Effectiveness

- Till now we focused on efficiency aspects
  - i.e., “How to efficiently execute a MM query?”
- It is now time to consider the effectiveness of the MM data retrieval process, which includes everything related to the user expectation!

- Effectiveness in term of:
  - quality of result objects
  - availability of simple but powerful tools, able to smooth the processes of
    - query formulation/personalization
    - result interpretation

Outline

- Quality of the results and relevance feedback techniques
- Use cases
- Demos of some applications

Quality of the results

- Traditional metrics for evaluating the quality of result objects are precision \( P \) and recall \( R \)
Precision and recall

\[
P = \frac{\text{Retrieved and Relevant}}{\text{Retrieved}}
\]

- measures the effect of false hits

\[
R = \frac{\text{Retrieved and Relevant}}{\text{Relevant}}
\]

- measures the effect of false drops

How can a user effectively search?

- Till now we have implicitly assumed that the user “knows” how to formulate her queries
- Although with traditional DB’s and a few attributes this might be a reasonable assumption, when we consider many attributes/features it is not clear how a user might guess the right combination of weights
- How can you define the 64 weights of a color-based search using the weighted Euclidean distance?

The idea of relevance feedback

- The basic idea of relevance feedback is to shift the burden of finding the “right query formulation” from the user to the system [RHO+98]
- For this being possible, the user has to provide the system with some information about “how well” the system has performed in answering the original query
- This user feedback typically takes the form of relevance judgments expressed over the answer set
- The “feedback loop” can then be iterated multiple times, until the user gets satisfied with the answers

Relevance judgments

- The most common way to evaluate the results is based on a 3-valued assessment:
  Relevant: the object is relevant to the user
  Non-relevant: the object is definitely not relevant (false drop)
  Don’t care: the user does not say anything about the object

- Information provided by the relevant objects constitutes the so-called “positive feedback”, whereas non-relevant objects provide the so-called “negative feedback”
  - It’s common the case of systems that only allow for positive feedback
- “Don’t care” is needed also to avoid the user the task of assessing the relevance of all the results
- Models that allow a finer assessment of results (e.g., relevant, very relevant, etc.) have also been developed
A practical example (1)

This is the initial query, for which 2 object are assessed as relevant by the user

Precision = 0.3 (including the query image)

I. Bartolini – MMDBs Course

A practical example (2)

These are the results of the “refined” (new) query, generated using the 1st strategy we will see

Precision = 0.6 (including the query image)

I. Bartolini – MMDBs Course

A practical example (3)

These are the results of the “refined” (new) query, generated using the 2nd strategy we will see

Precision = 0.8 (including the query image)

I. Bartolini – MMDBs Course

A practical example (4)

And these are the results obtained by combining the 2 strategies...

Precision = 0.9 (including the query image)

I. Bartolini – MMDBs Course
Basic query refinement strategies

- When the feature values are vectors, two basic strategies for obtaining a refined query from the previous one and from the user feedback are:

  **Query point movement:**
  the idea is simply to move the query point so as to get closer to relevant objects

  **Re-weighting:**
  the idea is to change the weights of the features so as to give more importance to those features that better capture, for the given query at hand, the notion of relevance

FeedbackBypass case study [BCW00, BCW01]

- New approach to interactive similarity query processing
- Increases the performances of traditional relevance feedback techniques; it complements the role of relevance feedback engines by storing and maintaining the query parameters determined during the feedback loop over time

  - We realized two implementations of FeedbackBypass:
    - The first one is based on Wavelet
    - The second one uses Support Vector Machine (SVM)

Use cases

- Content-based MM data retrieval
- Content-based MM data browsing
- Automatic MM data annotation

  - By focusing on the effectiveness of user provided tools (i.e., interfaces)
    - Query formulation
    - Result interpretation

Content-based MM data retrieval

- This use case is the one we have assumed till now…
- Many content-based MM data retrieval systems (both commercial and research) have been proposed in the last ten years
  - especially for image and video DBs

Windsurf case study: “St. Peter” query
(ABP99, BCP00, BP00, BC03, Bart09a, BCP+09, BCP10)

