotation of Music Collections (Sordo; 2012)
Propagation of tags through audio similarity

Auto-Tagging/Retrieval by Tag

Learning indexing labels from content features (Sordo; 2012)



Music Information Extraction from Web Pages

Web data is a rich source for all types of meta-data and semantic relations

Methods from NLP, IE, Named Entity Detection for data extraction

- **Genres, Moods, Similarities** using Rule Patterns
  (Geleijnse, Korst; 2006)
- **Band Members and Line-Up** using Rule Patterns
  (Schedl, Widmer; 2007)
- **Band Members, Discography, Artist Detection (**rule based)
  (Krenmair; 2010)
- **Band Members, Discography** using Supervised Learning
  (Knees, Schedl; 2011)
- **Album cover** detection and extraction
  (Schedl et al., ECIR 2006)

Department of
Computational
Perception

Microblogs as Text Sources


Radio Voorne @up_radiovoorne 7m
Nu op @RadioVoorne: **Justice** vs. Simian - We Are Your Friends #radiovoorne #nowplaying

Jai Jones @jonesjai 9m
Love this track. #NowPlaying **Justice** – Audio, Video, Disco. on #Spotify open.spotify.com/track/6iqBiGFv...

Sophie Stratford @sophiestratford
#nowplaying Helix - **Justice**

radio2XS @radio2XS 1h
JUSTICE - On 'n' On is #nowplaying @radio2XS. Listen at tiny.cc/2xs

?


Bawse @SneakerMike 9m
#nowplaying **daft punk** - something about us (remix)

Marcos Martin @marcos6m 1h
Escuchando The Grid por **Daft Punk** #nowplaying #tunesday tinysong.com/Yav2

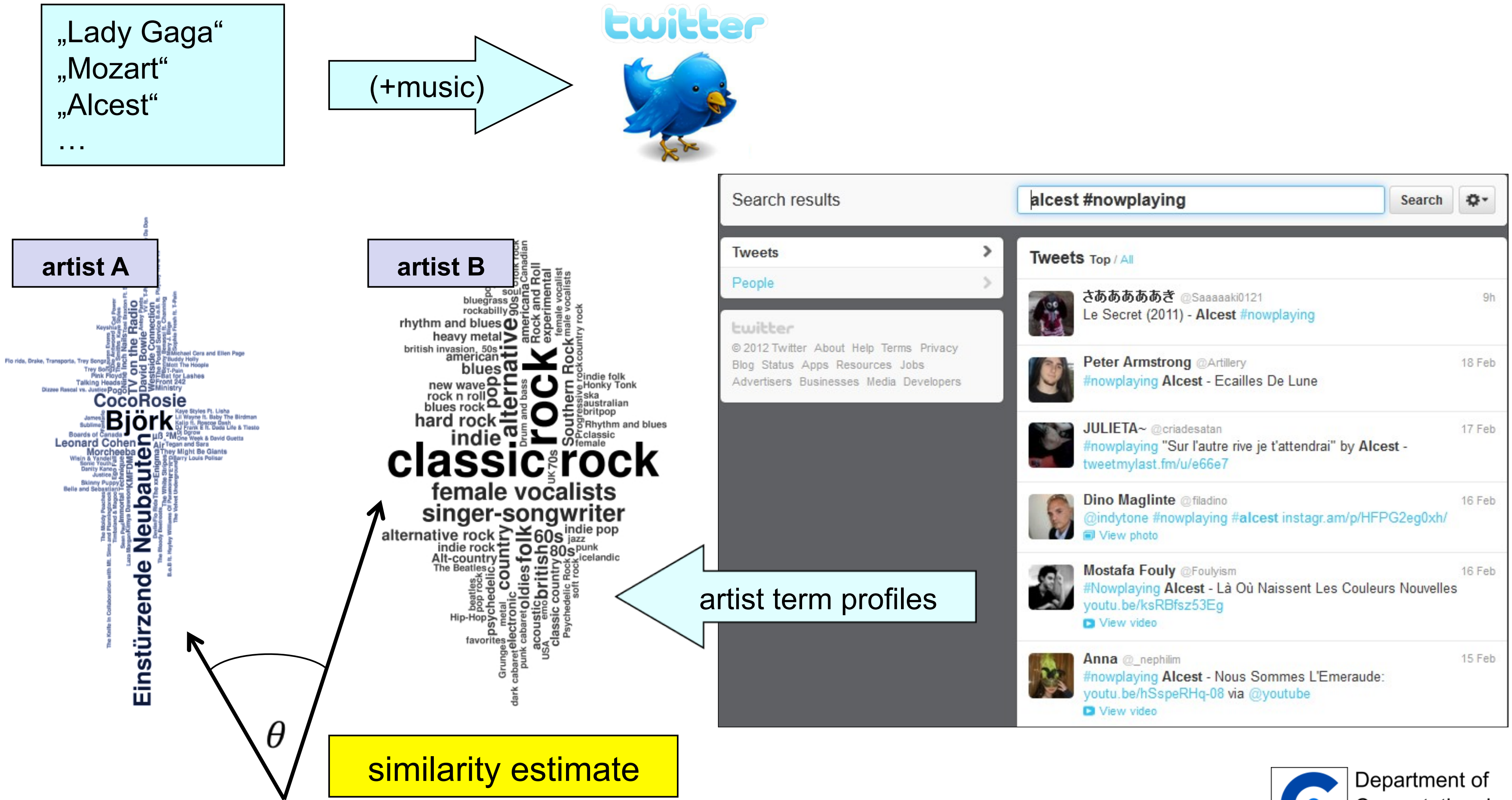
pencer @dozycats 1h
#nowplaying Tron Legacy (End Titles) - **Daft Punk**

