# Large Scale Image Retrieval based on User Generated Content

Ximena Olivares

TESI DOCTORAL UPF / 2010

Directors de la tesi

Prof. Dr. Ricardo Baeza-Yates
i
Dr. Roelof van Zwol

*To Linda*

# Acknowledgements

First of all I would like to acknowledge the members of the committee, who have been generous enough to have time to review this work.

I would like to thank my advisor Ricardo Baeza-Yates for giving me the opportunity to enroll in the PhD program of UPF and introduce me to the world of research, which little by little, I've been learning to love. Thanks for your patience, support and expertise delivered through all these years that are reflected in this work.

The completion of this thesis would not be possible without the invaluable help of Roelof van Zwol who has guided and helped me both personally and professionally. Thank you very much for letting me be part of your research team allowing me to meet valuable people with whom I could learn and work. I am not sure where my professional career will lead me, but surely the experience of these years will help me in whatever comes, thanks again for accepting the challenge of being my advisor and also for becoming my friend.

I would like to mention the people who helped me through my time as an intern in the multimedia research group: Lluis, Börkur, Reinier, Vanessa, Adam, Georgina, Massi. Thanks for the advice and contributions. To my fellow classmates Cristina, Liliana and Victor thanks for sharing these years as students, for the help provided in the good and bad times.

It has been 5 years of work, sorrows and joys, where many people consciously and unconsciously helped me to complete this phase. Thanks to Nacho, Pep, Fabiola, Chato, Maria Eugenia, Julia, Marc, Elizabeth, Virginia, Sergi.

A special mention to Nicole for helping me with the introduction, one day you will be my personal editor! Finally I would like to thank Christian for his company along this long path, and above all for being with me in those not so sunny days.

## Abstract

Online photo sharing systems provide a valuable source of user generated content (UGC). Most Web image retrieval systems use textual annotations to rank the results, although these annotations do not only illustrate the visual content of an image, but also describe subjective, spatial, temporal, and social dimensions, complicating the task of keyword based search.

The research in this thesis is focused on how to improve the retrieval of images in large scale context , i.e. the Web, using information provided by users combined with visual content from images. Different forms of UGC are explored, such as textual annotations, visual annotations, and click-through-data, as well as different techniques to combine these data to improve the retrieval of images using visual information.

In conclusion, the research conducted in this thesis focuses on the importance to include visual information into various steps of the retrieval of media content. Using visual information, in combination with various forms of UGC, can significantly improve the retrieval performance and alter the user experience when searching for multimedia content on the Web.

## Resumen

Los sistemas online para compartir fotos proporcionan una valiosa fuente de contenidos generado por el usuario (UGC). La mayoría de los sistemas de recuperación de imágenes Web utilizan las anotaciones textuales para rankear los resultados, sin embargo estas anotaciones no sólo ilustran el contenido visual de una imagen, sino que también describen situaciones subjetivas, espaciales, temporales y sociales, que complican la tarea de búsqueda basada en palabras clave.

La investigación en esta tesis se centra en cómo mejorar la recuperación de imágenes en sistemas de gran escala, es decir, la Web, combinando información proporcionada por los usuarios más el contenido visual de las imágenes. En el presente trabajo se exploran distintos tipos de UGC, tales como anotaciones de texto, anotaciones visuales, y datos de *click-through*, así como diversas técnicas para combinar esta información con el objetivo de mejorar la recuperación de imágenes usando información visual.

En conclusión, la investigación realizada en esta tesis se centra en la importancia de incluir la información visual en distintas etapas de la recuperación de contenido. Combinando información visual con otras formas de UGC, es posible mejorar significativamente el rendimiento de un sistema de recuperación de imágenes y cambiar la experiencia del usuario en la búsqueda de contenidos multimedia en la Web.

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

The rise in popularity of recent on-line photo sharing services, such as Flickr[1] and Picasa Web[2], has produced a very large, continuously growing, online collections of human-annotated digital images. Millions of photos are uploaded and annotated on a daily basis. The metadata provided by users is essential to make photos more easily retrievable by search engines, as keyword-based search is the de-facto model for query formulation on the Web.

However, retrieval models that are generally effective for text retrieval do not work as well for text-based image retrieval. Several factors complicate matters for text-based retrieval. First of all, textual annotations of images (metadata) are rather sparse and short as most users use only a few keywords to annotate their photos. Furthermore, the metadata provided does not solely serve the purpose of describing the visual content of an image. Metadata often includes spatial, temporal, and social references, as well as subjective/personal remarks and descriptions. This further diffuses the results achieved with keyword-based search on images. Finally, the keyword-based query formulation is powerful, but lacks the expressiveness that is inherent in an image. It is difficult for a user to express the visual characteristics of the desired image only using textual clues.

The latter problem has been extensively studied in content-based image retrieval, where the objective is to include the visual characteristics of an

[1] http://www.flickr.com

[2] http://picasaweb.google.com

image into the search process. Using the query by image content (QBIC) search paradigm similar images are retrieved for a given sample image by extracting visual features from all the images in the collection. The downside of this approach is that the user needs to begin the query process with a sample image. Alternatively, high level concepts are derived for the low level features that are extracted from the image content. The problem with this approach which is often referred to as the *semantic gap* problem, introduced by [25], is that for each concept a special concept detector is needed to translate the user information need into low-level image features. The latter makes the approach less suitable for widespread application on the Internet, where arguably no domain restrictions exist.

The main purpose of this thesis is to analyze, investigate, and research how to combine the metadata provided by the user with visual content information to improve the retrieval of images on a broad Web domain. To achieve this, different types of metadata available will be explored, and state-of-the-art algorithms and technologies that can be successfully applied on a large scale image collection such as the Web collection, will be researched.

## 1.1 The Wisdom of User Generated Content

As stated, Web image repositories contain a large amount of user annotated data, which allows to find implicit knowledge from the users. The problem with this knowledge is that it can be of questionable quality or be insufficient. Even if the data is good by its own, it does not provide information of the content of the images. Hence, to use the knowledge of *User Generated Content* (UGC), it is essential to combine this information with the image content.

The contents of an image can be described in a simple, efficient, and large-scale compatible way, using the appropriate features. This information can be combined with the metadata provided by the users (UGC), to improve the retrieval of the images in terms of precision.

The work in this thesis will be based on an image collection gather from over Flickr images. This will require working with images with high variability, but that contain millions of annotations provided by Web users.

The first task is to explore how visual annotations, e.g. notes in Flickr, can be used to enhance the retrieval performance of keyword-based queries. Using Flickr notes, users can highlight a certain region in the photo and

associate a tag (label) with this region. Although the specific intent might differ, people annotate notes in a similar fashion as they annotate images, which makes notes a good source of information for visual queries. Keyword-based queries that represent objects were associated to visual annotation provided by the users. For each text-based query, a set of visual annotations is retrieved, and for each visual annotation, a ranked list of similar images is obtained, based on their visual information. The next step is to combine these results to obtain the final ranked list. Rank aggregation over the partially ranked lists is proposed to build the final ranking. Results obtained using this approach are compared against systems using only content-based retrieval algorithms, only text-based retrieval algorithms, as well as different combinations of these systems. Results showed that using a combination of visual and textual information outperforms the results obtained using the other combinations.

The aggregation problem has been addressed by meta-search engines, where the results of different search engines are merged into one ranked list. In addition, different approaches for combining rank lists are explored: positional methods, such as Borda count model; or comparison based methods, such as Markov chains models. Also, a pre-retrieval aggregation method is compared with the rank aggregation ones. Results are obtained using each of the models, and the trade-offs involved are compared. Borda count model has been widely used in different scenarios, since it is a simple and efficient algorithm with good performance. On the other hand, Markov chain-based models require more sophisticated algorithms, but have been used in Web meta-search engines and proved to outperform Borda. The pre-retrieval aggregation method proposed leads to a gain in efficiency, maintaining high level of precision. To the best of our knowledge, this is the first time this rank aggregation approach has been applied to keyword-based image retrieval, using a combination of visual attributes and user generated data.

In addition to aggregation of visual annotations, one of the objectives is to improve the results obtained by differentiating between high and low quality annotations. To achieve this, different approaches to select the best annotations are studied that will lead to better aggregated results.

The need to process large collection of images has lead to the design of a media extractor that can easily extract visual features and textual information from the images, to obtain a combined system in a parallel fashion. Scalable methodologies are proposed to process a large scale image collection in an efficient manner using parallel computation as well as a description on how

to build an image retrieval system in an efficient way.

Next, another user generated content is studied: query logs. The visual information of the images is combined with data obtained from query logs extracted from the Yahoo! search engine. A hypothesis is that the click-through data contained in the logs is of great importance. When a user queries for an image, the search engine returns a list of thumbnail images, allowing the user to preview the results. As a consequence, the set of images that are clicked can be smaller, but the quality of the assessment much better. A learned framework for ranking images that employs click data is proposed to predict clicked result with high accuracy, combining textual and visual information.

Finally, the visual diversification of the images results is studied. The creation of a visually diverse ranking of the images results is proposed by using the visual content of the images.

## 1.2 Objectives

The thesis is centered around four main objectives and their derived hypotheses which are presented below. The common theme is the use of user generated content and the visual information depicted in the image itself.

### Objective 1: Use UGC to Improve Image Retrieval

Boost the performance of the image retrieval process, in terms of precision, by combining different types of user generated content (UGC). Usually, UGC is considered to cover textual information provided by the user (e.g. title, description), but it can be extended to the visual annotations provided by the users.

**Hypothesis 1** The use of UGC and visual information, specifically visual annotations, will improve the results of an image retrieval system in terms of precision.

**Hypothesis 2** Aggregating the results of different visual annotations, for the same topic, will significantly improve the retrieval performance in terms of precision. The agreement between the different result sets of the partial

searches will lead to a more focused result set for the aggregated result set, with a higher precision at the top of the ranking.

### Objective 2: Investigate Scalability of using Visual Information

Investigate scalable solutions for processing and analyzing visual content. Computing visual features is an expensive computational process, especially for Web-size collections.

**Hypothesis 3** Media content is available in large quantities, mostly unannotated. Image analysis and retrieval techniques do not live up to their textual counterparts in terms of retrieval performance. However large-scale applications of the techniques will improve performance.

### Objective 3: Use Click-through Data to Improve Image Retrieval

Research how using other forms of UGC, such as click-through data, that has been successfully used in Web text retrieval can improve image retrieval performance.

**Hypothesis 4** Click feedback, provided by users searching for images on the Web, can be used as an assessment of the image, providing strong signal of the quality and type of images a user is really looking for, besides its topical relevance.

### Objective 4: Use Visual Content to Improve Retrieval in Terms of Diversity

Though powerful in its simplicity, keyword-based query formulation does not allow a user to fully express the visual characteristics of their information need. Therefore alternative query formulations have been proposed. In this thesis, this issue is addressed by investigating how search result sets can be diversified to better address the various users needs.

**Hypothesis 5**

Based on the visual analysis of the content is possible to provide a meaningful clustering of the images and provide diverse search results.

## 1.3 Contributions

The research presented in this thesis extends the state of the art in large scale image retrieval by deploying UGC that now is available in vast amount. In particular, this has lead to multiple publications. The research presented in Chapter 3 has been published in ACM MM 2008 conference [40] and submitted for publication [41]. The media extraction framework described in Chapter 4 is used in the European project on Search Environments for Media (SEMEDIA) and in the applications presented in Chapters 5 and 6. The research of Chapter 5 is published in book chapter [38] and Chapter 6 is published in WWW 2009 [55]. The overall list of publications is:

- X. Olivares, M. Ciaramita, and R. van Zwol. Boosting image retrieval through aggregating search results based on visual annotations. In MM'08: Proceedings of the 16th ACM International Conference on Multimedia.
- R. H. van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual diversification of image search results. In WWW '09: Proceedings of the 18th international conference on World wide web.
- V. Murdock, R. van Zwol, L. Garcia, and X. Olivares. Image retrieval in a commercial setting. In ImageCLEF: Experimental evaluation in Image Retrieval, Springer 2010.
- X. Olivares, R. van Zwol, and R. Baeza-Yates. The Power of Visual Annotations in Image Retrieval. Submitted for publication.

## 1.4 Thesis Outline

The work presented in this thesis is driven by the objectives presented above. First, Chapter 2 presents a literature review of the different research areas addressed in this thesis. Objective 1 relates to Chapter 3, where the focus is

on image object retrieval. Objective 2 is addressed in Chapter 4, which discusses techniques and methodologies for large scale image processing. This enables the research described in Chapter 5 and 6 which address Objectives 3 and 4.

Forthcoming from the objectives are 5 hypotheses, which are validated by empirical evidence collected through experimentation. This leads to the conclusion presented in Chapter 7.

Chapter 2

# Related Work

The work presented in this thesis combines different fields of research. This chapter presents a literature review on the following fields and sub-fields:

- Text Based Image Retrieval
- Content Based Image Retrieval
  - Content Based Object Image Retrieval
  - Image features (global and local features)
- Large-scale image retrieval
- Rank Aggregation
- Machine learning in image retrieval
- Visual Diversification in image retrieval

## 2.1 Image Retrieval

### Text Based Image Retrieval

Photo sharing systems allow users to search image collections by submitting a keyword-based query. Images are then ranked according to their relevance

by means of text retrieval models. This type of search is based on the text that describes the images, such as their title, description, and tags. Photos in Flickr contain different types of meta-data, ranging from technical details to more subjective information. At a low level information concerns the camera, shutter speed, rotation, etc. At a higher level, the user that uploaded the image can include a title and description, which are more likely to be used to describe the image as a whole. The use of tags permits the user to describe what he thinks is relevant about the image using simple keyword combinations.

Ames and Naaman [1] present a qualitative study describing the motivations behind users tagging their pictures. They defined two main dimensions of the motivations: social and functional, and characterized the motivations whether they were used for themselves or their family (social), or as a way of complementing the context of the image (functional). Furthermore, Dubinko *et al.* [16] show that tags not only describe the specific contents of the images, but also additional information. They observed recurring categories such as: events (e.g. "Valentine's days" or "Thanksgiving"), personalities (e.g. "Pope"), and social media tagging (e.g. "What's in your fridge"). Another important characteristic of tag-based systems is the way people use the tags. Marlow *et al.* [33] analyzed tags used to describe images, and observed that most users use a few distinct tags, while a small group of users use large sets of tags.

The high degree of variability characterizing image tags has an impact on retrieval. As a consequence, the results of search systems based exclusively on tags is noisy and sub-optimal. Therefore, it is important to include intrinsic information about the image visual content in the retrieval process. Furthermore, this thesis focuses on approaches that can be applied at large on the Internet without training a large set of concept detectors [47]. The literature in this area is extensive so the scope of this thesis is limited to the state of the art in image object retrieval.

### Image Object Retrieval

Sivic and Zisserman [46] introduced the *bag-of-visual-words* architecture, and successful results have been reported for object retrieval on a large image collection containing buildings in Oxford. For every image in the collection, affine regions are extracted, and described by a SIFT [30] descriptor. This set of vectors are quantized to build a visual vocabulary as proposed in [14].

This approach allows to represent every image as a set of visual words, hence making it possible to describe them with a weighted vector and use standard text retrieval techniques to determine the similarity between images. Since the spatial arrangement of the visual words is crucial, they add a simple constraint to the spatial distribution of words. This approach is used as the baseline in Section 3.1. In [46] a region of a video frame is selected to obtain all the frames in the video where the selected object appears. Based on this work, Philbin *et al.* [43] have built a large-scale object retrieval system using a combination of images extracted from Flickr and images from the Oxford building database. A query image is submitted and a set of ranked images is returned. They presented considerations for creating a visual vocabulary and tested their results using building landmarks. In addition, Chum *et al.* [11] propose a query expansion approach where some of the retrieved images were used to reformulate the original image query improving the results obtained in [43]. The research outlined above uses image queries as input for the retrieval system.

One of the hypothesis of this thesis states that information obtained from tags, as well as the visual information contained in the image, should be combined to obtain better results, allowing to successfully use a keyword-based query to search over the image collection. In [29], an analysis is proposed of the patterns existing between the visual words of images that share a common set of tags.

Section 3.1 uses visual annotations, e.g. notes in Flickr, that associate a label to a region of the image. An example of visual annotation is shown in Figure 3.3. This type of annotations are valuable, since the associated text is typically highly relevant to the highlighted area in the image, and possibly less subjective in nature (i.e. more descriptive) than tags associated with the whole image. Given a text query, it is possible to obtain a set of visual annotations that can be used to search over the collection based on the content of the image. This will lead to partial lists of results from several sources, each of them from different visual annotations. It is then necessary to decide how to merge these results.

## 2.2 Rank Aggregation

This problem can be compared to the problem of determining the ranked list of winners in an election. A simple, and commonly used method is the Borda count model [5] that assigns a score to each element in the set of

ranked lists, and then sums the scores for each individual element. Various methods have been proposed for rank aggregation on the Web, in the context of meta-search engines [19, 17, 2]. To the best of our knowledge, this thesis is the first time this approach is applied to keyword-based image retrieval using visual annotations.

Aslam and Montague [2] investigate the problem of meta-search and compare different models. Their results show that Borda count is a simple and efficient algorithm with good performance. For this reason, this mechanism is used in the aggregation stage of the system presented in Section 3.1. One of the objectives of this thesis is to combine textual and visual information to improve retrieval performance, specifically precision.

### Rank Aggregation using Markov Chains

As defined in [17], the *rank aggregation* problem consists on ranking a list of several alternatives, based on one or more criteria, to obtain the best possible aggregation based on a consensus ranking of the alternatives. They studied the rank aggregation problem on a Web context, based on the existence of several search engines (general and special purpose) where none of them has proved to outperform the others.

Dwork *et al.* [17] study is motivated by the existence of Web spam, and they want to provide users with robustness of search overcoming the shortcomings and biases that each individual search engine can introduce. Their objective is to develop a robust technique for meta-search.

They propose using the Kemeny method, originally used in social choice theory, which minimizes the total disagreement between the different rankings and the final aggregation. One problem is that computing optimal solutions based on Kemeny's approach is NP-hard.

To measure the disagreement between the partial lists and the aggregation, the distance between the lists is used. Kemeny optimal aggregations has the property of, given a set of lists $t_1, \ldots, t_k$, the aggregation $\sigma$ minimizes the value of $K(\sigma, t_1, \ldots, t_k)$, where function $K$ measures the distance between the lists. The Kemeny optimal aggregation is the only one that satisfies rank aggregations Condorcet property. If there exists an alternative that defeats every other alternative in pairwise simple majority voting, it must be ranked first and this alternative is the Condorcet alternative.

Since computing Kemeny optimization is NP-hard the authors introduced a relaxation of the Kemeny optimality that ensures the Condorcet criterion. A list $\pi$ is a locally Kemeny optimal aggregation for partial results $t_1, \ldots, t_k$ if there is no $\pi^{'}$, obtained by single transposition of an adjacent pair of elements, for which $K(\pi^{'}, t_1, \ldots, t_k) < K(\pi, t_1, \ldots, t_k)$. It is also shown that using any initial aggregation of partial lists a locally Kemeny optimal aggregation can be constructed. The Condorcet criterion is considered as a spam-fighting property for meta-search.

In their work they compared Borda count aggregations with Markov chains methods. They proposed 4 specific Markov chains (presented in detail in Section 3.2) and they evaluated the results based on the distance between the candidates and the final aggregation. In their experimental setup, Markov chains outperforms Borda method.

The inclusion of this specific work relies in the fact that it has been broadly used for meta-search and as literature review. All rank aggregation works are based on Dwork *et al.*, introducing some minor variations.

The purpose of using Kemeny optimality is to ensure Condorcet criterion, reducing spam that could have been introduced in one of the search engines. Since the work presented in this thesis (Section 3.2) is based on only one document collection, there is no concept of "spam", hence the Kemeny optimization is not required.

## 2.3 Machine Learning on Image Retrieval

The related work presented in this section is focused on learning to rank images combining visual and textual information in a large scale collection using machine learning techniques.

Machine learning techniques have been used to solve multiple tasks related to image retrieval. For example, Support Vector Machines (SVMs) have been used in a variety of image classification tasks (such as [8, 24]). San Pedro and Siersdorfer [42] employ a classifier and low-level visual features as well as textual features to predict the attractiveness of an image. They selected as positive examples all photos with at least two favorite assignments from a large crawl of photos from Flickr, and similarly sized sample randomly drawn from photos that had not been indicated as favorite. They find significant improvements from the combination of textual and visual features.

Application presented in Chapter 5 does not attempt to identify attractive images, but to prioritize images that are clicked since they embody relevance, quality, attractiveness, interestingness, freshness, and other indefinable properties. Because of this the textual features used reflect the similarity between the query and the textual metadata, as opposed to the attractiveness of the image. None of the features used in [42] are employed in this work.

Central to the problem of image ranking is the problem of relating textual queries to visual image content. Tong *et al.* [53] propose a propagation method based on a manifold learning algorithm. A small portion of the images are labeled with keywords, and then the labels are propagated to the rest of the collection. Each image is assigned a relevance score for a given keyword by constructing a matrix where each row represents an image, each column represents a keyword, and each cell contains a relevance score. The intuition is to associate textual information with visual content. The experiments were conducted over a collection of 5,000 images extracted from the COREL data set.