Visual results: flat visualization

Visual results: spatial visualization

Effectiveness comparison example
Content-based image browsing

- Till now we have implicitly assumed that the user “knows”
  - what she is looking for
  - how to formulate her queries
  - e.g., QBE paradigm
- In some cases the user does not know at all what to look for; in these cases a “browsing” activity should be supported
  - to determine a good starting point for searching
  - to get an overall view of the DB contents
  - to give the user the ability to organize her MM collections (e.g., personal photos albums) in a semi-automatic way

Windsurf case study: flat browsing example

Spatial browsing example

PIBE case study [BCP06, BCP07]

- A novel adaptive image browsing engine
  - customizable hierarchical structure called Browsing Tree (BT)
  - graphical personalization actions to modify the BT
  - “local” reorganization of the DB
    - specific similarity criteria for each portion (sub-tree) of the BT
  - user customizations persist across different sessions
The semantic gap problem

- Characterizing the object content by means of low level features (e.g., color, texture, and shape of an image) represents a completely automatic solution to MM data retrieval
  - However, low level features are not always able to properly characterize the semantic content of objects
    - e.g., two images should be considered “similar” even if their semantic content is completely different

- This is due to the semantic gap existing between the user subjective notion of similarity and the one according to which a low level features-based retrieval system evaluate two objects to be similar
  - prevents to reach 100% precision results
Possible solution

- (Semi-)automatically provide a semantic characterization (e.g., by means of keywords or tags) for each object able to capture its content
  - e.g., ([sky, cheetah] vs. [sky, eagle])
- Combine visual features with tags by taking the best of the two approaches [LSD+06, LZL+07, DJL+08]

![sky, cheetah] ![sky, eagle]

Principles of Scenique

- Provides the user with two basic facilities:
  1) an image annotator (…we will see it in a few minutes!!), that is able to predict new tags for images, and
  2) an integrated query facility which allows the user to search and browse images exploiting both visual features and tags
    - possibly organized in visual and semantic facets
    - in the form of trees
    - default semantic facet to ensure compatibility with systems/devices that do not consider any tag organization (e.g., Flickr)

Scenique case study [BC08b, Bar09]

- Scenique: Semantic and ContENt-based Image QUErying
- Image retrieval and browsing system that profitably exploits both low level features and manually and/or (semi-)automatically associated textual annotations
  - Based on a simplified version of the multi-structural framework [FKK+05] which allows objects, (i.e., images in our case) to be organized into a set of orthogonal dimensions, also called facets
  - Photos of animals I took during my summer vacation

Technical details

- Feature-based retrieval
  - Images as a set of regions
- Semantic-based retrieval
  - Multiple tags associated to images
  - WordNet as lexical ontology [Miller 1995]
    - IS-A relation
  - Semantic “relaxation”
    - Semantic similarity criterion to compare terms [Lin 1997]
- Integration policies
"I want images of bears from the Arctic Ocean that look like the provided one."
Automatically infer semantics to MM objects

- **Automatic objects annotation requires** user intervention
  1. Relevance feedback
     - Exploiting user feedback to understand which are real relevant objects to the query
  2. Learning
     - The system is trained by means of a set of objects that are manually annotated by the user (training phase)
     - Exploiting the training set, the system is able to predict labels for uncaptioned objects: the test object is compared to training objects; labels associated to the “best” objects are proposed for labeling (labeling & testing phases)

Imagination case study [BC08a]

- Imagination: IMAGe (semi-)automatic annotation
- Images as set of regions
  - Labels are tags which are associated at the image level
  - Graph-based approach (à la Page Rank)
    - 3-level of graph objects
      - Images
      - Regions with low level features (i.e., color and texture)
      - Tags assigned to images
      - plus K-NN links computed on region similarities

“Given a new image provide tags that are affine to the image and semantically correlated to each other”
Intuitive example

regions

DB images

tags
deer, grass, bush
bear, rock, grass, water, ground
rock, bear, grass, water, ground

Imagination user interface

predicted tags

Annotation visual example within Scenique

"I want to annotate from scratch the selected image"
The same principles can be applied for the *annotation of videos*!

**SHIATSU:** Semantic-Based Hierarchical Automatic Tagging of Videos by Segmentation using Cuts

- Based on
  - Shot boundaries detection
  - Hierarchical annotations

---

Enjoy some demo applications 😊