david. @dvzLun 1h
#nowplaying LCD soundsystem - **daft punk** is playing at my house.

Department of
Computational
Perception

Microblogs as Text Sources: Scheme

(Schedl; 2012a)



Microblogs as Text Sources: Scheme

(Schedl; 2012a)

Large-scale study similar to (Schedl et al.; TOIS 2011)

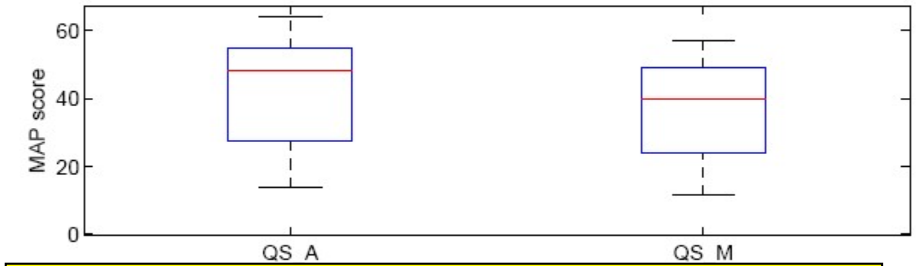
Investigating different aspects in modeling artist term profiles from microblogs (23,100 experiments):

- query scheme
- index term set
- term frequency
- inverse document frequency
- normalization with respect to document length
- similarity measure

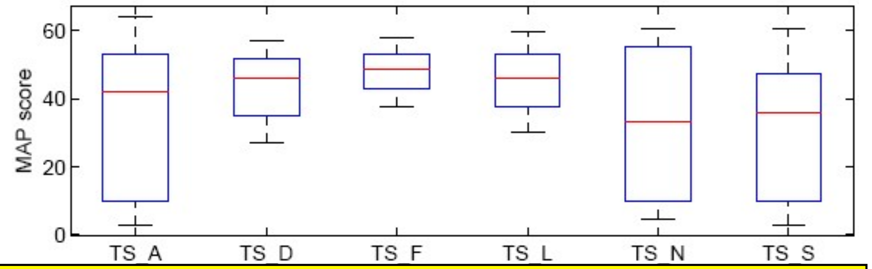
implemented in our ***CoMIRVA*** framework available from http://www.cp.jku.at/comirva