Tong and Chang [54] elicit explicit relevance feedback from users, and then employ active learning with a support vector machine, using features derived from the color and texture of an image to improve retrieval results. Although the application presented in Chapter 5 uses similar features, the main difference is that they use explicit feedback elicited from the user, while in this work, it is used implicit feedback in the form of clicks.

Cheng *et al.* [10] proposed a scalable relevance feedback mechanism using click data for Web image retrieval. Using Rocchio feedback, they add the vector of features representing the query to an "optimal query", which is the mean of the vectors of the clicked images. They rank images according to the cosine similarity between the new query vector and the feature vectors describing the images in the collection. Their textual features are based on *tf.idf* scores of the query and metadata associated with each image. Their visual features are a combination of three color features (color moment, auto-correlogram, and color texture moment). They evaluated their system in a simulated setting, using ten queries, retrieving from a collection of three million images crawled from photo sharing web sites.

Learning to rank from click data was first proposed by Joachims [27] for document retrieval. This thesis adapts his ranking mechanism to image retrieval. Joachims proposed that user clicks are an indicator of relative relevance. That is, a click at rank $j$ indicates that the document at rank

$j$ is more relevant than unclicked documents preceding it in the ranked list. Joachims work is the first in a series investigating using click data for learning ranking functions. Elsas *et al.* [18] extend this idea by learning ranking functions from search results with a committee perceptron using the LETOR dataset [52].

Ciaramita *et al.* [12] successfully adapted Joachims unbiased model to several learning frameworks: binary classification, ranking, and non-linear regression and showed positive results on a sponsored search task using commercial query log data. They demonstrate that a multilayer perceptron outperforms both the linear perceptron and a ranking perceptron. In their work, the features of an ad-query pair are based entirely on the textual representations of the ad. Their work differs in that the search engine is most interested in generating clicks on ads, thus learning to predict clicks is key to the task of ranking ads. In the case of image search, the search engine would like to encourage people to use the search engine, and thus attempts to maximize the relevance of the search results. On the other hand, the application described in Chapter 5, adopted their framework both for training and evaluation, and the images are represented by text in much the same way that ads are. Whereas ads are represented by keywords, titles and a short description, images are represented by tags, titles and a short description. Keywords and tags differ in character, but are similar in their brevity and conciseness. Their work is extended by showing how this unbiased framework, which is based on the bias introduced by a linear presentation of the results, can be applied to non-linear presentations of the results, such as is the case in image search. In addition, it is extended by considering the visual representation of the data rather than just its textual representation.

## 2.4 Visual Diversification

Based on the objectives of this thesis, in particular, Objective 5 that refers to provide diverse image search results, it has been suggested as hypothesis that this can be achieved by the use of image clustering. The following section first discusses the state of the art in image clustering, and then presents the literature on diversifying search results.

### Image Clustering

Most image clustering techniques are not dynamic, and therefore not suitable for clustering image search results. Due to the broad domain of the task, this work will focus only on unsupervised clustering techniques, which makes techniques such as presented in [26] unsuitable for this task. Furthermore, clustering techniques often partition the entire database to facilitate faster browsing and retrieval [21].

In [37] a method for extracting meaningful and representative clusters is presented that is based on a shared nearest neighbors (SNN) approach that treats both content-based features and textual descriptions (tags). They describe, discuss, and evaluate the SNN method for image clustering and present some experimental results using the Flickr collections, showing that their approach extracts representative information of an image set. Such techniques are often effective, but require extensive processing power to produce a final clustering. When clustering image search results, the input varies depending on the user's query and it is essential that the clustering technique is not only effective, but the results can be efficiently computed.

In Cai *et al.* [7] the problem of clustering Web image search results is studied by organizing the results into different semantic clusters that facilitates users' browsing. They propose a hierarchical clustering method using visual, textual and link analysis that is mainly targeted at clustering the search results of ambiguous targets. In a related paper by Wang *et al.* [59], they evaluate a different approach, named IGroup, for semantic clustering of image search results, based on a textual analysis of the search results. Through a user study they report a significant improvement in terms of efficiency, coverage, and satisfaction.

### Diversity in Search Results

In [60], they present a method for detecting and resolving the ambiguity of a query based on the textual features of the image collection. If a query has an ambiguous nature, this ambiguity should be reflected in the diversity of the result set. Furthermore, in [57] it is presented how the topical (textual) diversity of image search results can be achieved through the choice of the right retrieval model.

The objective of the application presented in Chapter 6 is to focus on visual diversity of the search results. The solution presented for the visual diversity

builds upon the results of these two papers, as it takes as input the ranked list of images produced by the retrieval models for topical diversity.

In Zhang *et al.* [62] diversity of search results is examined in the context of Web search. They propose a novel ranking scheme named Affinity Ranking to re-rank search results by optimizing two metrics: diversity and information richness. More recently, Song *et al.* [49] also acknowledge the need for diversity in search results for image retrieval. They propose a re-ranking method based on topic richness analysis to enrich topic coverage in retrieval results, while maintaining acceptable retrieval performance.

Zeigler [63] studied topic diversification to balance and diversify personalized recommendation lists that reflect the user's complete spectrum of interests. Although their system is detrimental to average accuracy, they show that the method improves user satisfaction with recommendation lists, in particular for lists generated using the common item-based collaborative filtering algorithm. They introduced an intra-list similarity metric to assess the topical diversity of recommendation lists and the topic diversification approach for decreasing the intra-list similarity.

In a different setting, Vee *et al.* [58] propose a method to return a set of answers that represent diverse results proportional to their frequency in the collection. Their algorithm operates on structured data, with explicitly defined relations, which differs from the setting in this work, which aims to diversify through visual content based on a dynamic ranking strategy, rather than using predetermined fractions.

The main purpose of this chapter was to give an overview of the most important literature in the research area, which several other works have been based on. Additional references are included in the chapters.

Chapter 3

# Image Object Retrieval

One way of enhancing the image object retrieval task is by exploiting the additional information provided by the users, to describe the contents of the image. This information is considered *User Generated Content* (UGC), and is provided either by the owner of the picture, or by the people interacting with the images.

Usually, UGC is considered to cover information such as the title, description, and tags linked to an image, but it can also be extended to the *visual annotations* provided by the users. This type of annotations associate a portion of the image to a textual description, making it possible to add semantics to specific regions of the image.

This chapter presents the results of combining different types of UGC, such as tags and visual annotations, to boost the performance of the image retrieval process. This process can be used as a proof of concept to extending the framework to a large scale image retrieval system, leveraging scalability and retrieval performance.

## 3.1 Image Object Retrieval based on Visual Annotations

This section proposes to use rank aggregation for merging the result sets obtained with a content-based image retrieval system that is fed with the visual annotations matching a given keyword-based query. Rank aggregation

is primarily used by meta-search engines, where the results from different search engines are merged into a new ranked list of results.

## Content-based Image Retrieval

This section describes the architecture of the image retrieval system. The system extends the framework proposed by Sivic and Zisserman in [46, 43] to handle the retrieval of photos based on visual characteristics. They successfully applied their framework on a domain-restricted collection to detect the same object in different photos. In their experiments they focused on detecting near-identical representations of buildings in Oxford. Their results are promising both in the dimension of scalability and retrieval performance.

In short, the framework consists of the following steps, for which a parallel with text retrieval can be made:

1. First, extract visual features (salient regions) from the images in the collection, and describe them with a high-dimensional descriptor.
2. Then, build a visual vocabulary from the high-dimension descriptions by quantifying and clustering them into a vocabulary of visual words. In this step the high-dimensional descriptions are lemmatized into similar visual words. This allows to describe each image as an histogram of visual words.
3. Based on the *bag-of-words* approach, it is possible to use existing text-retrieval models to build an index over the image collection, and similar images can be found using the "query by image content" paradigm.
4. Finally, a post-retrieval step is required to re-rank the results, taking the spatial structure of the image into account. This step is significantly more important in image retrieval than in text retrieval [46].

The following sections present a more detailed outline of this approach, complemented with some of the implementation specifics used in the experiments.

### Feature Extraction

In previous work several approaches to extract visual information (features) from images have been proposed [36]. A combination of these features is

(1) Extracted regions. (2) Overlaying a visual annotation. (3) Features describing the object.

Figure 3.1: Example feature extraction.

typically used to retrieve similar images. In the experiments presented in this chapter, the feature extraction process has been limited to extract only high-dimensional region descriptors from images, based on *Harris affine* and *Hessian affine* regions, as introduced by [35]. This is due to their invariance to rotation, translation and scale. The Harris affine regions are based on the points extracted with the Harris detector, which are later processed obtaining affine viewpoint covariant regions that represent corner structures. On the other hand, Hessian affine regions are based on processing the points obtained by the Hessian detector, resulting in affine viewpoint covariant regions, which represents blob structures.

When processing the image collection an average of 1,000 Harris regions and 1,066 Hessian regions per image were extracted. Each region is then described using a 128-dimension SIFT [30] descriptor vector. Figure 3.1.(1) shows the extracted Harris regions for one of the images in the collection. When a visual annotation is drawn over the image to mark an object, it is possible to select only the feature descriptors that are inside the bounding box of the annotation (see Figure 3.1.(2)), and ultimately, as shown in Figure 3.1.(3), use only those features to describe the object as input for searching.

### Visual Vocabulary

Once features have been extracted from the images in the collection, a visual vocabulary needs to be build. A vocabulary of size $k$ can be generated by clustering the SIFT descriptors into $k$ clusters. Based on a learned clustering model a *visual word* (cluster label) is associated with all the elements

contained in a cluster. Clustering large amounts of data, for large values of $k$, as in this case where $k$ can be in the order of tens of thousands, is a challenging task. As shown in [30, 28] approximate k-means clustering can adequately scale up for this type of task. Similarly, for the current experiments, it was implemented an approximate k-means algorithm paired with a kd-tree on the cluster set. Search for the nearest neighbor in the tree is carried out using a priority queue for the nodes, which are ranked according to the distance of the nodes hyper-rectangle from the query point. Search terminates when: (a) the queue is empty, in the case where the exact nearest neighbor has been identified; or (b) after reaching a maximum number of comparisons, in which case the result is only an approximation of the closest neighbor.

The clustering model, uses only one kd-tree, rather than several randomized ones, since the preliminary test showed limited benefits from using several trees, over one tree with a higher threshold for the maximum number of comparisons. The maximum number of comparisons was set to 1,200.

To learn the clustering model a set of 1 million SIFT descriptors randomly selected from the image collection was used. Then, tests were conducted using various sizes for the vocabulary, ranging from 1,500 to 10,000 clusters to determine the benefit of each scenario. Based on this results, the experiments presented in this chapter use a vocabulary of 10,000 words. Finally, the remaining descriptors are classified based on the learned k-means model.

To reduce the noise in the results, "outliers descriptors were removed from the set, where an outlier is a datapoint whose distance to the nearest centroid is greater than the average distance in that cluster plus twice the standard deviation of these distances. Similarly to stop-words filtering in text retrieval, the top 2.5% of the clusters with the largest population were also removed. Finally, three vocabularies are created: (a) an independent vocabulary of 10,000 words based on Harris affine features; (b) an independent vocabulary of 10,000 words based on Hessian affine; and (c) a vocabulary of 20,000 words created by merging the other two vocabularies.

### Vector Space Model

Following the traditional bag-of-words approach for text retrieval, an image can be represented as a weighted-term vector in the vector space model. Using the analogy with text retrieval, a tf-idf weight of the visual words is used to create the corresponding vector. The similarity between the images

can then be measured by calculating the cosine similarity of the weighted vectors, obtaining a normalized value in the range [0, 1]. Alternatively, it is possible to use one of the object's annotations to search the vector space, and find images that are likely to contain the object.

### Spatial Coherence Filter

A limitation of the bag-of-words approach is that all spatial information contained in the image is lost. Although two images can have a high degree of cosine similarity, the relative spatial coherence of the visual words between these two images can be low, which indicates that they are visually not similar at all. Therefore, an analysis of the spatial arrangement of the visual words between the query image and each of the retrieved images is needed, as also argued in more detail in [46, 43].

For this experiment a simple spatial coherence filter. was implemented. For every common visual word between two images, it is analyzed the common visual words contained in the surrounding area. This spatial constraint generates an additional similarity measure that is used to discriminate images that only have the visual words in common with the ones that also satisfy the spatial distribution of the elements.

## Aggregated Search with Visual Annotations

Previous chapters discussed how users annotate images at an on-line photo sharing services such as Flickr. In particular, users can attach labeled notes to the photos published on Flickr. Though not as popular as the photo tags, *notes* can be valuable to learn different visual representations of an object. This observation leads to one of the contributions of this thesis, where the aim is to improve the retrieval performance by aggregating the result sets for searches with visual object annotations in photos. Although the use of textual information in Web image retrieval systems has matured, the hypothesis is that it can be improved by complementing it with visual information, especially when the user's information need is specific and can not easily be described by a combination of keywords.

The widespread availability of visual annotations in Flickr provides a base collection of annotated objects where for each high-level concept a set of visual annotations is available that can be used to aid the user in his search.

The portion of the image enclosed by the visual annotations contains a set of visual words (as defined in Section 3.1), that are mapped to a particular concept defined by the text describing the visual annotation. When a user submits a text query, the system will use the visual annotations to obtain images that answer the user query. Each of these annotations are used to search for similar images, using the cosine similarity between the images that have a textual annotation that matches the user query. As a result, for every visual annotation, a set of similar images is obtained, which are re-ranked using the spatial coherence filter described in Section 3.1.

In Figure 3.2 the results for the query "apple logo" are shown. The upper three rows show the top 10 search results using three different visual annotations. To limit the search space, the search has been filtered on the image tags. Obviously, this already improves the results when searching with a single visual annotation. In the experiment of Section 3.1 it will be presented a comparison of: (a) tag-only, (b) tag & visual, and (c) visual search, that illustrates how the retrieval performance is influenced for each of the different combinations. The bottom row of Figure 3.2 shows the aggregated results.

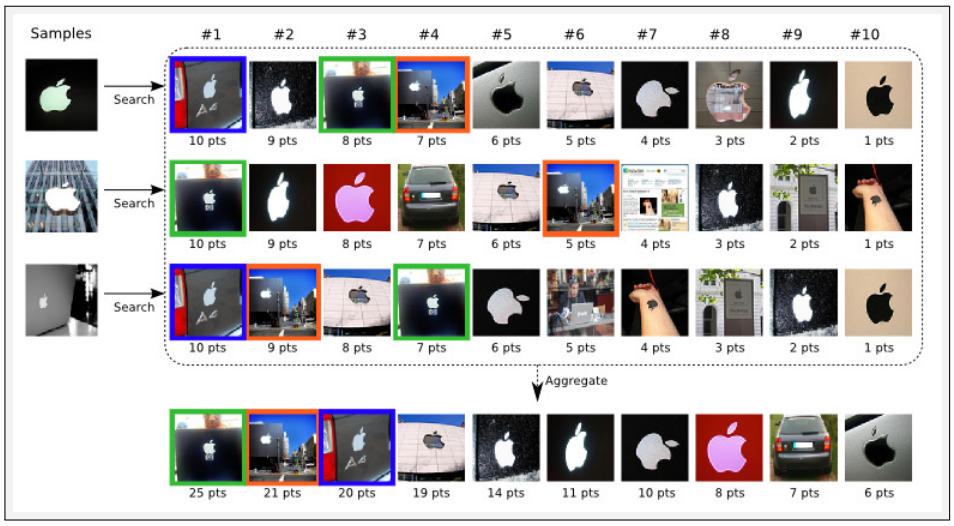



Figure 3.2: Aggregating the search results for the query "apple logo" using visual annotations.

The results from each of the visual annotations can be seen as individual sources of information that need to be merged into a single set of results.

This problem is similar to the one of a metasearch engine that needs to combine search results, or essentially, the combination of any set of ranked lists. Using the ranked position of the images, the results are merged using a voting mechanism introduced by Borda [5]. He models this problem as a set of *voters* (in our case each visual annotation) that must sort a set of *candidates* (the set of results) by assigning points to each of them, and a final list of ranked candidates must be obtained. For this, every voter assigns points to each of the candidates based on their position in their ranked list. The first element in the list is assigned with $k$ points, the second element is given $(k-1)$ points, until the last element is assigned 1 point. To obtain the final list of results, all the candidates are sorted by their total number of points.

The aggregated ranking favors images that are ranked high in several of the partial rankings. Whereas outliers, e.g. those results only retrieved by one of the sample images, will be demoted in the aggregated ranking. The intuition is that even though they match the textual tag, their content might not match the concept behind the query. Figure 3.2 shows a diagram of the aggregation process. For each of the samples their ranked list of results is presented. Every result image is assigned points according to their position in the list. This is illustrated in the first three rows. Finally, the aggregated results corresponds to the list of candidates, sorted by their total number of points, as in the last row. We can observe that the rank of the image returned in first position is a combination of the partial ranks, and likewise for the subsequent results.

## Evaluation

This section describes the set-up and outcome of the retrieval performance experiment that was performed to compare tag-based search, visual search based on sample object annotations, and aggregated visual search based on object annotations. First the hypotheses behind the evaluation are presented. Then the description of the experiments' set-up , and finally the results.

### Evaluation task

To evaluate the performance of the retrieval system, hypotheses 1 and 2, presented in Chapter 1, can be described in detail as follow:

H1: Rank aggregation over the results sets of content-based image retrieval with the visual annotations will significantly improve the retrieval performance in terms of precision. The agreement between the different result sets for the partial searches will lead to a more focused result set for the aggregated result set with a higher precision at the top of the ranking.

H2: Tag-based search combined with content-based image retrieval, using visual annotations will improve the retrieval performance, in terms of precision. When performing a textual search over an image collection a rather diverse set of results will be retrieved, as the annotations are usually very sparse and the textual clues do not allow for visual disambiguation. When searching with visual annotations it is possible to discover the different aspects of an object, and in combination with a filter on the textual annotations it is possible to retrieve more relevant results at the top of the ranking.

### Experimental Setup

The experiment consisted of: (a) defining a set topics to be used as queries; (b) implement and compare five different systems to test the forementioned hypotheses; (c) and collect relevance judgements on the results obtained by each system. The relevance assessment was done in a TREC-style fashion.

Below, the details of the different facets of the experiment are described.

**Image Collection** Different image collections have been used for object recognition, such as the CalTech collection [22], COIL collection [39], and the Corel collection [13]. They are widely used for object classification, recognition, and categorization tasks. The main characteristic of these collections is that they have well defined visual attributes for the objects represented in the images. Usually they contain images with uniform size, and low level of cluttering, which is not coherent with the scenario on the Web, where diversity is present on all possible dimensions. Although some of these collections were created by downloading images from Web pages, they have been manually selected to match a set of constraints. The current work will focus on images with high variability, Web-extracted, and annotated by Web-users. For this reason, instead of using one of the previous collections, a set of images collected from Flickr is used, without manually selecting them.

The collection contains 12,000 images were crawled using the public Flickr API, based on a set of tags that correspond with the topics selected for the experiment. As a result a set of images have been obtained that at least had one of the tags, but no restriction was made on whether they were relevant to their surrounding tags, or whether the object actually appeared on the image. In addition, for each photo the title, tags, and description have been collected. The collection contains 59,693 unique tags (from a total of 229,672 tags).

Photos in Flickr are made available in various resolutions, ranging from thumbnail size to the original size uploaded by the user. To leverage the number of features that can be extracted from the image and its corresponding processing time, the collection uses the medium size image, which have a resolution of at most 500x333 pixels.

Figure 3.3: Examples of visual annotations (notes) for a telephone booth.

**Topics** Using Flickr search logs, a set of 30 topics was derived. To obtain these topics, the queries were sorted by descending frequency, and then filtered for objects, then selecting the top 30 topics. Based on the the topics, they can be classified into four broad categories: (a) *fruits & flowers*, (b) *monuments & buildings*, (c) *brands & logos*, and (d) *general objects*. Table 3.1 shows the list of selected topics. For each topic a short description is defined that details the visual requirements, which can not be easily expressed

in keyword-based search. This additional information was used to guide the assessors in their judgements.

| Topic | Description |
|---|---|
| American flag | Picture of a cloth-made American flag. |
| Big Ben clock tower | View of the clock tower. |
| Arc de Triomphe | Front view of the arc. |
| Clock | Round mechanical clock. |
| Coke can | Can of coke. |
| CN tower | View of the skypod. |
| Dice | Any view of a dice. |
| Eiffel tower | Picture of the tower, taken from the base. |
| Engagement ring | Upper view, containing a stone. |
| Guitar | Body of a classical or electric guitar. |
| Soccer ball | Picture of an official-size soccer ball. |
| Statue of Liberty | Top view of the Statue of liberty. |
| Apple logo | Logo from Apple brand. |
| Rose | Top view of a rose. |
| Parthenon | Front facade. |
| Strawberry | Picture where the skin of the fruit is clearly shown. |
| Daisy | Top view of a daisy. |
| Moai | At least one visible Moai statue. |
| Sunflower | Top view of a sunflower. |
| Sushi roll | Piece of a cut sushi roll. |
| Golden Gate bridge | View of at least one of the main pillars. |
| McDonald logo | Big "M" from the McDonald logo. |
| Taj Mahal | Taj Mahal front facade. |
| Hot air balloon | Fully inflated hot air balloon without the basket. |
| Petronas Twin Towers | View of both towers with the skybridge between them. |
| Telephone booth | Classic UK red telephone boxes. |
| Butterfly | Picture containing the butterfly's wings. |
| Converse | Converse sneakers. |
| Watermelon | Watermelon showing the skin. |

Table 3.1: List of topics.

In addition to the topic descriptions, a visual example for each topic is presented, as depicted in Figure 3.4. Finally, for each topic, a set of 10 visual annotations was created which are used to feed the content-based image retrieval system with the visual examples. For example, see the annotations shown in Figure 3.3.


(1) CN tower.


(2) Apple logo.


(3) Arc de Triomphe.


(4) Eiffel tower.


(5) Big Ben clock.


(6) Guitar.


(7) Petronas towers.


(8) Statue of liberty.


(9) Taj Mahal.


(10) Moai.

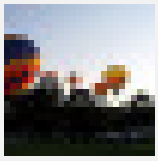

(11) American flag.

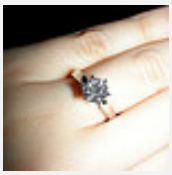

(12) Coke can.


(13) Converse sneakers.


(14) Brooklyn bridge.


(15) Dice.


(16) Hot air balloon.


(17) Engagement ring.


(18) Golden Gate bridge.


(19) Daisy.


(20) McDonald logo.

(21) Watermelon.


(22) Parthenon.


(23) Butterfly.


(24) Rose.


(25) Soccer ball.


(26) Clock.


(27) Strawberry.


(28) Sunflower.


(29) Sushi roll.


(30) Telephone booth.

Figure 3.4: Topic image examples.

**Systems** For the experiment it is possible to differentiate five variants of the system, which will be labeled as *S1* to *S5*. Each system uses as input a keyword-based query, and return a ranked list of image results.

S1: *Text-based retrieval.* The textual baseline for the experiment is based on the vector space model for text retrieval. Using the textual annotations (tags) of the images, the related images are retrieved for a given keyword-based query by measuring the cosine-similarity between the query and the image annotations.

S2: *Content-based image retrieval using visual annotations.* This system uses the keyword-based query to select (at random) one of the ten visual annotations that matches the query. Based on the extracted visual features that are within the bounding box of the visual annotation, the related images are retrieved, as described in detail in Section 3.1. Since the visual annotations are selected at random for each topic, it was constructed 25 random runs. For each of these runs, the average performance over the 25 repeated measurements in the results section was reported.

S3: *Aggregated ranking over the results of content-based image retrieval using visual annotations.* This system searches using all 10 visual annotations and apply rank aggregation over the 10 partial result lists that are computed for each topic, as discussed in more detail in Section 3.1. The top 25 results of the 10 partial rankings is used as input for the aggregation step.

S4: *Content-based image retrieval using visual annotations and a tag filter.* The approach of this system is similar to system S2, with an additional filter over the image annotations, which requires that the tags matches with all the query terms.

S5: *Aggregated ranking over the results of content-based image retrieval using visual annotations and a tag filter.* The approach of this system is similar to system S3, with an additional filter over the image annotations, which requires that the tags matches with all the query terms.

By comparing the results from these systems, it is possible to test each of the hypotheses presented on Section 3.1. Comparing system S2 versus S3 (or S4 versus S5) allows for testing hypothesis H1, which states that the

retrieval performance benefits from the rank aggregation over the partial results obtained by the visual annotations. Likewise, the comparison of S1 with S4 and S5 allows us to test hypothesis H2, which is focused on improving the retrieval performance by combining visual and textual search.

**Pooling and Assessments** To evaluate the retrieval performance of the systems, a blind review pooling method was implemented, as is commonly used in TREC [51]. The topic pools are based on the top 25 results for each topic retrieved by each of the systems. This typically represents the number of images shown on a result page. Systems S2-5 were pooled by selecting the top 25 results for each visual annotation, and then include a separate run for each of the three features (Harris, Hessian, and combined). The assessors were asked to judge the relevance of the results for a given topic on a binary scale, and they were instructed to take the information provided by the topic description into account. The assessment interface provided the assessor with the image, title, tags and description.

**Evaluation Measures** In this experiment the main focus was on achieving a high precision at the top of the ranking and not so much on recall. Therefore, the results section focuses on P@N, with N ranging from 1 to 25, which allows to investigate the quality of the ranking at early cut-off. An assumption is that giving the reader an idea of P@1-5 of the method is relevant. The reason is that not always in Web search engines the user is presented with a block of images, since it depends on the type of search used (e.g. Flickr displays the results vertically, Yahoo! image search displays them block-like, and Google universal search combines text and image results). For this reason, giving information about P@1-5 provides more detailed insight in the behavior of the model. Furthermore, it is reported the mean average precision (MAP) and binary preference (BPREF), which is claimed to be more stable for incomplete judgements [6].

### Results

**Feature Selection** Before addressing the main research questions, the retrieval performance was analyzed when varying the feature selection. In Section 3.1 two features were identified, Harris affine (HAR) and Hessian affine (HES), and a linear combination (COM) of the two features as the feature space. The feature selection affects all systems that use the visual

search (S2-5). Table 3.2 presents the performance of each of the four systems with the different features.

| Systems | S2 | | | S3 | | | S4 | | | S5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Feature | COM | HAR | HES | COM | HAR | HES | COM | HAR | HES | COM | HAR | HES |
| MAP | **0.05** | 0.04 | 0.04 | **0.11** | 0.10 | 0.10 | **0.20** | 0.18 | 0.19 | **0.24** | 0.23 | **0.24** |
| BPREF | **0.07** | 0.05 | 0.05 | **0.14** | 0.12 | 0.13 | **0.26** | 0.24 | 0.25 | **0.30** | 0.29 | 0.29 |
| P@10 | **0.31** | 0.26 | 0.27 | 0.48 | **0.49** | 0.48 | 0.71 | 0.69 | **0.72** | **0.80** | 0.77 | 0.79 |

Table 3.2: Retrieval performance for different features.

The values in bold indicate the best performing variant per system for each of the three measures (MAP, BPREF, and P@10). Though the differences are not significant, the combined (COM) approach, where the two feature spaces are concatenated, clearly is the preferred method according to all the measures for each system. Therefore, the discussion of the results, will limit to the combined variant.

**Summary Statistics** Table 3.3 presents the summary statistics of the retrieval performance experiment for the five systems. Each of the systems returned the top 25 results for the 30 topics, except system S4 and S5, where the filtering had a small impact on the number of results retrieved. Based on all four measures presented in the table (i.e. MAP, BPREF, P@5, and P@10 respectively), it can be concluded that S5, the system that is based on aggregated ranking over the results of content-based image retrieval with a tag filter, clearly outperforms the other systems.

| System | S1 | S2 | S3 | S4 | S5 |
|---|---|---|---|---|---|
| Number of Topics | 30 | 30 | 30 | 30 | 30 |
| Images Retrieved | 750 | 750 | 750 | 742 | 748 |
| Relevant | 2187 | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 393 | 149 | 301 | 494 | 562 |
| MAP | 0.12 | 0.05 | 0.12 | 0.20 | **0.24** |
| BPREF | 0.20 | 0.07 | 0.15 | 0.26 | **0.30** |
| P@5 | 0.53 | 0.34 | 0.55 | 0.72 | **0.82** |
| P@10 | 0.49 | 0.31 | 0.48 | 0.71 | **0.80** |

Table 3.3: Summary statistics.

### Precision at Early Cut-off

Figure 3.5 plots the graphs for precision at various cut-off points (P@N). The graphs allow for a more detailed analysis of the systems and their ability to rank relevant results near the top of the ranking. For S1, the tag-only run, the performance slightly decays from 0.57 to 0.49. As expected, the performance for S2, the system that uses content-based image retrieval with visual annotations, is lower than for S1 and ranges from 0.36 to 0.20. The results for system S3 show that the retrieval can be significantly improved by performing rank aggregation of the results obtained for S2. With the precision ranging from 0.63 to 0.40, the precision is almost twice as high. In fact, the relevancy of the top 5 results is even higher than for the tag-only run.

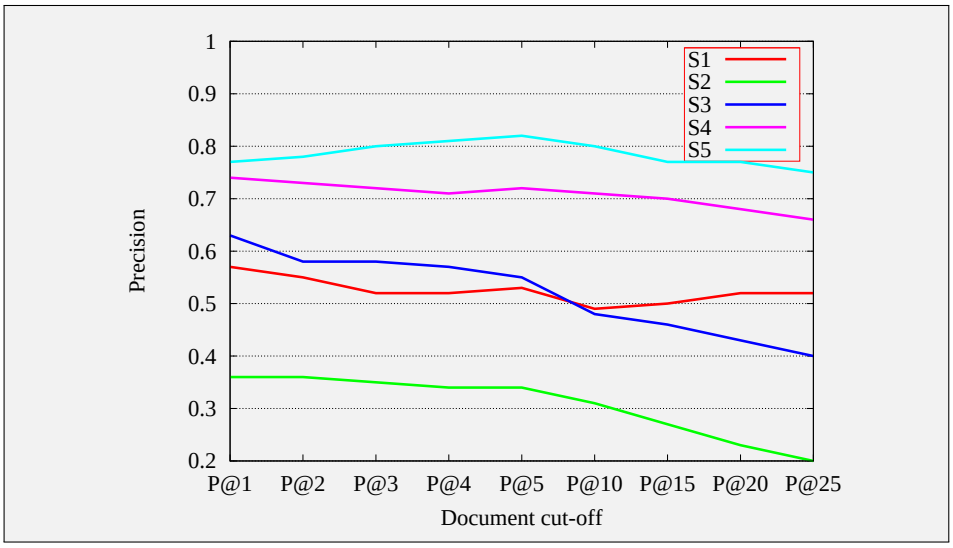



Figure 3.5: Precision at early cut off; systems overview.

The system variants S4 and S5 combine visual search with a textual filter. As shown in the figure, this leads to another significant increase in retrieval performance over the tag-only system S1 and the systems S2 and S3 that only use the visual features. The results show that precision over the top 25 ranges from 0.74 to 0.66 for S4, and that for S5 the precision is always higher than 0.75.

Therefore, it is concluded that in all cases rank aggregation over the result sets for content-based image retrieval with visual annotations leads to

a significant increase in retrieval performance as posed in hypothesis H1. Furthermore, the combined visual and textual approach shows significant improvements over the tag-only system, therefore, hypothesis H2 can be validated.

### Topic Analysis

The final part of the evaluation, presents a topic analysis to detect whether the observations of the previous two sections are caused by abnormalities in the performance for a subset of the topics. Figure 3.6 provides a topic histogram for the P@10. On the x-axis the P@10 ([0.0 -10.0]) is projected, while the y-axis projects the number of topics with the same P@10 rounded to one decimal precision.

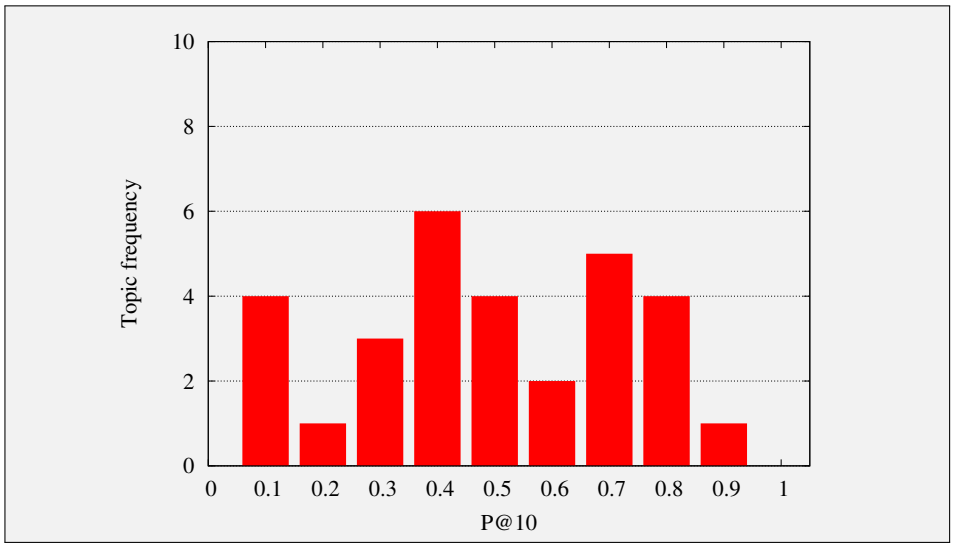



Figure 3.6: P@10: Precision after having seen the first ten results for system S1.

For system S1 the average P@10 is 0.49, with a standard deviation of 0.24, while the average P@10 for system S5 is 0.8 with a standard deviation of 0.19. This indicates that there is a significant and uniform increase in retrieval performance for all topics.

Finally, Figure 3.8 plots the MAP in a histogram for each individual topic per system, allowing a per-topic comparison. Based on the results, it is

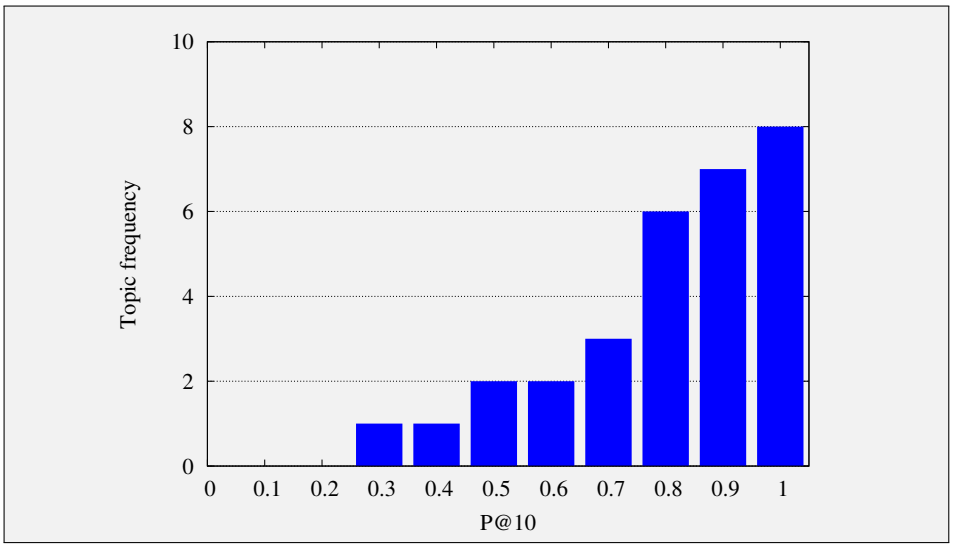



Figure 3.7: P@10: Precision after having seen the first ten results for system S5.

possible to extract several observations. First of all, it reveals that S5 and S4 are consistently better than S1. However, the performance on a number of topics is weaker when no textual information is present to limit the search space. For example, this can be observed on the performance of the topics: "butterfly", and "watermelon" with systems S2 and S3. One explanation is that those images (or visual annotations) contain many small non-characteristic visual words, which can easily be mistaken.

A good example of the importance of combining visual and textual features can be observed in topic "butterfly" (see Figure 3.9). Using visual features gives poor results, which can be due to multiple salient regions in the images, and the diversity of different butterflies. When combining visual and textual information is possible to distinguish the images that actually represent a butterfly, obtaining higher MAP.

### Summary

This section studied the problem of keyword based image retrieval on a diverse image collection, such as typically found in on-line photo sharing services. The available human annotations allow for existing text retrieval

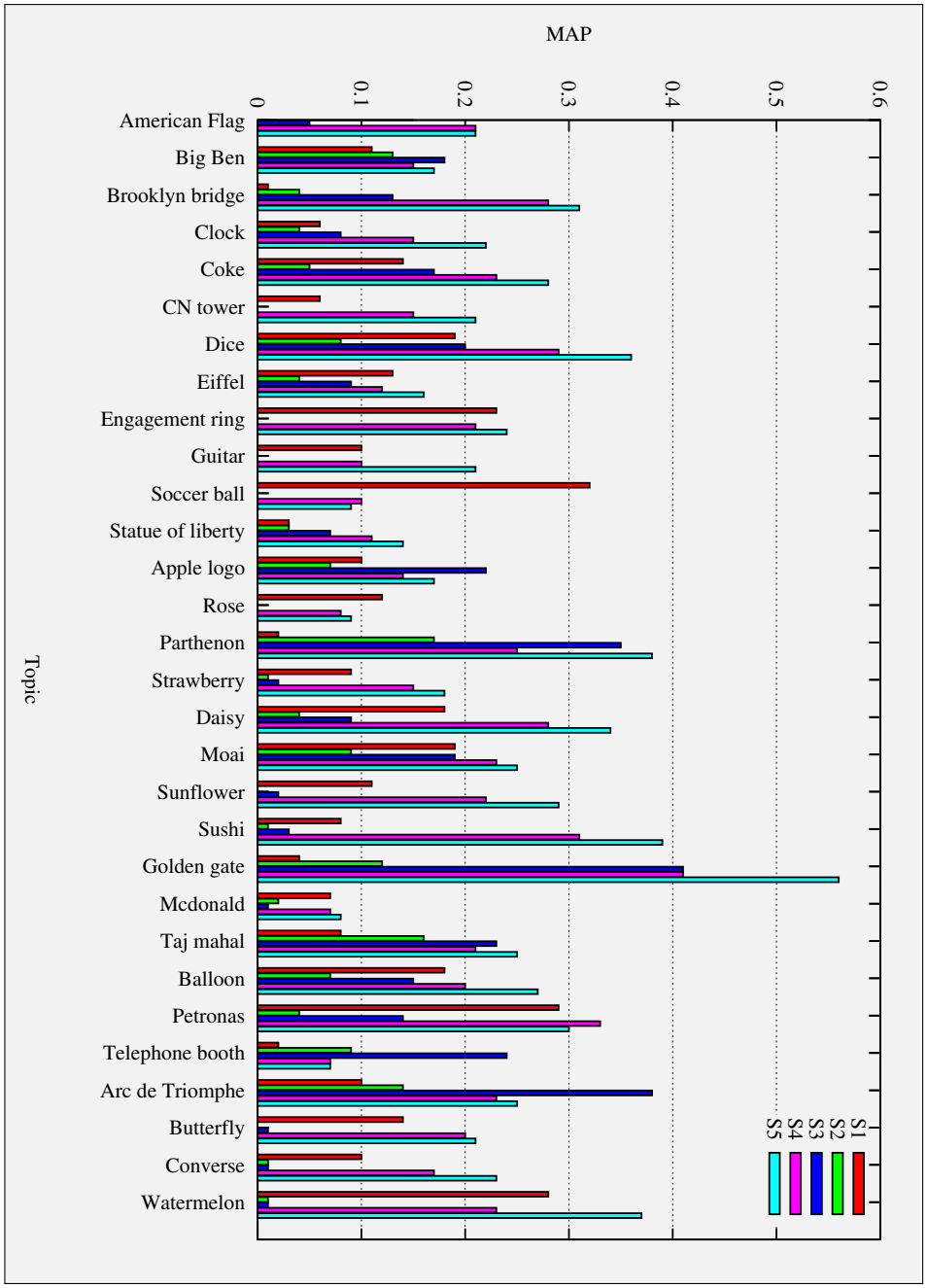



Figure 3.8: MAP histogram, per topic.

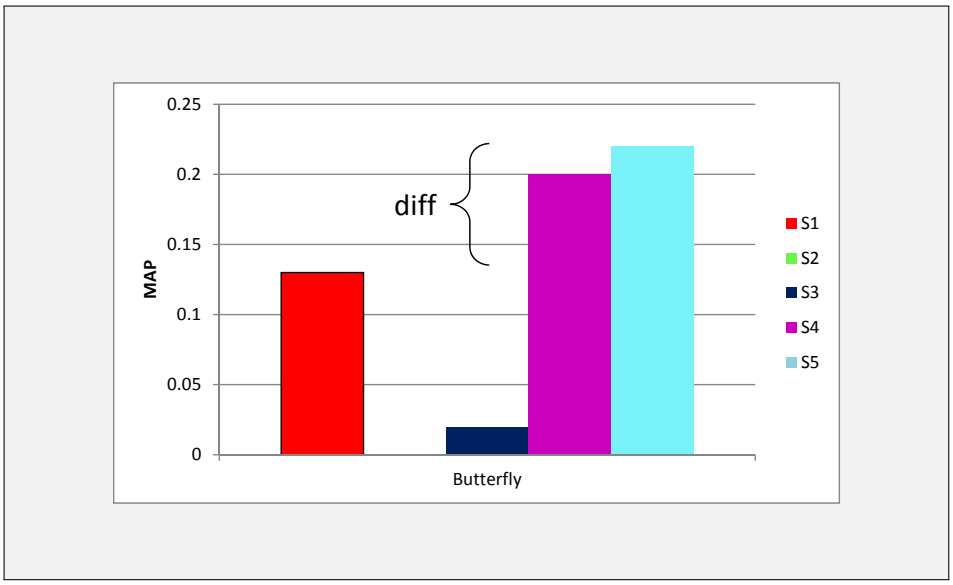



Figure 3.9: MAP of the different systems, for the topic "Butterfly". It can be observed how only using visual features (S2 and S3) give poor performance, but combining with text gives a boost in precision ("diff").

models to work on such large corpora, but due to the sparsity of the information provided with the photos these models are not optimal.