Microblogs as text-based source: Results

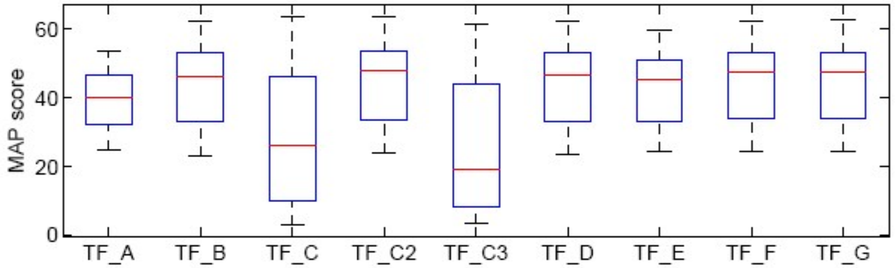
(Schedl; 2012a)



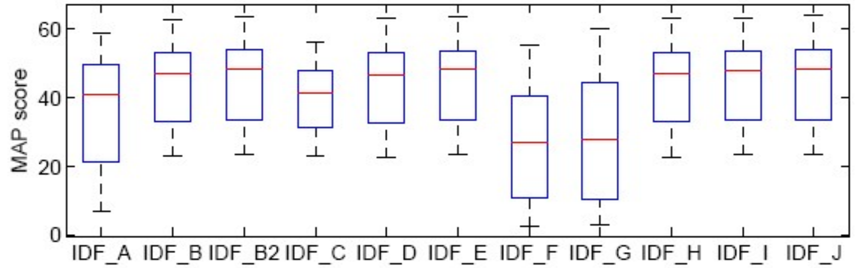
use query scheme "artist name"