Central in the research was the question: "How can we deploy the visual annotations, also known as *notes* in Flickr, to enhance the retrieval performance?". In more detail, the use of rank aggregation was proposed to combine the result sets of a content-based image retrieval system that uses the visual annotations to retrieve similar images. The results of the retrieval performance experiment clearly showed that the quality of the results significantly improves when applying the rank aggregation on the results obtained with the content-based image retrieval system. Moreover, the results of the aggregated visual search show a marginal improvement when compared with the tags-only run.

When extending the visual search with a textual filter on the tags it is possible to further limit our search space, and show another significant boost in retrieval performance in terms of precision.

## 3.2 Aggregation

Section 3.1 presents a methodology to make use of user generated content on an image retrieval system to improve retrieval performance. This is achieved by using visual annotations to select a list of images that visually match the query and then aggregate these partial lists to obtain a final ranked list.

This problem is known as the *rank aggregation problem* [17] where individually ranked lists from multiple sources need to be combined into a final rank to obtain a consensus ordering.

In this section different aggregation methods for image object retrieval will be analyzed. Aggregation can be seen as a pre-retrieval or post-retrieval process, were the difference lies in when the visual words of the annotations are taken in consideration. Post-retrieval refers to using the visual annotations to retrieve a set of partial lists, and after the retrieval step, find a suitable form of aggregating the partial results, which is known as rank aggregation. Pre-retrieval uses the fact that each annotation represents a set of visual words, so the challenge is to find a proper way to combine the different visual words of the annotations and obtain a single representation that can be used to retrieve the images. This procedure can be seen as building visual concept models for the different topics.

### Rank Aggregation

Rank aggregation can be found in many situations. It was first studied in social choice and voting theory applied to political elections [45]. Now it has also been applied to document filtering, document classification and clustering, web spam detection, meta-search, and similarity search.

The rank aggregation problem can be modeled as a voting mechanism, were the candidates correspond to the collection of images to be sorted and the voters correspond to each visual annotation. Each voter will generate a ranked list of the *top-k* images and the main goal is to find a final ranking that combines the *top-k* lists in an efficient and consensus way.

There are different methods to combine the ordered lists which can be classified into two groups: (a) using the ordinal rank assigned to a candidate in the ordered list or (b) using the score assigned to each candidate.

### Borda Count

The Borda count method is a positional method that uses the order of the candidates to assign points to them, and a final list of ranking candidates must be obtained.

For this, every voter assigns points to each of the candidates, based on the position in their ranked list. The first element in the list is assigned with $k$ points, the second element is given $(k-1)$ points, until the last element is assigned 1 point. To obtain the final list of results, all the candidates are sorted by their total number of points.

Given a universe $U$ of candidates, in this case the complete collections of images numbered as $i = 1, 2, \ldots, n$, the different voters submit the partial top-k lists $t_1, t_2, \ldots, t_\ell$ where $\ell$ is the number of voters and $|t_i| \leq |U|$.

Each candidate $c \in U$ and each element on the list $t_i$ is assigned a score.

$$B_i(c) = \#\text{ of candidates ranked below c} \in t_i$$

And the total Borda count score:

$$B(c) = \sum_{i=1}^{n} B_i(c)$$

### Markov Chains

The use of Markov chains was first introduced by Dwork *et al.* [17] they propose the use of Markov chains to obtain an aggregated list of results.

As also stated in their work, the simplifications in time to obtain the stationary probability distribution will be used. They proposed four different Markov chains.

General method to obtain an initial aggregation from the partial lists using Markov chains.

The states in the chain will be the $n$ candidates, the transition probabilities will depend on the partial lists. This approach allows to handle partial lists. Because its compares all the $n$ candidates between them.

Four Markov chains are proposed in [17]. For each Markov Chain a transition matrix must be built.

***MC*$_1$**: If the current state is the candidate P then the following state will be chosen uniformly from the set of all candidates that are ranked higher or equal than P in any list that also contains P.

The transition probability for $MC_1$ can be written as:

$$P = diag(\frac{1}{\sum_{j=1}^{n} q_{1j}}, \ldots, \frac{1}{\sum_{j=1}^{n} q_{nj}})Q \tag{3.1}$$

Where:

$$Q = (q_{ij})_{n\times n} = \frac{1}{\ell}\sum_{k=1}^{l} Q_k$$

$$Q_k = (q_{ij}^{(k)})_{n\times n}$$

$$q_{ij}^{(k)} = \begin{cases} 1 & \text{if } j >_{t_k} i \text{ or } j = i \\ 0 & \text{otherwise} \end{cases}$$

Where $j >_{t_k} i$ indicates that the element $j$ is located below the element $i$ in the list $t_k$

***MC*$_2$**: If the current state is P, the following state Q will be chosen by selecting uniformly a list $t$ from all partial lists $t_1, t_2, \ldots, t_\ell$ than contains P. Then, the candidate Q is selected randomly from the set of candidates ranked higher or equal than P.

$$P = \frac{1}{\ell}\sum_{k=1}^{\ell} P_k$$

$$p_{ij}^{(k)} = \begin{cases} \frac{1}{m} & \text{if } j >_{t_k} i \text{ or } j = i \\ 0 & \text{otherwise} \end{cases}$$

***MC*$_3$**: If the current state is P, the next state will be picked by uniformly selecting a list $t$ from the set of lists $t_1, t_2, \ldots, t_\ell$ that contains P. Then

uniformly choose a candidate Q that is contained in $t$. If $t(Q) < t(P)$ the next state will be Q else stay in P.

$$p_{ij}^{(k)} = \begin{cases} \frac{1}{n} & \text{if } j >_{t_k} i \\ \frac{n-m}{n} & \text{if } j = i \text{ and } m = \#\{j | j >_{t_k} i\} \\ 0 & \text{otherwise} \end{cases}$$

***MC*$_4$**: If the current state is P, the next state will be picked by uniformly select a candidate Q from the union of all candidates from all the lists. If $t(Q) < t(P)$ for the majority of the lists that contains P and Q, then the next stage will be Q else stay in P.

$$Q = (q_{ij})_{n \times n} = \frac{1}{\ell} \sum_{k=1}^{\ell} Q_k$$

with:

$$q_{ij}^{(k)} = \begin{cases} 1 & \text{if } j >_{t_k} i \\ 0 & \text{otherwise} \end{cases}$$

and:

$$p_{ij}^{(k)} = \begin{cases} \frac{1}{n} & \text{if } q_{ij} > \frac{1}{2} \\ \frac{n-m}{n} & \text{if } j = i \text{ and } m = \#\{j | q_{ij} > \frac{1}{2}\} \\ 0 & \text{otherwise} \end{cases}$$

### Comparison

To compare the retrieval performance of each of the aggregation methods presented, an experiment was conducted using the results obtained in Section 3. Since system S5, which combines visual information with the textual tag information, outperformed the other retrieval system, it was used as input for the partial lists to be aggregated. Then, all set of lists were combined using the aggregation methods described in this section.

Table 3.4 presents a summary of the results obtained using each of the aggregation algorithms. Figure 3.10 shows a comparison of the precision at different document cut-off.

By analyzing the results obtained, it can be observed that Borda aggregation outperforms the results obtained with Markov chains. These results differ from the results presented by Dwork *et al.* [17], but there are some considerations that explain the differences in the problem definition.

| Measure | Aggregation | | | | |
|---|---|---|---|---|---|
| | MC1 | MC2 | MC3 | MC4 | Borda |
| Number of Topics | 30 | 30 | 30 | 30 | 30 |
| Images Retrieved | 748 | 748 | 748 | 748 | 748 |
| Relevant Images | 2187 | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 515 | 515 | 492 | 472 | 563 |
| MAP | 0.2105 | 0.2088 | 0.1975 | 0.1785 | **0.2439** |
| P@1 | 0.6667 | 0.6667 | 0.6000 | 0.6000 | **0.7667** |
| P@2 | 0.7000 | 0.6833 | 0.6667 | 0.6000 | **0.7833** |
| P@3 | 0.7111 | 0.7000 | 0.6889 | 0.6000 | **0.8000** |
| P@4 | 0.7417 | 0.6917 | 0.6917 | 0.6333 | **0.8083** |
| P@5 | 0.7267 | 0.7000 | 0.6867 | 0.6400 | **0.8267** |
| P@10 | 0.7267 | 0.7033 | 0.6900 | 0.6300 | **0.8033** |
| P@15 | 0.7089 | 0.6956 | 0.6711 | 0.6556 | **0.7800** |
| P@20 | 0.6983 | 0.6867 | 0.6667 | 0.6383 | **0.7750** |
| P@25 | 0.6867 | 0.6400 | 0.6560 | 0.6293 | **0.7507** |
| Avg. Induced Footrule | 106.39 | 106.69 | **98.87** | 117.56 | 114.92 |

Table 3.4: Summary statistics.

First of all, Dwork *et al.* [17] work uses partial lists obtained from different sources, which most probably have different indexes, that might generate disjoint result lists. In the present work, all the partial lists are obtained from the same collection, hence it is unlikely to have disjoint result lists. Another consideration is the metric used to determine the "quality" of the aggregated result list. Dwork *et al.* [17] work is based on Footrule measure, which represents the distance between two lists. In the present work is more important to compare using the precision at document cut-off.

These two issues are tightly related. Since in Dwork *et al.* [17] work the partial lists are obtained from different sources, and there is no concept of assessing the quality of each source, each list is considered equally important. Hence, the "ideal" aggregated result list should combine the lists, by minimizing the distance to each of the partial list. For this reason, using the Footrule distance to compute the "quality" of the aggregated list, is a good solution.

On the other hand, in the present work all the partial lists are obtained from the same collection and it exists a relevance assessment of the collection, so the measure of the "quality" of the aggregated list should consider preci-

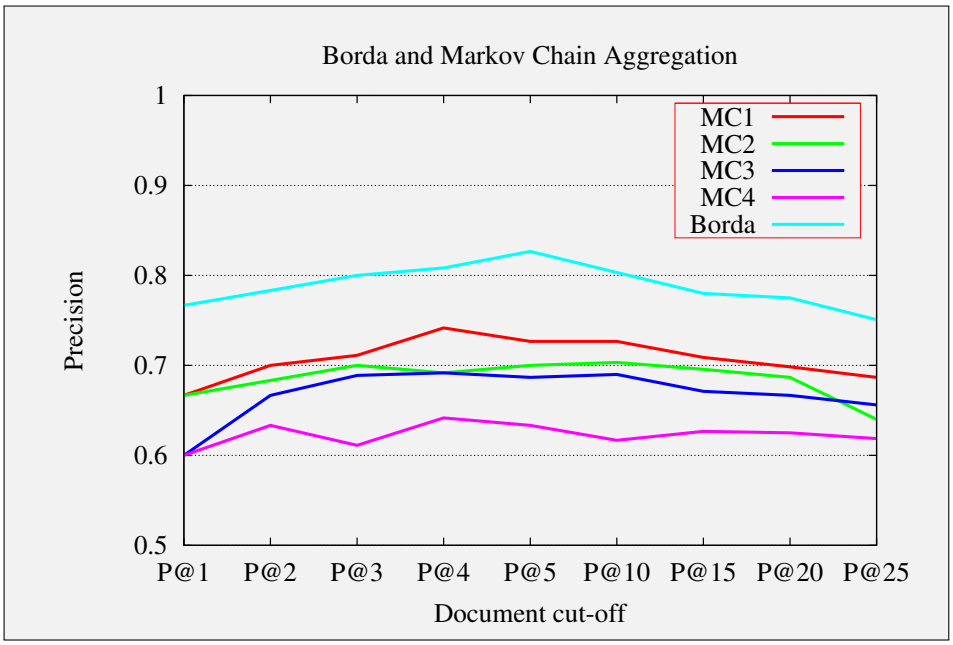



Figure 3.10: Precision at early cut off for Borda and Markov Chain Aggregation.

sion at early document cut-off. This measure can lead to discarding some, or most, of the results of some of the partial lists without degrading, and actually improving, the retrieval performance of the aggregation algorithm.

### Visual Concept Models

For every topic there is a set of visual annotations used to describe the visual characteristics of the topic. Furthermore, when using the *bag-of-visual-words* approach, each annotation can be represented as a weighted set of *visual words*. By using the analogy with text retrieval, each annotation is equivalent to a "query", and each visual-word corresponds to a "query term".

Instead of using each "query" to obtain a ranked partial list and then merging these lists into the final result (as seen on previous section), the idea is to create a visual concept model that will be used to query once the document collection. This approach allows to improve the performance, in terms of speed, by reducing the number of queries required to build the final result.

The problem that arises is how to create this artificial query, based on the original set of queries, in such a way that retrieval performance, in terms of precision, leverages the performance, in terms of speed.

Using the distribution of the visual words, this section presents three different approaches for the pre-aggregation step: Sum, Max, and Min.

**Sum** The idea is to create a synthetic image by aggregating the visual words of the sample images, were the weight of each resulting visual word is the sum of the weights from all the sample images. A description of the algorithm is presented in Algorithm 3.11. Finally, the average precision at different document cut-off is compared.

```
1: I ← {set of images}
2: T ← {set of topics}
3: for t ∈ T do
4:     v_t ← 0
5:
6:     S_t ← {set of sample images for topic t}
7:     for s_i ∈ S_t do
8:         C_i ← {set of classes (with weights) of image s_i}
9:         for [c, w] ∈ C_i do
10:            // c ≡ class name
11:            // w ≡ weight of class
12:            v_t[c] ← v_t[c] + w
13:
14:    v̄_t ← v_t / |v_t|
15:    σ_t ← rank(I, v̄_t)
16:
17: Σ ← {list of rankings} ≡ {σ_0, σ_1, ..., σ_n}
18: trec_eval(Σ)
```

Figure 3.11: Algorithm: Pre-aggregation using sum.

**Max** The steps are similar to the "sum", with the only difference being how to build the set of visual words. In this case, when iterating over the classes of the image, the algorithm also considers the number of visual words in each image. Then, for each class, it picks the weight from the image that had

more visual words of that class (see Algorithm 3.12). Finally, the average precision at different document cut-off is compared.

```
1: I ← {set of images}
2: T ← {set of topics}
3: for t ∈ T do
4:     v⃗_t ← 0⃗
5:     n⃗_t ← 0⃗
6:
7:     S_t ← {set of sample images for topic t}
8:     for s_i ∈ S_t do
9:         C_i ← {set of classes (with weight and # of visual words) of
   image s_i}
10:        for [c, w, m ≡ {number of visual words of class c}] ∈ C_i do
11:            // c ≡ class name
12:            // w ≡ weight of class
13:            // m ≡ number of visual words
14:            if m > n⃗_t[c] then
15:                n⃗_t[c] ← m
16:                v⃗_t[c] ← w
17:
18:    v̄_t ← v⃗_t / |v⃗_t|
19:    σ_t ← rank(I, v̄_t)
20:
21: Σ ← {list of rankings} ≡ {σ_0, σ_1, ..., σ_n}
22: trec_eval(Σ)
```

Figure 3.12: Algorithm: Pre-aggregation using max.

**Min** The steps are similar to the "max", except that now the algorithm picks the weight from the image that had less occurrences of that class (see Algorithm 3.13). Finally, the average precision at different document cut-off is compared.

### Summary

This section covered different forms of aggregating (rank aggregation and post-retrieval aggregation) the visual annotations for a given topic, to ob-

```
1: I ← {set of images}
2: T ← {set of topics}
3: for t ∈ T do
4:     v_t ← 0⃗
5:     n_t ← 0⃗
6:
7:     S_t ← {set of sample images for topic t}
8:     for s_i ∈ S_t do
9:         C_i ← {set of classes (with weight and # of visual words) of
   image s_i}
10:        for [c, w, m ≡ {number of visual words of class c}] ∈ C_i do
11:            // c ≡ class name
12:            // w ≡ weight of class
13:            // m ≡ number of visual words
14:            if m <= n_t[c] or n_t = 0 then
15:                n_t[c] ← m
16:                v_t[c] ← w
17:
18:    v̄_t ← v_t / |v_t|
19:    σ_t ← rank(I, v̄_t)
20:
21: Σ ← {list of rankings} ≡ {σ_0, σ_1, ..., σ_n}
22: trec_eval(Σ)
```

Figure 3.13: Algorithm: Pre-aggregation using min.

| Aggregation | Sum | Max | Min | S5 |
|---|---|---|---|---|
| Number of Topics | 30 | 30 | 30 | 30 |
| Images Retrieved | 748 | 748 | 748 | 748 |
| Relevant | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 562 | 552 | 508 | 562 |
| MAP | 0.2474 | 0.2397 | 0.2156 | 0.2400 |
| P@5 | 0.8200 | 0.8000 | 0.8067 | 0.8200 |
| P@10 | 0.8233 | 0.7900 | 0.7533 | 0.8033 |

Table 3.5: Summary statistics, comparing the different pre-aggregation strategies with S5.

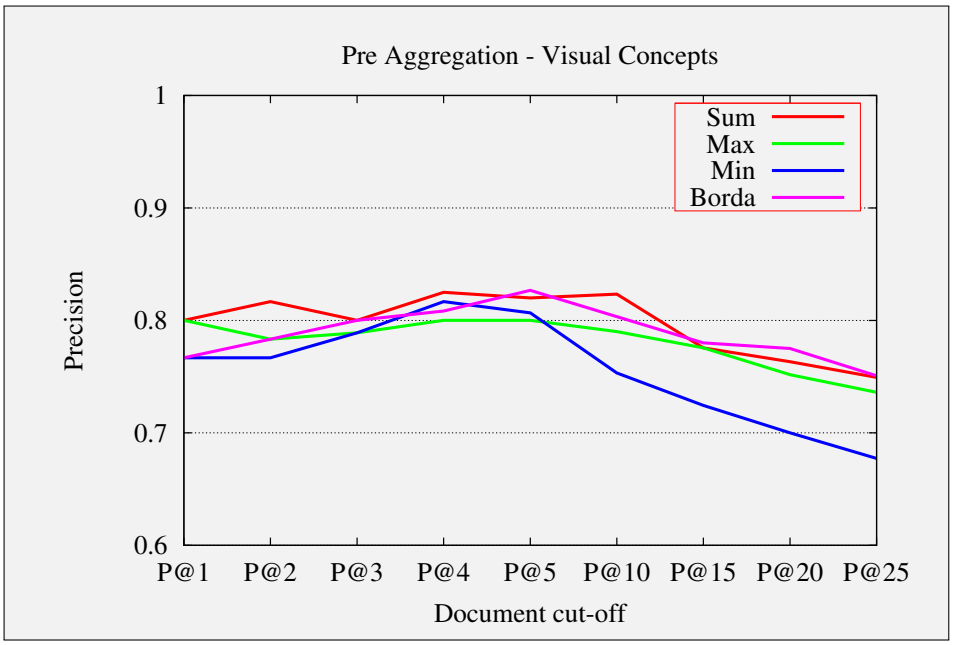



Figure 3.14: Precision at early cut off; Pre-aggregation combined.

tain a ranked list of results. Rank aggregation is a well known methodology that is being used in Web meta-searching to obtain a ranked list of results, from multiple sources. Compared the retrieval performance of using Markov chains, introduced by Dwork *et al.* [17], versus the previous results obtained using Borda count. In these experiments, Borda outperforms Markov chains, which contradicts Dwork *et al.* [17] results, but considering the differences between the collections used, the results are consistent. Finally, by exploiting the fact that annotations are a set of visual words, a new query has been formulated by pre-aggregating them. Three different mechanisms to pre-aggregate the results were compared with the ones obtained by combining visual and textual features (system S5), and the performance of doing this pre-retrieval aggregation is promising.

## 3.3 Annotation Selection

This section will propose and experiment with different annotation selection approaches to determine the best way to select the visual annotations that

help improve the retrieval of images. It will be presented and analyzed four approaches for filtering the results:

1. Compare every partial result list obtained from each sample annotation with the aggregated list, and compute the Induced Footrule distance between them.

2. Compute Mean Reciprocal Rank of every partial result. The main idea is that voters that agree with the rest of the voters will be given a bigger weight, while voters that tend to disagree with the rest will be given a smaller weight.

3. Instead of using the ranking position of the result images, use the similarity score between the images in the result list.

4. Optimal number of annotations used.

### Induced Spearman Footrules

The *Spearman Footrule distance* permits to compute the similarity of two ranked lists, based on comparing the position of the elements in each list. Given two ranked lists, $\sigma_1$ and $\sigma_2$, the Footrule distance $F(\sigma_1, \sigma_2)$ can be defined as:

$$F(\sigma_1, \sigma_2) = \sum_{i=0}^{n} |\sigma_1(i) - \sigma_2(i)|$$

where, $\sigma(i)$ is the rank position of image $i$.

Furthermore, this distance can be extended to compute the similarity between a top-k ranked list, and the top-k aggregated list of results, as presented in [17]. They introduce the *induced Spearman Footrule* to compute the similarity, by first projecting the results list to only contain the elements in the aggregated list. Hence, given a ranked list $\sigma$ and the aggregated list $\sigma_{agg}$, the *Induced Spearman Footrule*, can be defined as:

$$F_{ind}(\sigma, \sigma_{agg}) = \sum_{i=0}^{n} |\tilde{\sigma}(i) - \sigma_{agg}(i)|$$

where $\tilde{\sigma}$ is a sublist of $\sigma$ containing only the elements in $\sigma_{agg}$, and preserving the original order.

By using only a projection of the list over the aggregated list, it allows to favor the similar elements in the top of the rank, hence allowing to filter out the lists that differ most from the aggregated one. This process of filtering is described in Algorithm 3.15.

$I \leftarrow$ {set of images}
$T \leftarrow$ {set of topics}
**for** $t \in T$ **do**
    $S_t \leftarrow$ {set of sample images for topic $t$}
    **for** $s_i \in S_t$ **do**
        $\sigma_i \leftarrow$ rank(I, $s_i$)
    $\Sigma \leftarrow$ {list of rankings} $\equiv \{\sigma_0, \sigma_1, \ldots, \sigma_n\}$

    $\sigma_{agg} \leftarrow$ borda_aggregation($\Sigma$)

    **for** $\sigma_i \in \Sigma$ **do**
        $\tilde{\sigma}_i \leftarrow$ projection($\sigma_i$, $\sigma_{agg}$)
        $\tilde{F}_i \leftarrow$ spearman($\tilde{\sigma}_i$, $\sigma_{agg}$) $\equiv \sum_{j=0}^{k} |\tilde{\sigma}_i(j) - \sigma_{agg}(j)|$

    $\tilde{F} \leftarrow$ {list of induced footrules} $\equiv \{\tilde{F}_0, \tilde{F}_1, \ldots, \tilde{F}_n\}$
    **for** $\tilde{F}_i \in \tilde{F}$ **do**
        $\delta_i \leftarrow$ percentage_difference($\tilde{F}_i$, $\tilde{F}$) $\equiv \frac{\tilde{F}_i - min(\tilde{F})}{min(\tilde{F})}$

    $\Upsilon \leftarrow$ {set of thresholds} $\equiv \{0.10, 0.25, 0.50, \ldots\}$
    **for** $\epsilon_i \in \Upsilon$ **do**
        $\Sigma_\epsilon \leftarrow$ filter($\Sigma$, $\delta_i$, $\epsilon_i$)
        compute_performance($\Sigma_\epsilon$)

Figure 3.15: Algorithm: Calculate Induced Spearman Footrule.

To analyze the retrieval performance of doing annotation selection using induced Footrules, the rank algorithm S5 is used as presented in Section 3.1, since it outperformed the other rank algorithms. Figure 3.16 presents the resulting precision at early cutoff by comparing different thresholds used to select the annotations being used. Table 3.6 shows the detailed data for different thresholds.

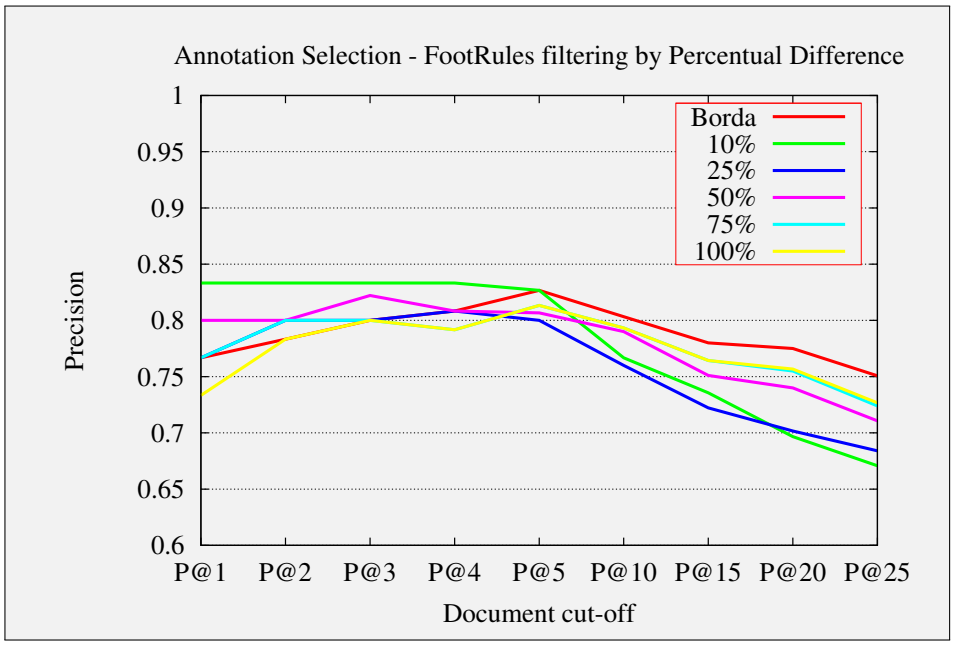



Figure 3.16: Precision at early cut off; Annotation selection using induced Footrules.

From the data presented in Table 3.6, one can observe that for early document cutoffs (e.g., P@1 to P@5) the best retrieval performance can be obtained by using only the sample annotations whose percentual difference is small (less than 10%).

### Reciprocal Rank

Reciprocal rank of a ranked list is the inverse of the first occurrence of a "correct" answer. In the context of image retrieval using visual annotation samples, a "correct" answer can be considered the occurrence of one of the

| Measure | Induced Footrule filtering | | | | | |
|---|---|---|---|---|---|---|
| | 10% | 25% | 50% | 75% | 100% | Borda |
| Number of Topics | 30 | 30 | 30 | 30 | 30 | 30 |
| Images Retrieved | 722 | 722 | 722 | 730 | 732 | 748 |
| Relevant Images | 2187 | 2187 | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 503 | 513 | 533 | 543 | 545 | 563 |
| MAP | 0.2078 | 0.2112 | 0.2244 | 0.2278 | 0.2278 | **0.2439** |
| P@1 | **0.8333** | 0.7667 | 0.8000 | 0.7667 | 0.7333 | 0.7667 |
| P@2 | **0.8333** | 0.8000 | 0.8000 | 0.8000 | 0.7833 | 0.7833 |
| P@3 | **0.8333** | 0.8000 | 0.8222 | 0.8000 | 0.8000 | 0.8000 |
| P@4 | **0.8333** | 0.8083 | 0.8083 | 0.7917 | 0.7917 | 0.8083 |
| P@5 | **0.8267** | 0.8000 | 0.8067 | 0.8133 | 0.8133 | **0.8267** |
| P@10 | 0.7667 | 0.7600 | 0.7900 | 0.7933 | 0.7933 | **0.8033** |
| P@15 | 0.7356 | 0.7222 | 0.7511 | 0.7644 | 0.7644 | **0.7800** |
| P@20 | 0.6967 | 0.7017 | 0.7400 | 0.7550 | 0.7567 | **0.7750** |
| P@25 | 0.6707 | 0.6840 | 0.7107 | 0.7240 | 0.7267 | **0.7507** |

Table 3.6: Summary statistics, Induced Footrule filtering.

other sample images. So, given a set $S$ of sample images, and a ranked list $\sigma$, the reciprocal rank can be defined as:

$$\rho(\sigma, S) = \begin{cases} \frac{1}{\text{minindex}(\sigma,\ S)} & \text{if } \sigma \text{ contains an element of } S \\ 0 & \text{otherwise} \end{cases}$$

where $minindex(\sigma, S)$ returns the position (index) of the first occurrence of an element of $S$ in the list $\sigma$.

By assigning the reciprocal rank as a quality score to each ranked list, is possible to exclude the annotations that returned lists with low score, since this can indicate that the list is not representative of the other samples. Algorithm 3.17 describes the procedure to test different combinations of partial lists.

Similar to the experiment using the induced Footrule distance, to analyze the retrieval performance of this algorithm, the results of rank algorithm S5 have been used as presented on Section 3.1. The results are presented on Figure 3.18 and the Table 3.7.

From these results, is possible to observe that filtering at higher values of $\rho$ ($\rho \geq \frac{1}{10}$ ) produces better results than using Borda aggregation.

1: $I \leftarrow$ {set of images}
2: $T \leftarrow$ {set of topics}
3: **for** $t \in T$ **do**
4: $S_t \leftarrow$ {set of sample images for topic $t$}
5: **for** $s_i \in S_t$ **do**
6: $\sigma_i \leftarrow$ rank(I, $s_i$)
7: $\rho_i \leftarrow$ reciprocal_rank($\sigma_i$, $S_t$) $\equiv \frac{1}{\text{min\_index}(\sigma_i, S_t)}$
8:
9: $\Sigma \leftarrow$ {list of rankings} $\equiv \{\sigma_0, \sigma_1, \ldots, \sigma_n\}$
10:
11: $\Upsilon \leftarrow$ {set of thresholds} $\equiv \{1, 1/2, 1/5, \ldots\}$
12: **for** $\epsilon_i \in \Upsilon$ **do**
13: $\Sigma_\epsilon \leftarrow$ filter($\Sigma$, $\rho_i$, $\epsilon_i$)
14: compute_performance($\Sigma_\epsilon$)

Figure 3.17: Algorithm: Compute Reciprocal Rank.

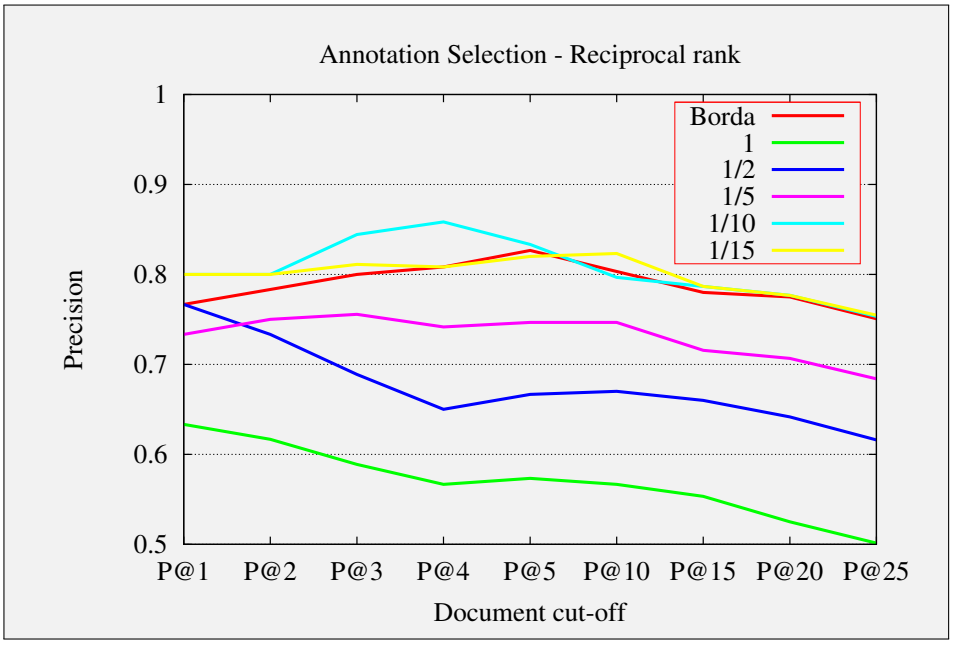



Figure 3.18: Precision at early cut off; Annotation selection using reciprocal rank.

| Measure | Reciprocal rank filtering | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 1/2 | 1/5 | 1/10 | 1/15 | Borda |
| Number of Topics | 30 | 30 | 30 | 30 | 30 | 30 |
| Images Retrieved | 504 | 627 | 676 | 748 | 748 | 748 |
| Relevant Images | 2187 | 2187 | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 376 | 462 | 513 | 565 | 566 | 563 |
| MAP | 0.1669 | 0.2020 | 0.2265 | 0.2456 | **0.2467** | 0.2439 |
| P@1 | 0.6333 | 0.7667 | 0.7333 | **0.8000** | **0.8000** | 0.7667 |
| P@2 | 0.6167 | 0.7333 | 0.7500 | **0.8000** | **0.8000** | 0.7833 |
| P@3 | 0.5889 | 0.6889 | 0.7556 | **0.8444** | 0.8111 | 0.8000 |
| P@4 | 0.5667 | 0.6500 | 0.7417 | **0.8583** | 0.8083 | 0.8083 |
| P@5 | 0.5733 | 0.6667 | 0.7467 | **0.8333** | 0.8200 | 0.8267 |
| P@10 | 0.5667 | 0.6700 | 0.7467 | 0.7967 | **0.8233** | 0.8033 |
| P@15 | 0.5533 | 0.6600 | 0.7156 | **0.7867** | **0.7867** | 0.7800 |
| P@20 | 0.5250 | 0.6417 | 0.7067 | **0.7767** | **0.7767** | 0.7750 |
| P@25 | 0.5013 | 0.6160 | 0.6840 | 0.7533 | **0.7547** | 0.7507 |

Table 3.7: Summary statistics, Reciprocal rank filtering.

### Score Rank

The idea of classifying each partial list of results based on the sample images contained in each list can be extended to consider the actual similarity value, instead of the position. In this case, the similarity score is used, which was obtained when computing the cosine similarity between the images. So, given a partial list $\sigma$, and a set $S$ of sample images, the "score rank" can be defined as:

$$\pi(\sigma, S) = \begin{cases} \text{maxscore}(\sigma, S) & \text{if } \sigma \text{ contains an element of } S \\ 0 & \text{otherwise} \end{cases}$$

where $maxscore(\sigma, S)$ returns the maximum score of the images from $S$ that are contained in the list $\sigma$.

By computing the score rank for each partial list, it is possible to filter, by using different thresholds, the lists that have a low score rank. The idea behind this approach is that lists where the most similar "correct" image is not too similar to the sample annotation, then the list is not representative of the concept behind the query. Algorithm 3.19 describes the procedure used to filter based on the score rank.

1: $I \leftarrow$ {set of images}
2: $T \leftarrow$ {set of topics}
3: **for** $t \in T$ **do**
4: $S_t \leftarrow$ {set of sample images for topic $t$}
5: **for** $s_i \in S_t$ **do**
6: $\sigma_i \leftarrow$ rank(I, $s_i$)
7: $\pi_i \leftarrow$ max_score($\sigma_i$, $S_t$)
8:
9: $\Sigma \leftarrow$ {list of rankings} $\equiv \{\sigma_0, \sigma_1, \ldots, \sigma_n\}$
10:
11: $\Pi \leftarrow$ {list of max_scores} $\equiv \{\pi_0, \pi_1, \ldots, \pi_n\}$
12: **for** $\pi_i \in \Pi$ **do**
13: $\delta_i \leftarrow$ inverse_percentage_difference($\pi_i$, $\Pi$) $\equiv \frac{max(\Pi)-\pi_i}{max(\Pi)}$
14: $\Upsilon \leftarrow$ {set of thresholds} $\equiv \{10\%, 25\%, 50\%, \ldots\}$
15: **for** $\epsilon_i \in \Upsilon$ **do**
16: $\Sigma_\epsilon \leftarrow$ filter($\Sigma$, $\pi_i$, $\epsilon_i$)
17: compute_performance($\Sigma_\epsilon$)

Figure 3.19: Algorithm: Compute Score Rank.

Figure 3.20 and Table 3.8 show the results obtained when applying this method over the results from S5. The results using score rank filtering do not show a clear distinction when to exclude annotations from the aggregated result list. The retrieval performance does not follow a clear pattern allowing to characterize different behaviors.

### Annotation Count

The goal is to investigate how many annotations are needed to get a good stable performance in terms of precision. It has been already shown that using multiple annotations is a good idea, but how many annotations are necessary?

Figure 3.21, based on the data in Table 3.9, shows how performance, in terms of precision at 5, is affected in relation to the number of visual annotations. When only one example is used, there is a large variance in the precision. On the other hand, increasing the number of annotations, reduces the variance and at the same time improves the performance.

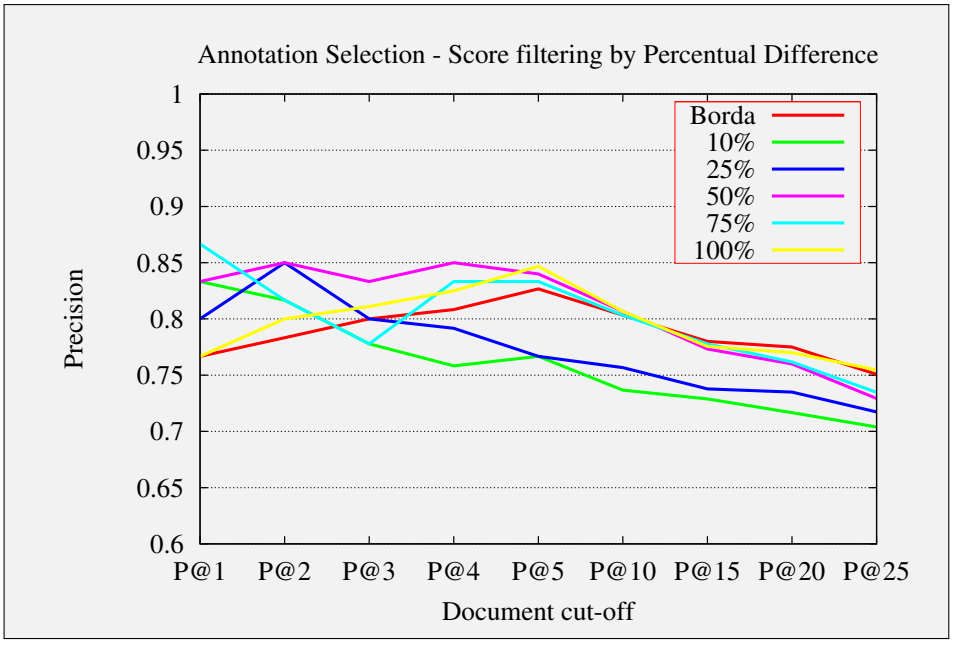



Figure 3.20: Precision at early cut off; Annotation selection using score rank.

| Measure | Score rank filtering | | | | | |
|---|---|---|---|---|---|---|
| | 10% | 25% | 50% | 75% | 100% | Borda |
| Number of Topics | 30 | 30 | 30 | 30 | 30 | 30 |
| Images Retrieved | 748 | 748 | 748 | 748 | 748 | 748 |
| Relevant Images | 2187 | 2187 | 2187 | 2187 | 2187 | 2187 |
| Relevant Retrieved | 528 | 538 | 547 | 551 | 566 | 563 |
| MAP | 0.2191 | **0.2259** | 0.2407 | 0.2416 | 0.2450 | 0.2439 |
| P@1 | 0.8333 | 0.8000 | 0.8333 | **0.8667** | 0.7667 | 0.7667 |
| P@2 | 0.8167 | **0.8500** | **0.8500** | 0.8167 | 0.8000 | 0.7833 |
| P@3 | 0.7778 | 0.8000 | **0.8333** | 0.7778 | 0.8111 | 0.8000 |
| P@4 | 0.7583 | 0.7917 | **0.8500** | 0.8333 | 0.8250 | 0.8083 |
| P@5 | 0.7667 | 0.7667 | 0.8400 | 0.8333 | **0.8467** | 0.8267 |
| P@10 | 0.7367 | 0.7567 | **0.8067** | 0.8033 | **0.8067** | 0.8033 |
| P@15 | 0.7289 | 0.7378 | 0.7733 | 0.7778 | 0.7756 | **0.7800** |
| P@20 | 0.7167 | 0.7350 | 0.7600 | 0.7617 | **0.7700** | 0.7750 |
| P@25 | 0.7040 | 0.7173 | 0.7293 | 0.7347 | **0.7547** | 0.7507 |

Table 3.8: Summary statistics, Score rank filtering.

From the data on Table 3.9, shows that precision becomes more stable when more than 5 annotations are used (for small P@ is even optimal). This shows that small number of annotations (3 to 5) makes it possible to have a stabilized precision. On the other hand the variability is larger as shown in Figure 3.22.

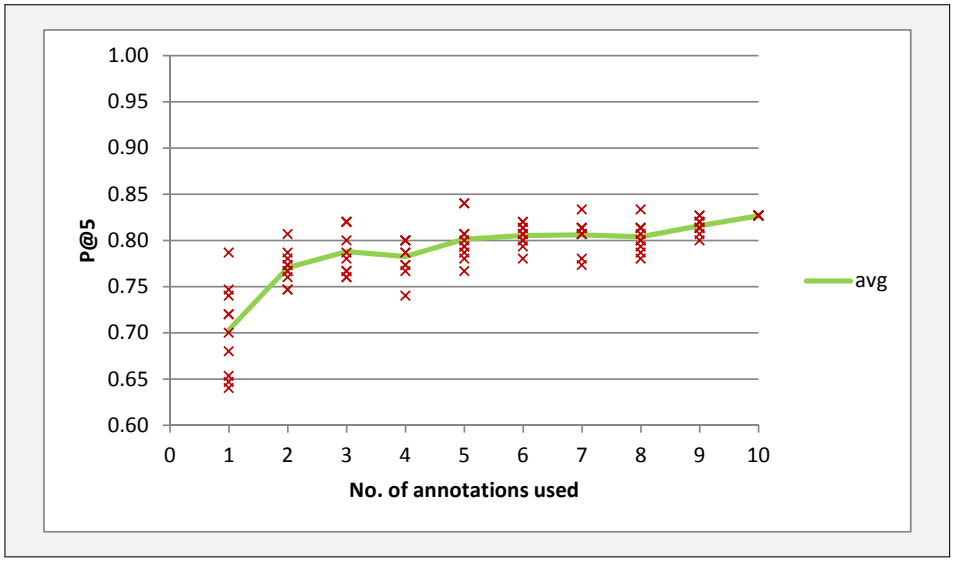



Figure 3.21: Annotation count, precision at 5.

| Notes | P@1 | P@3 | P@5 | P@10 | P@25 |
|---|---|---|---|---|---|
| 1 | 0.7567 | 0.6978 | 0.7033 | 0.6930 | 0.6393 |
| 2 | 0.7667 | 0.7878 | 0.7707 | 0.7357 | 0.6845 |
| 3 | 0.7867 | 0.8011 | 0.7880 | 0.7583 | 0.6893 |
| 4 | 0.7433 | 0.7889 | 0.7827 | 0.7677 | 0.7039 |
| 5 | **0.8167** | **0.8156** | 0.8013 | 0.7780 | 0.7228 |
| 6 | 0.8000 | 0.8000 | 0.8053 | 0.7930 | 0.7229 |
| 7 | 0.7900 | 0.8133 | 0.8060 | 0.7950 | 0.7352 |
| 8 | 0.7800 | 0.8155 | 0.8040 | 0.7970 | 0.7405 |
| 9 | 0.7867 | 0.8133 | 0.8160 | 0.7977 | 0.7469 |
| 10 | 0.7667 | 0.8111 | **0.8267** | **0.8067** | **0.7547** |

Table 3.9: Annotation count, average precision at 1, 3, 5, 10 and 25.