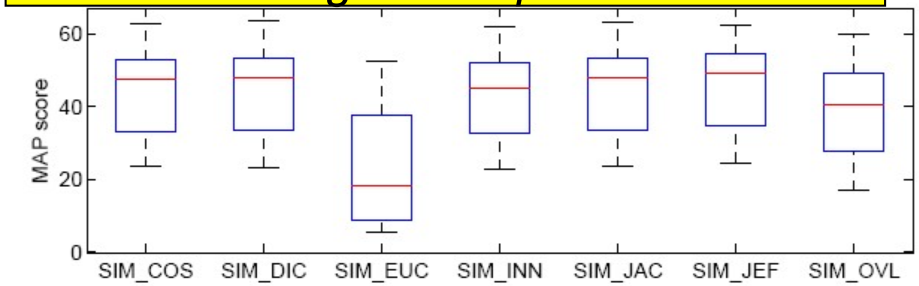
music-specific dictionary favorable



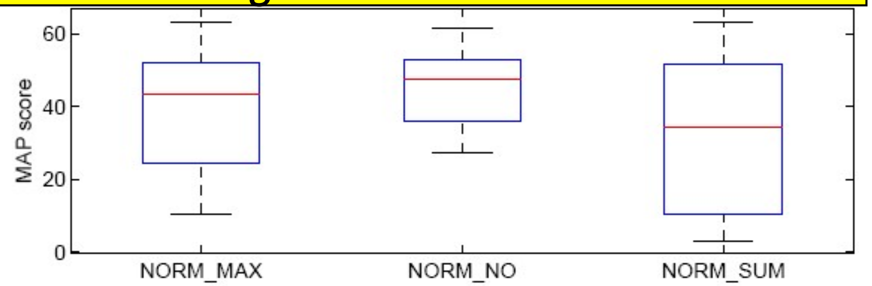
use *log* or *Okapi BM25*



use *logarithmic* formulations



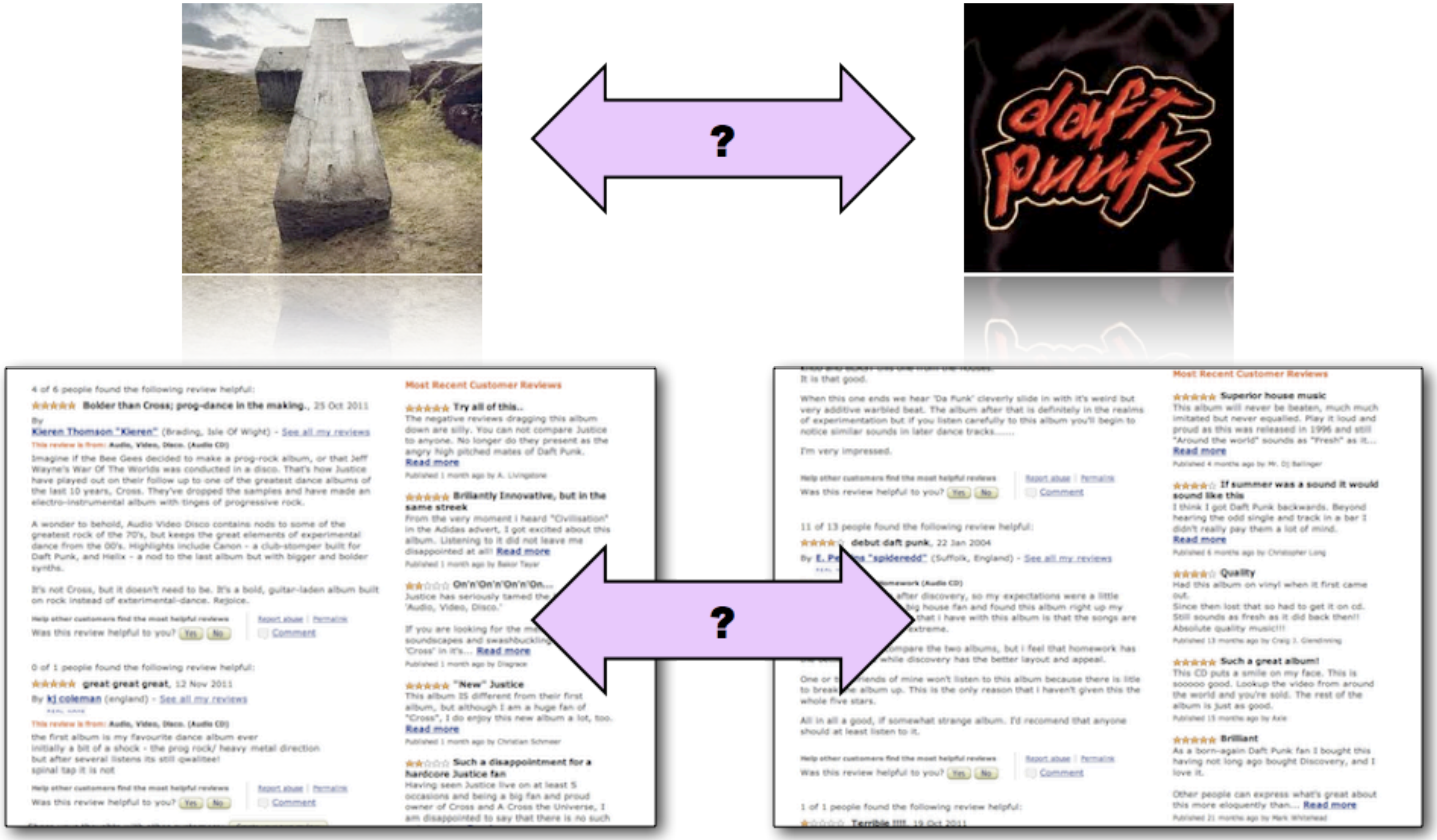
don't use Euclid; use Jeffrey or Inner Prod.



no document length normalization

Department of
Computational
Perception

Product Reviews as Text Sources



Department of
Computational
Perception

Product Reviews as Text Sources

Exploiting sources such as *Amazon.com* or *Epinions.com* (Hu et al.; 2005)

This review is from: Ray of Light (Audio CD)

This is Madonna's work of art. And this CD is the very best collection of any music she has ever produced since "Erotica." Madonna's lyrics are beautiful and strong because even after 9 years it still stands the test of time. It's completely impossible for this CD to be dated; with the electronica feel to it and fast moving dance numbers, such as the title-track this CD was way ahead of its time. Even in the double-00's "Ray of Light" is still very important as both a dance record and a record of reflection and interpersonal renewal.

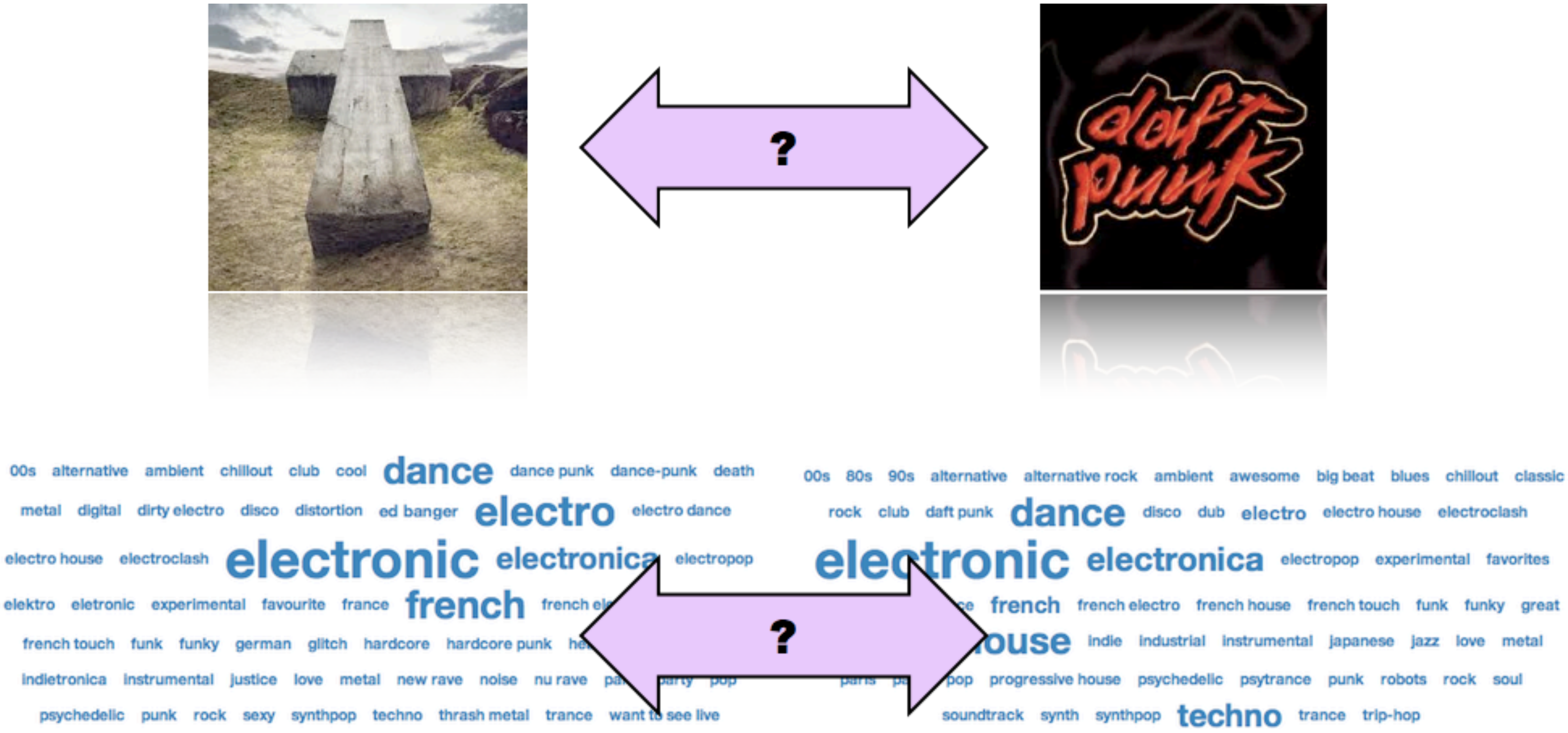
This review is from: **Never Gonna Give You Up (MP3 Download)**

This is truly Astley's greatest opus.
The track is flawless. It is instantly accessible, but features many hidden layers and pleasures that cannot be discovered upon the first listen alone. With this and all of his other fantastic work, it's no wonder that Radiohead calls Astley their "greatest inspiration."