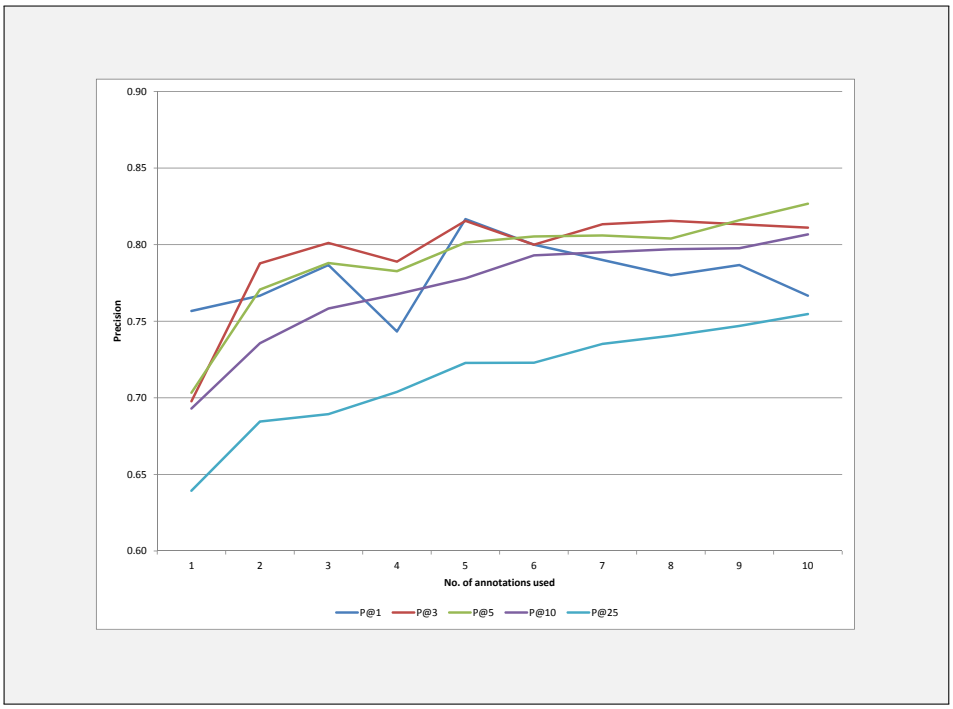



Figure 3.22: Average precision at 1, 3, 5, 10, and 25 for different number of visual annotations.

## Summary

Previous sections have shown that using annotations help improve retrieval performance of an image retrieval system. An interesting question is to determine the number of annotations required to obtain good results. This section presented three different approaches: (a) use the distance between the ranked list and the lists obtained by each annotation to detect outliers; (b) use mean reciprocal rank, based on the intuition that if a result list contains other annotations, then they refer to the same object; and (c) using tf-idf scoring as a quality measure of the list.

Finally, a graph showing how precision at 5-document cut-off (P@5) varies as the number of annotations used changes. An important contribution in this section is not necessarily the number of annotations required, but the fact that for a collection of 12K images using 5 annotations makes it possible to have a stabilized precision. This leads to the intuition that for a large

collection of images it will not be necessary to have too many annotations.

## 3.4 Discussion

Section 3.1 describes an image object retrieval using visual annotations. A significant improvement on precision can be achieved by (a) combining text and visual information, and (b) by using multiple visual annotations in combination with rank aggregation.

In Section 3.2, different methods for aggregating annotations are studied. The results obtained show that, for our dataset, Borda count outperforms Markov chains, which have been widely used in Web meta-search. Also a pre-retrieval aggregation approach is introduced by combining the *visual words* of the annotations into a single signature. This approach gives a slightly better performance, in terms of precision, than Borda count, and more efficient, in terms of computation time.

Finally, in Section 3.3, the number of annotations required is analyzed by comparing different methods for annotation selection. The most important conclusion in this section is that the number of annotations required for reaching a stable precision is not big, which can give the intuition that for larger collections it won't be necessary to have a large number of annotations for object retrieval.

Chapter 4

# Large Scale Image Processing

The previous Chapter 3 has proved that including visual information in the image retrieval process improves the quality of the results in terms of precision. But computing visual features is an expensive process especially when the collection grows large. Therefore Web image retrieval system do not include the computations of expensive visual features.

This is why a media extractor has been designed that can easily extract visual features and textual information to obtain a multi-modal system in a parallel fashion.

This chapter contains a description of the methodologies proposed to process a large scale image collection in an efficient manner using parallel computation as well as a description on how to build an image retrieval system in an efficient and scalable way. The media pipeline to process image features, as described in this chapter, is used in the applications presented in Chapters 5 and 6.

## 4.1 Media Extractor

The need to process a large collection of images has lead to the implementation of a media pipeline. This pipeline is used to process a set of images and

extract their main characteristics and features in a distributed way using the MapReduce paradigm.

Even though the computation required to process an image might not be a complex process, processing large amounts of data requires a huge amount of processing time, unless the job can be performed in a parallel and distributed fashion. A solution for this problem is to use the *MapReduce* paradigm presented in [15]. This programming model allows programmers to perform computations on a parallel fashion hiding the details of parallelization, fault-tolerance, data distribution, and load balancing. *MapReduce* offers a solution to handle and process large scale collections (which are the scope of the applications in Chapters 5 and 6) in an scalable manner.

## MapReduce

The main steps in the MapReduce model are: (a) input partitioning; (b) Map function; (c) data sorting; (d) Reduce function; (e) output writing. The first step splits the input data into independent blocks, which are sent to each of the nodes in the cluster. Then, each node executes the Map function over each line of the input data, generating a set of $< key, value >$ paired data. Afterwards, the data is sorted, based on the $key$, grouping all the values that share the same $key$. Then, each group of values are sent to different nodes to perform the Reduce function, which generates a final set of $< key, value >$ data. Finally, the output is written to the file system.

In this model, the user only needs to implement the two functions (Map and Reduce), which can be described as:

- Map: $(k_1, v_1) \rightarrow \{(k_2, v_2)\}$
  The Map function receives as input a pair $(k_1, v_1)$ and returns a set of one or more intermediate pair values $\{(k_2, v_2)\}$.

- Reduce: $(k_2, \{v_2\}) \rightarrow \{v_3\}$
  Then the Reduce function receives as input all the values corresponding to one of the keys generated by the Map function, and returns a set of values.

The Apache Hadoop [23] project is an open source project for scalable, distributed computing that holds several sub-projects. The main subproject

is Hadoop-MapReduce which consists on a MapReduce implementation for easily writing applications under this paradigm.

Hadoop provides its own file system (Hadoop Distributed File System - HDFS), which stores the data in a cluster of data nodes and provides reliability through data replication. One of the main characteristics of HDFS is its *rack-awareness*, allowing to reduce network traffic by taking into consideration the geographic location of servers. The MapReduce framework uses this information to prioritize assigning data processing tasks to the nodes that contain the data.

To execute a MapReduce task, the user needs to specify the input/output locations (in HDFS), the Map/Reduce functions, and other job parameters (e.g. number of nodes), defining the *job configuration*. The Hadoop job client submits the job and configuration to the *job tracker*, which manages the queue of all running jobs. Finally, it distributes the job over the cluster of nodes.

### Pipeline Implementation

The MapReducer job was implemented in Python using the Hadoop Streaming utility which is included in the Hadoop-MapReduce software framework. This utility allows to create and run jobs using any executable as a mapper or reducer.

Figure 4.1 shows an outline of the pipeline used to extract features from a set of images. First, the system receives as input a list of Flickr image identifiers which are used to download the corresponding image to further process it. For every image, the pipeline extracts a set of local and global features.

**Local Feature Extraction** In Chapter 3 local features are introduced as a suitable feature for image object retrieval and also for the capability of building fast retrieval systems based on these features using the bag of visual words approach. The local features used are:

**Harris-Laplace Affine Regions** Affine viewpoint covariant region detector. It computes the Harris interest point that corresponds to regions where the gradients are larger in two directions, which is the case for corners structures. The computation of the Harris interest points is

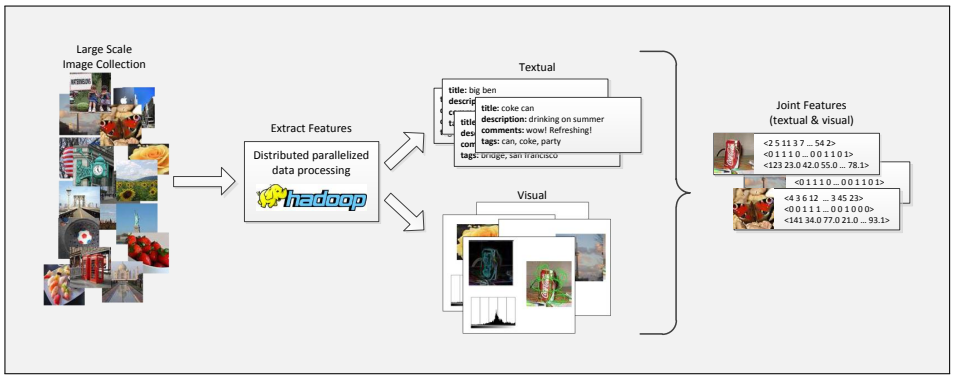



Figure 4.1: Outline of pipeline feature extraction process.

over a multi-scale representation, and then it selects the points where the Laplacian is maximal over scale. As a result, it obtains a set of distinctive points which are invariant to scale, rotation, and translation, and is also robust to illumination changes. For each point an affine region is computed by maximizing the intensity gradient isotropy over an elliptical region. These regions are described using a SIFT of 128 dimension.

**Hessian Laplace/Affine Regions** The Hessian affine regions are similar to Harris Laplace/Affine regions, but in this case it computes regions based on the Hessian detector, which detects blob structures. The final region is obtained in the same way as the Harris Laplace/Affine regions. These regions are described using a SIFT of 128 dimension.

**Global Feature Extraction** Global features are faster to extract than local ones and can be used for specific image processing purposes. In a first implementation of the media pipeline, 7 global features have been included. These features represents the texture, color and edges in a image. They are standard image descriptors that are included in MPEG-7.

**Color Histogram**. A color histogram describes the global color distribution in an image. To compute the color histogram, we define a discretization of the RGB color space into 64 color bins. Each bin contains the number of pixels in the image that belong to that color range. Two color histograms are matched using the Bhatta Charrya Distance [3].

**Color Layout**. Color layout is a resolution invariant compact descriptor of colors used for high-speed image retrieval [44]. Color layout captures the spatial distribution of the representative colors in an image. The image is divided into 64 blocks. For each block a representative color is obtained using the average of the pixel colors. Every color component ($YCbCr$) is transformed by a 8×8 DCT (discrete cosine transformation) obtaining a set of 64 coefficients, which are zigzag-scanned and the first coefficients are nonlinearly quantized.

**Scalable Color**. Scalable color can be interpreted as a Haar-transform applied to a color histogram in the HSV color space [44]. First, the histogram (256 bins) values are extracted, normalized and nonlinearly mapped to a 4-bit integer representation. Afterwards, the Haar transform is applied across the histograms bins to obtain a smaller descriptor allowing a more scalable representation. Two feature vectors are matched using a standard $L_1-$norm.

**CEDD**. The color and edge directivity descriptor (CEDD) incorporates both color and texture features in a histogram [9]. It is limited to 54 bytes per image making this descriptor suitable for large image databases. First, the image is split in a preset number of blocks; a color histogram is computed over the HSV color space. Several rules are applied to obtain for every block a 24-bins histogram (representing different colors). Then 5 filters are used to extract the texture information related to the edges presented in the image and classified in vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. Two descriptors are matched using the Tanimoto coefficient.

**Edge Histogram**. The edge histogram represents a local edge distribution of the image [44]. First, the image is divided in a 4x4 grid. Edge detection is performed to each block and the edges are grouped into 5 types: vertical, horizontal, 45 degrees diagonal, 135 degrees diagonal and non directional edges. The feature therefore consists of $16\times5 = 80$ coefficients. For matching two feature vectors the standard $L_1-$ norm is used.

**Tamura** Tamura *et al.* [50] identified properties of the images that play an important role to describe textures based on human visual perception. They defined six textural features (coarseness, contrast, directionality, line-likeness, regularity and roughness). We used 3 Tamura features

to build a texture histogram: coarseness, contrast and directionality. The Tamura features are matched using the standard $L_2-$norm.

**Process Outline** The main goal of the media pipeline is to receive images and output image features. Input images can be received in a variety of ways: URL, flickr ids, binary files, uncompressed streams, video files, etc. The media pipeline is responsible to receive this variety of formats and produce a common image format (jpg files) that is easily recognized by the feature extractors. Once the intermediate format is produced, the pipeline distributes the image among the extractors and collects the results. At the end of the process, the pipeline produces image features from the heterogeneous collection of images.

For example, in one of the applications developed, the pipeline received a list of flickrids (every image in Flickr has a public unique identifier) and the correspondent URL for the medium size image on Flickr. Every line of the input corresponds to a $< key, values >$ pair $< flickrid, URLimage >$, and each pair is processed by the mapper.

In this case, the first step is to download the image using the given URL. If the image is not available, then an exception is raised and a $< key, value >$ pair is generated indicating that the image was not found. If the image is available, it is downloaded and the feature extraction process starts. First, the local features are extracted. For each image, a set of temporary $< key, value >$ pairs are generated, where the key is the name of the feature (Harris or Hessian) and the value corresponds to the SIFT descriptor of the regions obtained for every feature: $< feature_{name}, flickrid + SIFT_{descriptors} >$. Then, for each image, the global features are extracted and a $< key, value >$ pair is obtained using as value the histogram of the feature: $< feature_{name}, flickrid + Histogram_{descriptor} >$.

Finally, the images are encoded using base64 encoding to be saved as text in such a way that the images can be available for further analysis or visualization. The $< key, value >$ pair generated in this step corresponds to $< image, flickrid + base64 >$. If any error occurs during any of the steps in the process an exception is raised. In this case the $< key, value >$ pair generated is $< errorlog, flickrid + errormessage >$.

The purpose of the Reducer program is to group the features and write the output in the desired format. The *key* values, as described before, correspond to the name of the feature extracted, so it is used to name the

output file (i.e. each output file will be named after a feature name) and will contain only one kind of feature. The final output $< key, value >$ pair will be $< flickrid, feature_{value} >$.

Figure 4.3 shows an schema of the media pipeline as described before.

```
1:  for i ∈ STDIN do
2:      flickrid ← key
3:      url ← value
4:
5:      # download image
6:      image ← downloadImage(flickrid)
7:
8:      # compute local features
9:      for every local feature do
10:         J ← computeLocalFeature(image)
11:         for j ∈ J do
12:             key ← localFeat_name
13:             value ← flickrid + SIFT_descriptor
14:             |print|(< key, value >)
15:
16:     # compute global features
17:     for every global feature do
18:         K ← computeGlobalFeature(image)
19:         key ← globalFeat_name
20:         value ← flickrid + Histogram_descriptor
21:         |print|(< key, value >)
22:
23:     # encode image in base64
24:     StringBase64 ← encodeBase64(image)
25:     |print|(< base64, flickrid + Stringbase64 >)
```

Figure 4.2: Algorithm: Media Pipeline basic mapper.

The pipeline has been built in such a way that any feature can be added or removed from the extraction process just by specifying the requirements in the configuration file, allowing to easily customize the image processing.

As seen, this pipeline can be extended with new feature extractors without problem: a new feature extractor should be able to read the common format

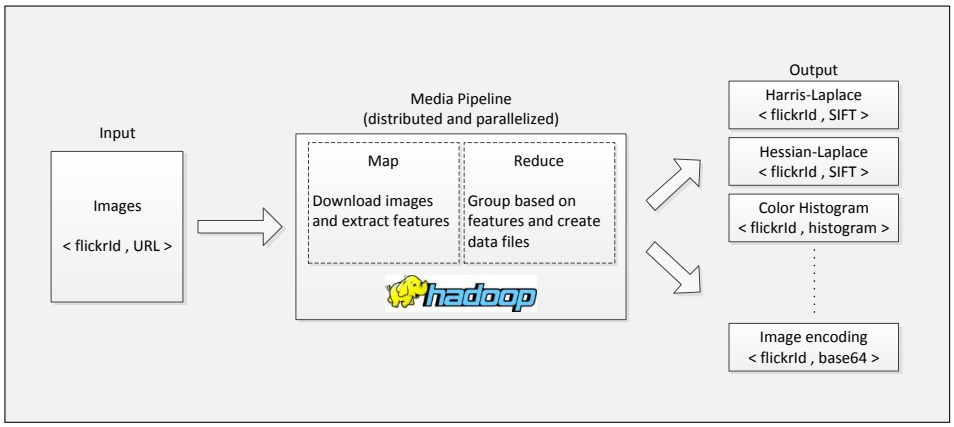



Figure 4.3: Description of the MapReduce job to download images and extract features.

and should be able to produce results as the pipeline expects. In the same way, new image formats can be handle as long as it can be converted to the intermediate format by the pipeline.

Besides the MapReduce scalability advantages, this implementation allows for the usage of heterogeneous image formats and for the low cost of adding new features to the process.

## 4.2 Building the Visual Vocabulary

As described in Chapter 3 to build an image retrieval system, based on the bag-of-visual-words approach, is necessary to build a visual vocabulary. The main issue that arises when using this approach is scalability. For example, from a medium size Flickr image it can be extracted 2,000 regions, in average. Hence, for a "medium" size collection of 12,000 images (used in Chapter 3) there are approximately 7 million SIFT vectors that need to be classified into visual words. Furthermore, it is required to run the clustering process, which is the most resource consuming stage.

To process a "large" collection of 1 million images the number of SIFT vectors can reach 600 million, making necessary to find an scalable solution to extract, cluster, and classify the features.

Mahout [32] is an Apache project with implementations of distributed machine learning algorithms on the Hadoop platform, including clustering, collaborative filtering and categorization. This library was used to perform the clustering task needed to obtain the visual vocabulary.

To prove the feasibility of using the *bag-of-visual-words* approach in a large scale image collection, 1 million images were processed, extracting Harris visual features. A subset of SIFT vectors was selected from these features to build the cluster model using Mahout K-means clustering. In this step, was necessary to modify the Mahout implementation to support our data format, and finally obtain a visual vocabulary that was built in a parallel and distributed way.

## 4.3 Indexing Images Using Lucene and SOLR

To build a retrieval system using the *bag-of-visual-words* approach and combine visual and textual features in a large scale image collections is essential to provide an efficient framework. Using a relational database to store and query the collection does not scale properly, showing severe performance degradation. For this reason, was decided to use an indexer and search engine to build the retrieval system.

Apache Lucene [31] is a high-performance information retrieval library that permits full text indexing and searching capabilities. Another interesting feature is the ability to perform efficient incremental indexes, which allows to connect it at the end of the media pipeline. The library provides an API to index and query the collection through Java.

Apache SOLR [48] is a search server running on top of Lucene, providing HTTP REST-like interface to index and query the index using XML or JSON. This allows to integrate easily the searching capabilities with other programming languages, as well as providing cache capabilities and a Web admin interface. By describing the data to be indexed as an XML document, SOLR is capable of adding it to the index, or if the document is already indexed, it updates it with the new data.

Using a Hadoop-MapReduce job that combines the textual features and the image local features is possible to build an XML file that can be indexed by Lucene-SOLR. Figure 4.4 shows the outline of the process were all the features from an image are aggregated to generate the XML (see Figure 4.5) that will be indexed. In this process, the local features are also classified

based on the cluster model so the final XML contains a field named *classes* that represents the classes from the cluster model.

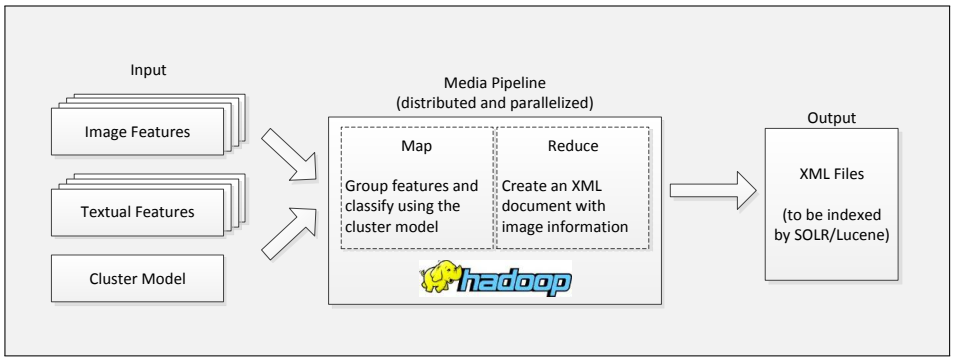



Figure 4.4: Description of the MapReduce job to create XML files.

```
<doc>
  <field name="id">1000061098</field>
  <field name="classes">7318 6253 3314 ...</field>
  <field name="points">5 132 2247 5 132 2247 ...</field>
  <field name="graph">41 53 41 381 41 442 41 483</field>
  <field name="thumb">8312 3092 842 2240 ...</field>
</doc>
```

Figure 4.5: Sample XML file to be indexed by SOLR-Lucene.

## 4.4 Discussion

The main purpose of this chapter is to validate hypothesis 2, introduced in Chapter 1. This section presents a media extractor framework, capable of processing large amount of media information, allowing to successfully combine the extraction of text and visual features in a parallel fashion. This framework takes into consideration the fact that, typically, search engines will only keep the full-sized image for limited period of time (∼seconds) for the extraction of all the features.

Chapter 5

# Ranking Images with Clicks, Tags and Visual Features

The following chapter presents a learned framework for ranking images which uses an additional form of UGC: click-data from image search logs. This framework corresponds to an application of the media extractor presented in Chapter 4, which uses a large collection of approximate 3.5M images.

By analyzing the query logs from an image search engine, is possible to exploit the click data as *user generated content*. This source of information can be used to enhance the intrinsic data from the query and/or the image collection.

The hypothesis is that the click on an image for a given query is a much stronger signal than a click on a snippet in Web search. A user can make a more sophisticated assessment of the relevance of an image based on the image thumbnail, than a user assessing the relevance of a document, given the summary snippet containing one or two lines of text. As the user can often see the entire content of an image by viewing the thumbnail, he may not click on many images, but when he does it can be considered a much more conclusive indicator of the relevance of that image to the query. Thus, while the system can rely on textual similarity between the query and the textual metadata to return topically related images, it can improve the quality of the results by ranking images according to their visual appeal as well. Although visual appeal is subjective, human perception of attractiveness of photos

has been found to be influenced by measurable quantities such as color distribution and coarseness [42].

This Chapter presents a learned framework for ranking images that employs click data from image search logs. Click data is generated in virtually limitless amounts by the users of search engines. Previous work in learning to rank Web results [27, 12] relies on the list structure of the results to determine which results are relevant and which are not. The assumption is that the user scans the ranked list of results from the top to the bottom until they find a relevant result, which they click. If the user clicks a result at rank three, they are assumed to have rejected the results at ranks one and two. Thus the click at rank three is considered to be a relative preference for that result; it is considered to be more relevant than the first two results. This structure of the clicked results is key to the success of learning to rank Web search results.

One important contribution of this work is that it is the first to use search engine click data to rank images in response to a user query. It is demonstrated that the block structures developed for list-based representations of the search results can be applied to image results, which have a grid based presentation. This is a significant difference because, in a list-based presentation, the second result always appears below the first result, and is followed by the third result. In image search, the placement of the image on the page is dependent on the browser dimensions, which are established by the user each time they open their browser. There is no guarantee that the third image will be to the right of the second image, and to the left of the fourth image. Furthermore, we do not know whether users scan the page from left to right, or top to bottom, and to what degree they use peripheral vision to reject non-relevant images. This work demonstrates that in spite of this, it is possible to predict clicked results with a high degree of accuracy.

As a second contribution, it is shown that a machine-learned model based on either textual or visual features outperforms the standard retrieval baseline, and combining text and visual features significantly improves retrieval over either feature set alone. This hypothesis is backed by the fact that textual features relate the content and context of the image to the query, while the visual features represent the aspects of the image that impelled the user to click.

As a third contribution, it is investigated whether a small subset of features account for the performance of the classifier. The present work finds that instead, the visual features work in combination to discriminate between

the clicked and non-clicked class, and that no single visual feature or class of features accounts for the performance of the classifier. By contrast, the textual features are less democratic, and one or two textual features carry most of the discriminative power.

## 5.1 Multilayer Perceptron

In learning from click data, massive amounts of continually changing data are available to analyze. In principle, several machine learning algorithms could be used. In practice, this amount of data requires an algorithm that is efficient and scalable. From the candidate algorithms, the perceptron was selected because it is efficient and has an online formulation so the training data need not be stored in memory all at once. Although the perceptron has been criticized for being limited to modeling linear relationships in the data, in this works it is used a multilayer perceptron with a sigmoidal hidden layer, which allows the modeling of arbitrarily complex patterns [4]. This can be an important feature in the retrieval task, where input signals come from different modes (textual and visual) whose combination via latent units can provide a powerful representation.

## 5.2 Click Data

The data consists of approximately 3.5 million distinct public images from Flickr, and approximately 600,000 unique queries, with their search results collected from the query logs of the Yahoo! image search engine. [1] The actual number of queries is more than 600,000 since many queries will be issued more than once, and the image results presented to the user may be different each time the query is issued, as well different users issuing the same query may click on different images.

The search results are filtered, by eliminating images that are not publicly available from Flickr. Then, for each query, blocks of images are constructed such that each block contains one clicked image as the positive example, and all un-clicked images displayed higher in the ranking as negative examples. Blocks with no negative examples are discarded (e.g. if the user clicked on all first $k$ results).

[1] http://images.search.yahoo.com visited January 2010

Thus, if a user clicked on results at ranks one, three and five in response to a query, two blocks are constructed: (a) First block has a positive example from the image at rank three, and one negative example from the image at rank two. The click at rank one is discarded because it is not possible to say that the user preferred the image at rank one over some other image they saw before. (b) Second block contains a positive example from the image at rank five, and two negative examples from the images at ranks four and two. This is shown in Figure 5.1. For the experiments presented in this section, 1,167,000 blocks were trained, and tested on approximately 250,000 blocks. Parameters were tuned on a held-out set, to maximize the prediction accuracy, using only the textual features. Also, the number of hidden layers was tuned, and the number of training iterations. The optimal performance required fewer than ten training iterations, and one hidden layer.

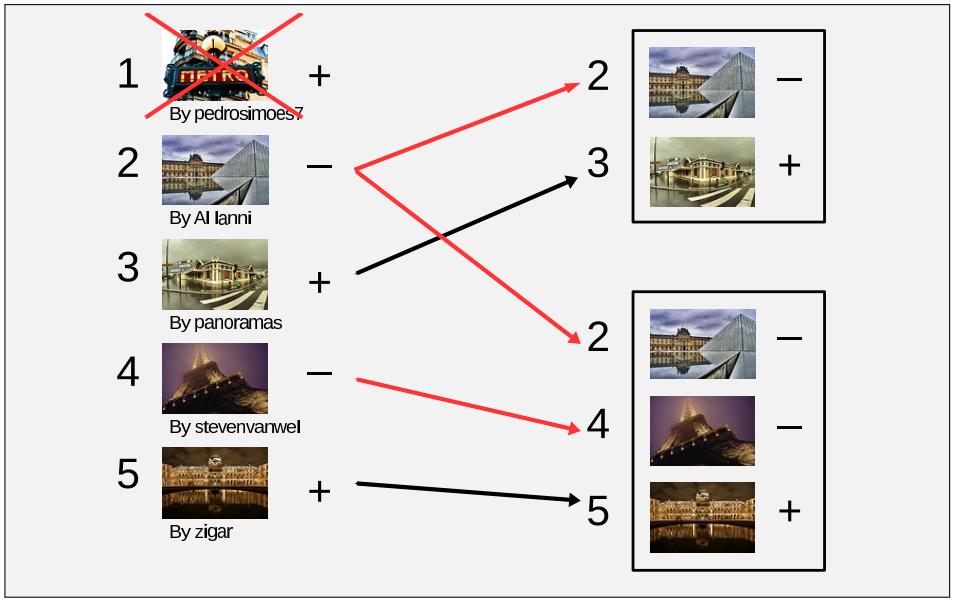



Figure 5.1: Blocks are constructed from the ranked list of clicked results in response to the query "Paris" as follows: Clicks at rank one are discarded. For each clicked image, a block consists of the clicked image and all non-clicked images ranked higher. In each block, clicked images are labeled as positive examples, and non-clicked images are labeled as negative examples. All photos shown in this figure are posted on Flickr.

## 5.3 Image Processing

For each image, it was collected the textual metadata associated to it, that is the title, the tags and the description. The tag sets are entered by the owner of the image, as well as by other people who have viewed the image, although the collected data show that the vast majority of tags are entered only by the owner. Tag sets are composed almost entirely of content terms, although some of the terms may not be helpful for the purpose of retrieval. For example, owners of an image might tag the image with the name of the camera, or the length of exposure. In addition the same tag set may be used to tag multiple images uploaded in bulk. In Flickr, the tags are lower-cased, spaces are removed, and commas are converted to spaces, thus tags may consist of terms that have been concatenated. As shown in the Figure 5.2, the title might contain terms that are not useful for the purposes of text-based retrieval, such as the date the image was taken. The descriptions often are written in natural language, and may be a sentence or two in length. Frequently the tags, title and description will be written in more than one language.

Due to the characteristics of the image repository, it is not possible to have textual metadata for all the images, or the metadata can be incomplete. Also, for some images it was possible to collect the metadata, but the image itself was no longer available for download. So for a certain portion of the data, the image was represented by either the text or the visual features.

## 5.4 Data Representation

After collecting the images, they were processed using the media pipeline described on Chapter 4. For each image, it was computed twelve textual features, and seven visual features. In addition a final binary feature, set to one for every example in the data, was intended to reduce the bias in the data. The feature set is normalized by row and by column as described below.

### Textual Features

For each image, it was defined four text components: (a) set of tags, (b) title, (c) description, and (d) the concatenation of the other three. Over

**Paris - Gare du Pont Cardinet - 28-07-2007 - 9h03**

Jolie gare des années 1920. Influence angkorienne et ressemblance frappante avec le palais du roi Narai, à Lopburi...Nice, small Paris railway station from the 1920's. Through the Angkorian influence, strong similarities with the King Narai's Palace, in Lopburi....

**Tags:**
gare
pont
cardinet
paris
batignolles
RER
1920
1925
Angkor
ankorien
Angkorian
Lopburi
Narai
pluie
rain
reflet
reflection
zebra
passage
piétons

Figure 5.2: An example of the data. The title for the image appears above the image, the description is below, and a sample of the tags appear to the right of the image. The title and description are entered by the person at the time of uploading the image. The tags may be entered by the owner of the image, or by other Flickr users. This image was taken by panoramas and appears with its metadata on the Flickr website.

each of this components, it was computed the cosine similarity between the query and the image where the terms were weighted by their *tf.idf* score. In addition, it was computed the maximum *tf.idf* score of a query term in the image, and the average *tf.idf* score of the term in the image.

### Visual Features

To represent the visual characteristics of an image, seven global features were computed: (a) color histogram, (b), color autocorrelogram, (c) color layout, (d) scalable color, (e) CEDD, (f) edge histogram, and (g) Tamura.

A complete description of each of these visual features is presented on Chapter 4.

**Normalization** The feature vectors were normalized by column and then by row. The mean and the standard deviation were computed for each column, except for the column representing the bias feature, and applied to each element of the matrix using the standard score:

$$SSFV(i,j) = \frac{FV(i,j) - \mu_j}{\sigma_j} \tag{5.1}$$

Where $FV(i,j)$ is the feature value in the row $i$ and column $j$, $\mu_j$ is the mean of the feature values on the column $j$ and $\sigma_j$ is the standard deviation of the column $j$. Rows were normalized with the L1 norm:

$$NFV(i,j) = \frac{SSFV(i,j)}{Norm(i)} \tag{5.2}$$

## 5.5 Evaluation and Results

To evaluate the prediction of the clicked event, there are several factors that need to be considered.

First of all, each block contains exactly one clicked event. For this reason, metrics that give a sense of the overall quality of the ranked list, such as Mean Average Precision, Precision at $k$, and Normalized Discounted Cumulative Gain (nDCG), are not meaningful. For example, if there are ten images in a block, precision at rank ten will always be 0.1.

As an alternative it could be possible to aggregate the block data per query, and evaluate the results that were returned for a given query, independent of users or session. This approach is flawed, because the results shown to one user, in a given session, might not be the same set of results shown to another user, or even to the same user in a different session. Because of this, it's not possible to state that an image clicked at rank three in one session was the same image shown at rank three in another session. The images shown to the user in response to the same query might have been completely different, and generate a completely different response.

Thus when evaluating a system using click data, the block structure must be preserved. In practice, to produce a ranked list of images, the features are computed over the query and the candidate images, and then use the trained model to produce a prediction for each image. Since the features are computed over a query-image pair, and do not depend on the clicks, the model will produce a prediction for each pair independent of the other pairs. Producing a ranking is then simply a matter of presenting the predicted-clicked images first in the ranked list.

Using the score obtained from the learning algorithm, it is possible to rank the images in each block, and then, using metrics that indicate the rank of the clicked event, evaluate how well the system predicted the clicked event in a block. For this purpose, two metrics were used: (a) Accuracy, which measures the frequency with which the system predicted the clicked event at the top of the ranked list; and (b) Mean Reciprocal Rank (MRR), which measures the rank of the clicked event.

The vector space model provides a retrieval baseline, with cosine similarity and *tf.idf* term weights, where the image is represented by the concatenation of its textual annotations. Then, the images for a given query are ranked within a block by the cosine similarity between the vector of term weights representing the query, and the vector of term weights representing the image.

For the learned baseline, the perceptron was trained with the cosine similarity feature over all fields concatenated, plus the bias feature, with the rows and columns normalized as described before.

Since both, the learned baseline and the retrieval baseline, are acting on the same information (with the exception of the bias feature which serves as a prior on the data), it is expected for them to be comparable. Table 5.1 shows the results of the retrieval baseline, and the learned baseline.

By incorporating textual features - albeit simple ones - it allows to weight the information carried in the tags, the title, and the description differently. This is important because each field differs substantially in character. The results shown in Table 5.1 confirm the hypothesis that ranking benefits from learning different weights for each of the metadata fields.

Using visual features to rank images seems intuitive because it allows to determine whether a photo is relevant based on the visual content of the photo itself. Since the metadata is not presented in the ranked list, and only shown after the image has been clicked, it is not considered in this ranking.

| | Accuracy | MRR |
|---|---|---|
| Retrieval baseline | 0.4198 | 0.6186 |
| Learned baseline | 0.4073 | 0.6104 |
| Text features | 0.5484 | 0.7034⋆ |
| Visual features | 0.5805 | 0.7233⋆† |
| Text + Visual features | 0.7512 | 0.8365⋆† |

Table 5.1: The results for predicting the clicked event in a block. The results indicated with a star are statistically significant compared to the baselines. The results indicated with a dagger are statistically significant compared to the model trained with textual features. All results are significant at the $p < 0.001$ level.

The visual features provide an indication of the content of the photo, and the intuition is that photos clicked in response to similar queries would have similar visual characteristics. The results for ranking solely on the visual characteristics of the data are shown in Table 5.1 The MRR results for both the textual features and the visual features are statistically significantly better than for the baseline results, at the $p < 0.001$ level. Finally, since accuracy is a binary measure, they were not tested for statistical significance.

The textual features and the visual features cover completely different aspects of the images. As both features perform well on their own, it is reasonable to expect that the combination of both types of features to outperform either category in isolation. The results in Table 5.1 confirm this intuition. The results for MRR for the text and visual features combined are statistically significant at the $p < 0.001$ level compared to the results for either the textual features or the visual features alone.

## 5.6 Analysis of Features

As presented at the beginning of this section, one of the objectives is to determine if a subset of the visual features accounts for the results. For example, it can be stated that people find pictures of other people interesting and click on faces even if they are not relevant to the query. To investigate this, the weight vector produced by the perceptron is examined. In the presented models, the feature weights range from approximately -2 to 2. Features closer to zero carry less discriminative information in the model

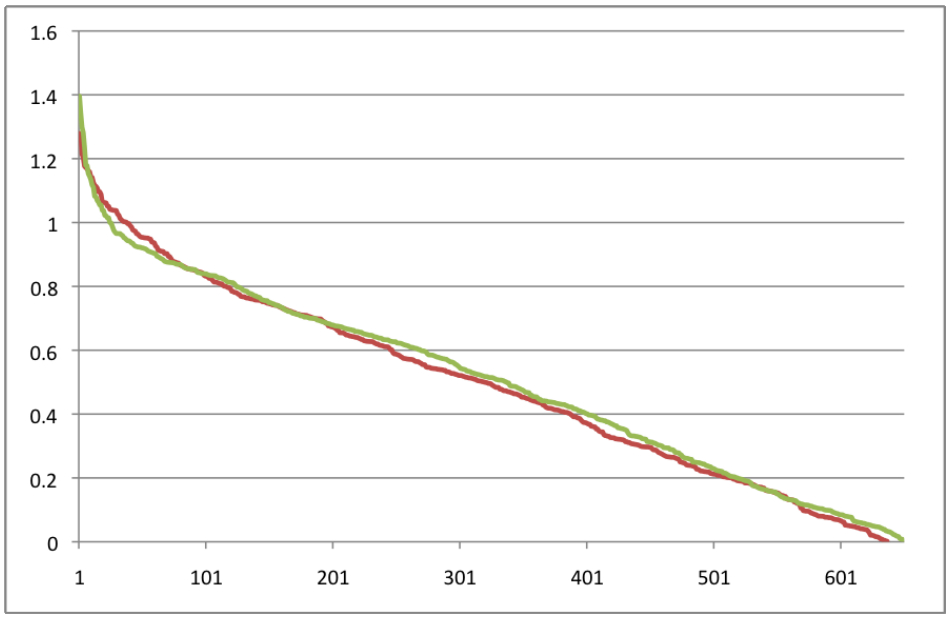



Figure 5.3: The distribution of feature weights as given in the model trained on visual features, and on all features.

than features farther from zero. Figure 5.3 shows the distribution of the absolute values of features for two models: one trained on only the visual features, the other trained on all features. Table 5.2 shows the ten most discriminative visual features, of which the top five are omitted from Figure 5.3 to make it easier to view. This shows that no single feature accounts for the discriminative power of the model. The distribution is more-or-less democratic, even with the top features included.

It would be convenient if it was possible to rely on a single class of features, e.g. the color histogram features or the Tamura features, to predict the clicked images. Unfortunately this did not prove to be the case. Figure 5.4 shows that the classes of visual features are more or less evenly distributed between highly discriminative features, and features with weights closer to zero. This leads to the conclusion that the features work in combination to determine which images will be clicked.

The textual features are more straightforward. Table 5.2 shows the top ten textual features, ranked by their weights in the model trained only on textual features. Cosine similarity between the query and the tags, all fields and the

| Rank | Visual Feature | Text Feature |
|---|---|---|
| 1 | CEDD_144 | Tags_sim |
| 2 | Tamura_3 | all_sim |
| 3 | CH_64 | title_sim |
| 4 | EH_19 | all_ave_tfidf |
| 5 | CEDD_79 | all_max_tfidf |
| 6 | SC_12 | tags_max_tfidf |
| 7 | EH_50 | title_ave_tfidf |
| 8 | SC_35 | desc_max_tfidf |
| 9 | CEDD_78 | title_max_tfidf |
| 10 | AC_80 | desc_sim |

Table 5.2: The ten most discriminative visual features and textual features, ranked by weights produced by models trained only visual (textual) features.

title provide the most information. The description field does not provide additional information and it is also the most sparsely represented field. We believe the similarity between the query and the tags to be particularly useful because more images are associated with tags than with the other textual fields, and tags are particularly succinct. They lack stop words, and often directly indicate the content of the image. In all models the bias feature contributed little.

### Discussion of Results

When a user queries for an image, and is presented with a grid of thumbnails, they often find what they were looking for without the need to click on an image. This is in contrast with traditional document retrieval on the Web, where the user is presented with a list of snippets which are surrogates for the document, and are much less likely to contain the information the user was looking for. For this reason, the click on an image seems to be a much stronger indicator of the image's relevance or interestingness.