Allows for sentiment analysis and associated rating prediction

Very prone to attacks (remedy: consider “helpfulness” ratings)

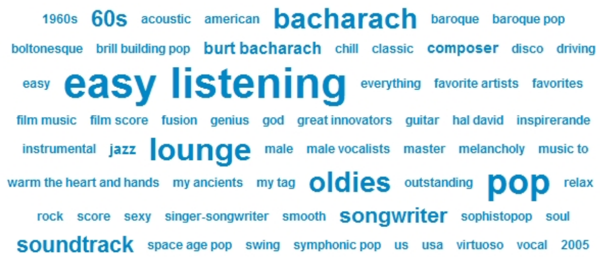
Community Tags as Text Sources



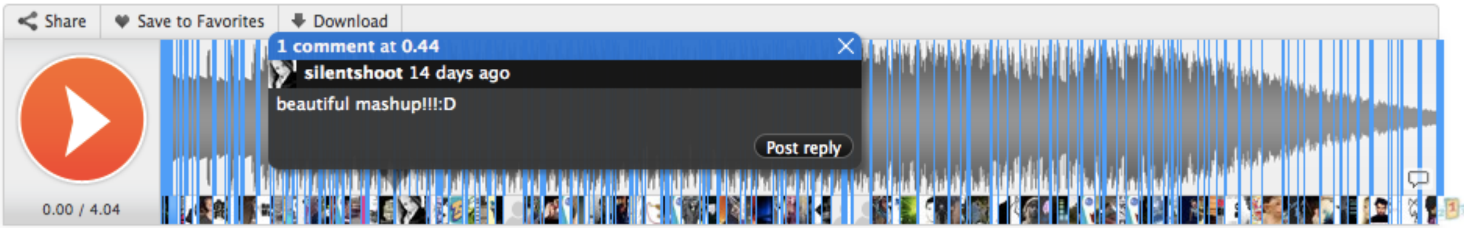
Department of Computational Perception

Tag Sources

- **Community**
  e.g., Last.fm



  e.g., Soundcloud (annotations along timeline)



- **Games with a purpose** (GWAP)
  e.g., Tag-a-Tune
  (Law, von Ahn; 2009)

- **Autotags** (see before)

Department of
Computational
Perception

Community Tags as Text Sources

Treating collections of tags (e.g., from Last.fm) as documents

(Pohle et al.; 2007) (Levy, Sandler; 2008) (Hu et al.; 2009)

- Retrieve tags for artist or track from Last.fm
- Cleaning of noisy and redundant tags:
  manually or automatically (Geleijnse et al.; 2007)
- List of collected terms is treated as text document and TF·IDF'd
  (Levy, Sandler; 2007)
- Optionally, LSA to reduce dimensionality
- Comparison of vectors via cosine similarity (or overlap score)

- Data often available in standardized fashion, dedicated terms for music
- Lower dimensionality
  e.g., 13,500 tags vs. >200,000 Web terms (Levy, Sandler; 2007)
- Depends on community, needs annotators
- Hacking and Attacks!

Department of Computational Perception

Lyrics as Text Source

?

Before day break there was none
And as it broke there was one
The Moon, the sun, it goes on 'n' on
The winter battle was won
The summer children were born
And so the story goes on 'n' on
Come woman if your life beats
Those we buried with the house keys
Smoke and feather where the fields are green
From here to eternity
Come woman on your own time

?

Around the world, around the world
Around the world, around the world
Around the world, around the world

Around the world, around the world
Around the world, around the world
Around the world, around the world

Around the world, around the world
Around the world, around the world
Around the world, around the world

Department of
Computational
Perception

Lyrics as Text Source

Topic Features (Logan et al.; 2004)

- Typical topics for lyrics are distilled from a large corpus using (P)LSA ("Hate", "Love", "Blue", "Gangsta", "Spanish")
- Lyrics are transformed to topic-based vectors, similarity is calculated via L_1 distance

- Alternative approaches use TF·IDF with optional LSA and Stemming for **Mood Categorization** (Laurier et al.;2009) (Hu et al.; 2009)

Rhyme Features (Mayer et al.; 2008) (Hirjee, Brown; 2009)

- Phonetic transcription is searched for patterns of rhyming lines (AA, ABAB, AABB)
- Frequency of patterns + statistics like *words per minute, punctuation freq.* etc.

Other Features (Mahedero et al.; 2005) (Hirjee, Brown; 2009)

- Language, structure

Text-based Similarity Approaches: Summary

	Web-Terms	Microblogs	Reviews	Tags	Lyrics
Source	Web pages	platform	shops, platform	Web service	portal
Community-based	depends	depends	yes	yes	no
Level	artists	artists (tracks)	albums	artists (tracks)	tracks (artists)
Feature Dimensionality	very high	high	possibly high	moderate	possibly high
Specific Bias	low	low	personal	community	none
Potential Noise	high	high	low	moderate	low

Similarity from Co-Occurrences

Idea: expect entities that occur frequently in the same context to be similar

Data sources considered:

- Page count estimates from Web search engines
- Shared folders/search queries on the *Gnutella* file sharing network
- Collaborative filtering on playcounts from *Last.fm*
- Occurrences in playlists

Department of Computational Perception

Search Engine Page Count Estimates

(Schedl et al.; 2005)

For all pairs of artists: query **`“artist 1” “artist 2” +music +review`**
For each artist: query **`“artist” +music +review`**

Google

Use **page counts** for sim. (results in quadratic page count matrix)

$$sim_{pc\_cp}(A_i, A_j) = \frac{1}{2} \cdot \left(\frac{pc(A_i, A_j)}{pc(A_i)} + \frac{pc(A_i, A_j)}{pc(A_j)} \right)$$

To avoid quadratic number of queries: download top 100 pages for each artist and parse for occurrences of other artists (linear complexity)

NB: asymmetry of *pc* matrix can be used to identify prototypical artists!

Shared Folders in a P2P Network

Make use of meta-data transmitted as files names or ID3 tags in P2P network OpenNap (Whitman, Lawrence; 2002) (Ellis et al.; 2002)

Information gathered from users' shared folders (no file downloads!)

Similarities via artist co-occurrences in collections (cond. prob.)

Sparse co-occurrence matrix

Experiments on Gnutella network (Shavitt, Weinsberg; 2009):

- meta-data highly inconsistent
- can be used as song-based similarity measure and to estimate localized popularity/trends (matching IP addresses difficult!)

Last.fm Playcounts

Use *explicit* or *implicit* ratings of users or interpret number of plays of a song as a “rating”

Results in a user-track rating matrix

Use standard **collaborative filtering** approaches to predict similarities (or to recommend unknown music)
e.g., (Resnick et al.; 1994)

Item-based: compare tracks by calculating similarity on vectors over all users

User-based: find similar users by comparing listening pattern vectors; use to find relevant/similar tracks yet unknown to user

Playlist Co-Occurrences

Analysis of co-occurrences of artists and songs on radio station playlists and compilation CD databases (CDDB) (Pachet et al.;2001)

$$sim_{pl\_cooc}(A_i, A_j) = \frac{1}{2} \cdot \left[\frac{cooc(A_i, A_j)}{cooc(A_i, A_i)} + \frac{cooc(A_j, A_i)}{cooc(A_j, A_j)} \right]$$

Analysis of 29K playlists from “Art of the Mix” (Cano, Koppenberger;2004): artists similar if they co-occur in playlist (highly sparse)

Analysis of >1M playlists from “MusicStrands” (Baccigalupo et al.; 2008):

- distance in playlists taken into account $\beta_0 = 1, \beta_1 = 0.8, \beta_2 = 0.64$

$$dist_{pl\_d}(A_i, A_j) = \sum_{h=0}^{2} \beta_h \cdot [d_h(A_i, A_j) + d_h(A_j, A_i)]$$

- playlist prediction using case-based reasoning

Department of Computational Perception

Co-occurrence-based Approaches: Summary

	Web Co-Ocs	Playcounts	P2P nets	Playlists
Source	search engines, Web pages	listening service	shared folders	radio, compilations, Web services
Community-based	no	yes	yes	depends on source
Level	artists	tracks	artists (tracks)	artists (tracks)
Specific Bias	"wikipedia"-bias	popularity	community	low
Potential Noise	high	low	high	low

Department of
Computational
Perception