By considering the click to be an indicator of the relevance of the image, it is possible to develop and evaluate image rankings with less dependency on editorial data. Another important factor is that click data is available on a large scale. It is already collected by the system and thus imposes no additional burden to the user or the search engine. In terms of creating an evaluation set, large volumes of queries can be used for training and

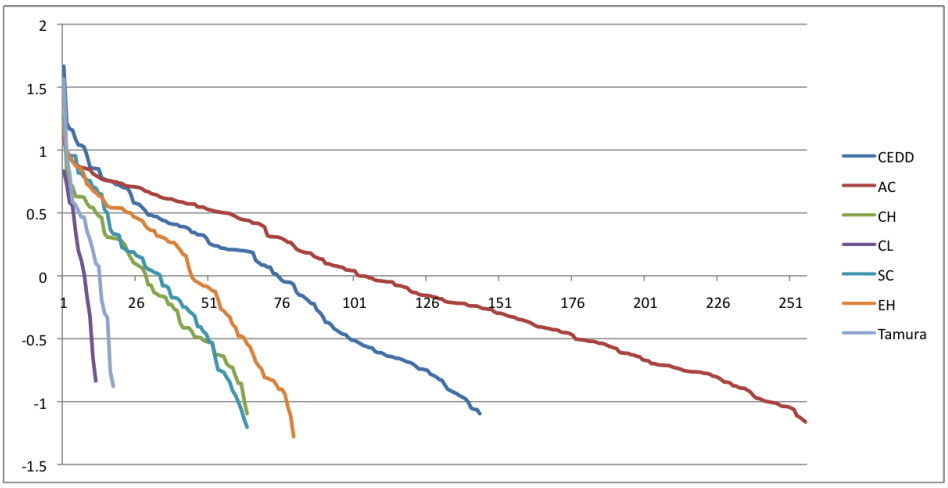



Figure 5.4: The distribution of feature weights for each class of visual feature.

evaluation, making the sample of data more representative of the population. If the system relies on editorially created data sets, the system will depend on labeling data with human assessors. There is a limit to the number of queries that can be assessed, and the depth in the rankings that can be labeled. Furthermore, it is not clear how to sample from the query stream to create a data set that represents what people are looking for. Click data suffers from none of these restrictions. Finally, what people look for in images, and on the Web in general, changes by season, holidays, current events, movie releases, and so forth. To reflect this, editorial assessments have to be done more often than practical, while large-scale click data can be sampled at any time.

Unlike Web search, the results of image search are not presented in a ranked list. Therefore, the block construction mechanism used might not be optimal, as it is not possible to safely assume that the user favored the clicked image over other images presented higher in the ranked list. It is more likely they favored the clicked image over the surrounding images. However, the layout of the images in the browser is dynamic, and the browser may be resized by the user at any time during the session. The work presented in this section demonstrate that it is feasible to predict the clicks based on the ranking of images, without considering the position of the clicked image in relation to the un-clicked images.

One of the main findings of this work relates to effective deployment of (low-

level) visual features for large scale image retrieval. It has been shown how visual features in combination with click data can be deployed effectively by a multilayer perceptron and achieve statistically significant improvement over the machine learned approach based on textual features. The combination of textual and visual information provides an additional boost in performance. A natural explanation for this is that the different features cover unrelated aspects of the image. Furthermore, no single subset of visual features accounts for the performance of the classifier. Bringing these features together makes the results both textually and visually relevant.

## 5.7 Discussion

The previous section demonstrates how to apply the block structure developed for list-based results presentations to a grid-based image search presentation. Although the assumptions about the bias due to the results presentation in Web search do not hold for image search, the resulting block structure still can be used to accurately predict clicked images. Therefore it is not necessary to know the layout of the image results to predict the clicked event.

Furthermore, to the best of our knowledge, this is the first study to show that the (global) visual features derived from the image content can be deployed effectively in combination with click data on a large scale, and that it outperforms text-based search. This provides evidence for the notion that users decide to click on an image based on the visual information depicted in the thumbnail.

Finally, it has been shown that textual and visual content can be combined in a principled and efficient way using a multilayer perceptron. In practice, is straightforward to use the model to produce a ranking of images, because the features depend on textual and visual properties of the image which are known, and not on its click history, which is unknown at the time of ranking. The perceptron is optimized to be efficient and scalable. It comes with an online version, such that training data need not to be stored and the training can be updated in a dynamic way.

Chapter 6

# Visual Diversification

This chapter studies different methods to diversify image search results, based on image features. The media extractor introduced in Chapter 4 is used to extract the different visual features used in this application.

In search applications, image search results are usually displayed in a ranked list. This ranking reflects the similarity of the image's metadata to the textual query, according to the textual retrieval model of choice. There may exist two problems with this ranking.

First, it may be lacking visual diversity. For instance, when a specific type or brand of car is issued as query, it may very well be that the top of this ranking displays many times the same picture that was released by the marketing division of the company. Similarly, pictures of a popular holiday destination tend to show the same touristic hot spot, often taken from the same angle and distance. This absence of visual diversity is due to the nature of the image annotation, which does not allow or motivate people to adequately describe the visual content of an image.

Second, the query may have several aspects that are not sufficiently covered by the ranking. Perhaps the user is interested in a particular aspect of the query, but does not know how to express this explicitly and issues a broader, more general query. It could also be that a query yields so many different results, that it is hard to get an overview of the collection of relevant images in the database.

This chapter propose to create a visually diverse ranking of the image search results, through clustering of the images based on their visual characteristics.

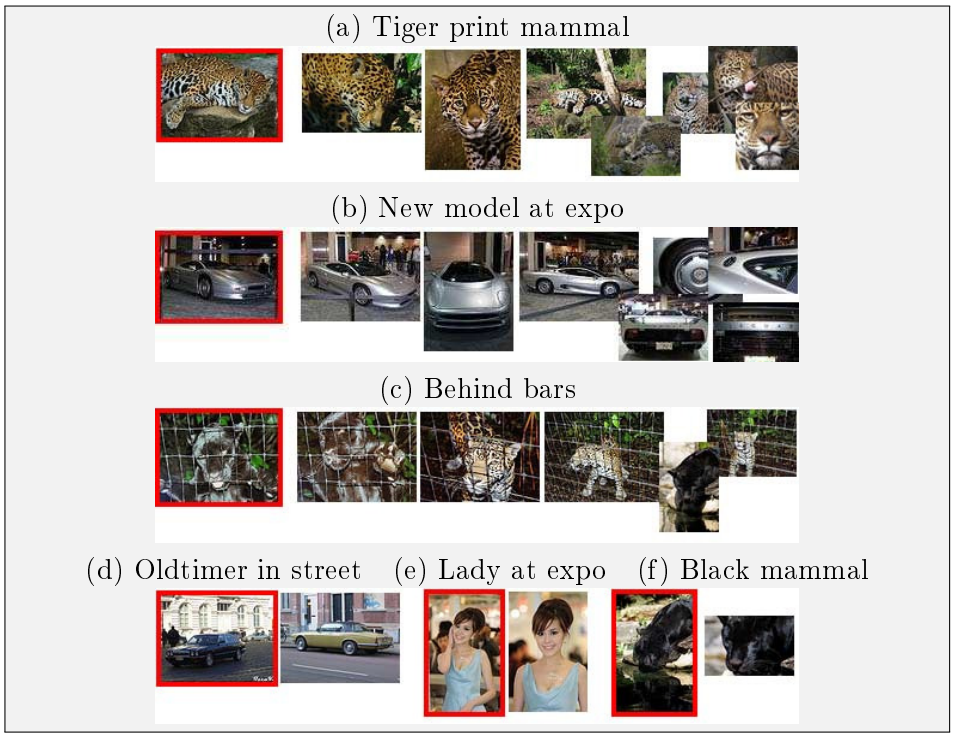



Figure 6.1: Example clustering: output of the reciprocal election algorithm for query *jaguar*. Cluster representatives are indicated by a red border.

To organize the display of the image search results, a cluster representative is shown to the user. Depending on the interest of the user in one of the representatives, he can then explore the other images in that cluster. This approach guarantees that the user will be presented a visually diverse set of images.

An example clustering of one of the algorithms presented, is given in Figure 6.1. The example uses the ambiguous query "*jaguar*". The image search result is not only ambiguous from a topical point of view (car, mammal), but also from a visual point of view. The algorithm separates mammals with a tiger print from black mammals and mammals behind bars. It also groups pictures from a new car model at an expo from cars in the street, and groups the accidentally found pictures of a lady at a car expo. The cluster representatives together form a diverse set of image search results.

The current chapter introduces new methods to diversify image search re-

sults. Given a user query, it first dynamically determines the appropriate weights of visual features, to best capture the discriminative aspects of the resulting set of images that is retrieved. These weights are used in a dynamic ranking function that is deployed in a lightweight clustering technique to obtain a diverse ranking based on cluster representatives. Three clustering algorithms are proposed, that are both effective and efficient, called: *folding*, *maxmin*, and *reciprocal election*.

In the case of folding, the original ranking is respected by preferring higher ranked items as representatives over lower ranked items. Maxmin, on the other hand, discards this original ranking and aims for maximal visual diversity of the representatives. The key idea behind reciprocal election is to let each image cast votes for other images that it is best represented by: a strategy close to the intuition behind a clustering.

These methods were implemented, and a performance evaluation was conducted over a large scale user-study using 75 topics of both an ambiguous and non-ambiguous nature.

## 6.1 Image Similarity

One of the key elements to any clustering algorithm or retrieval system, is a similarity measure between the objects. In content-based image retrieval or clustering, it is common to use several features simultaneously while calculating the similarity between images. These features represent different aspects of the image, such as color features, edge features, texture features, or alternatively concept detectors [44]. Each feature has its own representation (e.g. a scalar, a vector, a histogram) and a corresponding matching method (e.g. Euclidean distance, Hamming metric).

The fusion of different modalities into a single ranking is not trivial. Various techniques have been proposed to effectively fuse multiple-modalities into a single ranking, using a simple linear weighting, principle component analysis [56], or by using a weighted schema for aggregating features based on document scores [61].

This section introduces a dynamic ranking strategy that weights the importance of the different features based on the (normalized) variance of the similarities of all images in the results set. Although the similarity measure defined on the images has to reflect visual similarity, the clustering algorithms presented in this can use any distance measure between two images.

## Dynamic Feature Weighting

Based on the features described below, the similarity between two images can be expressed by six similarity values. These values, that may be of entire different range and distribution, need to be aggregated into one value for use in the clustering algorithm. Moreover, it is a priori unclear what is the relative importance of these features within the context of a specific set of image search results.

One assumption, is that the images retrieved by the textual retrieval model are topically relevant to the query. For each feature, the variance is computed over all image similarities within the set of image results. This variance is used as a weighting and normalizing factor at the same time. The image similarity according to a certain feature is divided by the variance of that feature in the result set. This brings image similarities according to different features in a similar range, and assigns a larger weight to features that are a good discriminator for the results that are presented to the user. The rationale is that when the variance of a certain feature is small, the images in the result set resemble each other in terms of that feature closely and thus it is a striking feature for this specific set.

More formally, the similarity between two images $a$ and $b$ is calculated as follows:

$$d(a,b) = \frac{1}{f} \sum_{i=0}^{f} \frac{1}{\sigma_i^2} d_i(a,b)$$

where, $f$ is the total number of features, $d_i(a,b)$ is the similarity between $a$ and $b$ in terms of the $i$-th feature and $\sigma_i^2$ is the variance of all image similarities according to the $i$-th feature within this set of image search results.

## Features

To characterize the visual contents of an image, six different features were extracted: (a) color histogram, (b) color layout, (c) scalable color, (d), CEDD, (e) edge histogram, and (f) Tamura. A complete description of each feature is presented on Chapter 4.

## 6.2 Clustering Algorithms

The present work uses three clustering algorithms: *folding*, *maxmin*, and *reciprocal election*. The folding algorithm appreciates the original ranking of the search results as returned by the textual retrieval model. Images higher in the ranking have a larger probability as being selected as a cluster representative. In one linear pass the representatives are selected, the clusters are then formed around them. The maxmin approach also performs representative selection prior to cluster formation, but discards the original ranking and finds representatives that are visually different from each other. Reciprocal election lets all the images cast votes for other images that they are best represented by. Strong voters are then assigned to their corresponding representatives, and taken off the list of candidates. This process is repeated as long as there exists unclustered images.

## 6.3 Experimental Setup and Results

The test corpus consists of a pool of 75 topics that were randomly selected from the Flickr search logs. Based on the method for resolving query ambiguity as presented in [60], the pool was divided in two groups: 25 textually ambiguous queries, and 50 textually non-ambiguous queries. This enables to measure the difference in performance of the visual clustering methods on both types of queries. Afterwards, for each query it was retrieved the top 50 results from a slice of 8.5 million photos from Flickr.

To retrieve a list of 50 results for the non-ambiguous queries a dual index relevance model was used, that produces a focused result set. On the other hand, to obtain the top 50 results of the ambiguous queries, a tags-only index relevance model was used, that produces a balanced list of diverse results. The details of both retrieval models are described in [57]. The intuition behind these choices is simple. If the terms in a query are textually diverse, then it is desired to produce a diverse set of images that embodies many possible interpretations of the user's query. Consider for example the query "jaguar", which carries at least three different word-senses that are present in the Flickr collection: the mammal, the car, and the operating system. On the other hand, if a query is textually non-ambiguous, e.g it has a clear dominant sense, the precision can be improved by returning more focused results. The query "jaguar x-type", which describes an specific model of the car, serves as an example for a non-ambiguous query. In both cases, the

result sets produced contain visually diverse images on which the described methods are tested.

To evaluate the performance of the proposed algorithms, their output is compared with a clusterings created by human assessors. The following sections present details on the establishment of the ground truth, the evaluation criteria and the experimental results.

### Human Assessments

To establish a ground truth, the set of 75 topics was divided into 8 independent, unbiased assessors and they were asked to cluster the images based on their visual characteristics. To do this, a well-defined procedure was implemented:

1. Select a topic, and inspect the top 50 results during at least one minute. This allows the assessors to get an overall impression of the images in the result set, to get a rough idea of how many clusters will be needed, and of their level of inter cluster dissimilarity. At any point in the assessment, the assessor could switch to this overview.

2. Form image clusters by assigning each image to a cluster pool by entering the cluster id. In total the assessor could create 20 clusters, and he/she could undo the last assignment if needed to correct for errors. See Figure 6.2 for an example of this interface.

3. Once all images in the results were assigned to a cluster, the assessor was asked to label each cluster and to identify one image in each cluster that could serve as a cluster representative.

At the end of this process, 200 topic clusterings were created by the assessors, because each topic was assigned to multiple assessors. This allows to calculate inter assessor variability, that provides a baseline during the performance evaluation of the algorithms.

### Evaluation Criteria

Comparing two clusterings of the same data set is an interesting problem itself, for which many different measures have been proposed. In this work,

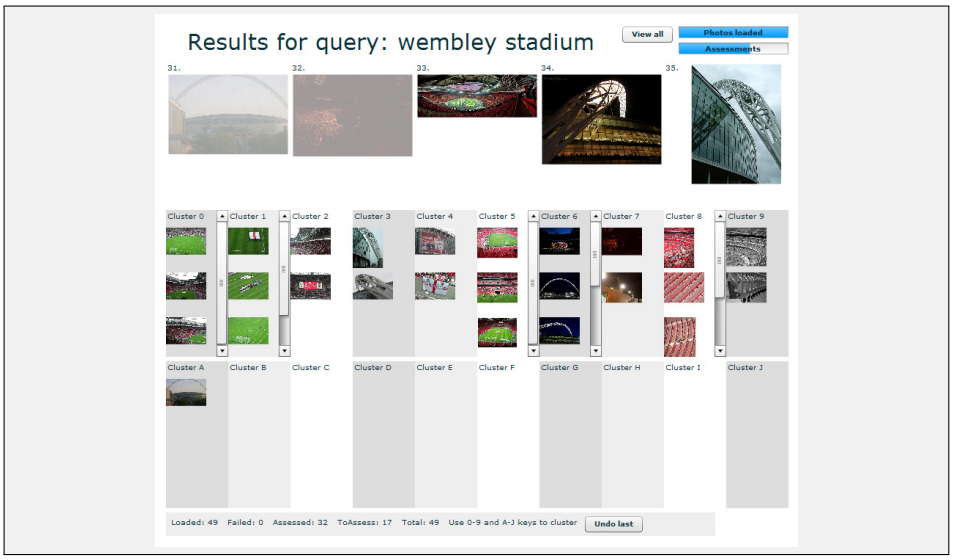



Figure 6.2: Example of the clustering interface used by the assessors.

two clustering comparison measures were adopted, each of them appreciating different properties.

The first clustering comparison measure used is the Fowlkes-Mallows index [20] that can be seen as the clustering equivalent of precision and recall. A high score of the Fowlkes-Mallows index indicates that the two clusterings are similar.

The second clustering comparison measure is the *variation of information* criterion, $VI(C, C')$, as introduced by Meilă [34]. The variation of information coefficient focuses on the relationship between a point and its cluster. It measures the difference in this relationship, averaged over all points, between the two clusterings, hence a low variation of information score indicates that two clusterings are similar.

### Results

All 200 clusterings of the 75 topics that were obtained as a result of the human assessments are compared to the clusterings generated by the different techniques. Using the described comparison measures, variation of information and the Fowlkes-Mallows index, performance is evaluated. Afterwards,

| | Inter-assessor variability | Random | Reciprocal election | Folding | Max-min |
|---|---|---|---|---|---|
| $FM$ | 0.419 | 0.139 | 0.250 | 0.282 | 0.214 |
| $VI$ | 1.463 | 2.513 | 1.975 | 2.081 | 2.129 |

Table 6.1: Average performance over all topics and assessors.

the results are presented for ambiguous topics separately, non ambiguous topics separately and all topics together.

**Interassessor Variability and Random Clustering.** The inter assessor variability is used as a base line for the performance evaluation. A technique can not be expected to produce clusterings that resemble on average the human created clusterings better than the assessors agree among themselves. To put a bound on expected performance on the other end as well, the human created clusterings were compared with randomly generated clusterings, so, for each topic, a random number (between 2 and 20) of clusters was generated. Every image was clustered randomly into one of the clusters, using a uniform distribution. As a result, it is expected that the performance of each of the three methods to lay within these two performance bounds.

**Results on Fowlkes-Mallows Index.** The best performing method according to the Fowlkes-Mallows index is folding, followed by reciprocal election and maxmin. Mean values and first and third quartiles are given in Figure 6.3 for both ambiguous and non ambiguous topics. The boxes show the average and the first and third quartiles for all comparisons, i.e. 50% of the 200 clustering comparisons fall within the box. The figure is showing the performance of reciprocal election, folding and maxmin; as well as a comparison results between a randomly generated clustering and the inter-assessor agreements according to the same comparison measure. It is important to note that a higher $FM$-index corresponds to better performance, as it indicates more agreement between the method and the assessors on point pairs that fall in the same cluster. Table 6.1 presents performance of the methods averaged over all topics. From this table, it can be observed that folding ($FM$-index = 0.282) outperforms both, reciprocal election ($FM$ = 0.250) and maxmin ($FM$ = 0.214).

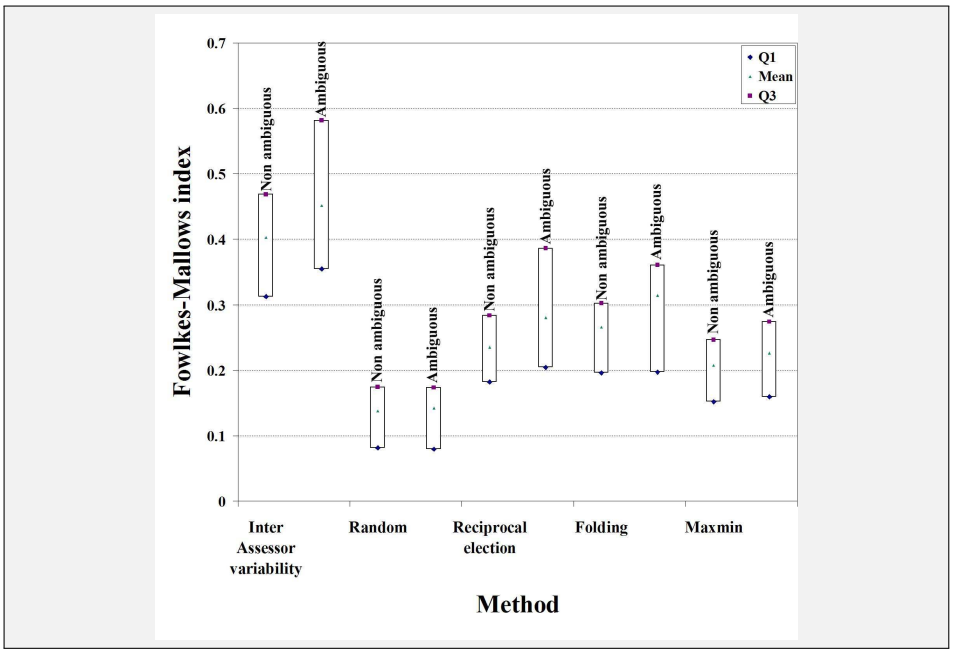



Figure 6.3: Performance of the three methods on the Fowlkes-Mallows index, compared to human assessments and the random baseline.

To test that these differences are statistically significant, $p$-values were computed. The null-hypothesis that all methods perform equally well is rejected both times, with $p = 0.006$ for reciprocal election and $p = 2.3 \times 10^{-9}$ for maxmin. Moreover, Figure 6.4 shows per topic the $FM$-index for folding against the $FM$-index for reciprocal election and maxmin. For every topic under the equality line $y = x$, folding outperforms the other method. With respect to reciprocal election, folding outperforms 58% of the topics, and for maxmin this value is 73%.

The Fowlkes-Mallows index measures the degree of agreement on point pairs that fall in the same cluster under both clusterings. This measure is therefore rather sensitive to the number of clusters. The folding approach benefits from its strong mechanism to automatically and dynamically select a proper number of clusters.

**Results on Variation of Information Metric.** A different relative performance is given by the variation of information criterion. According to

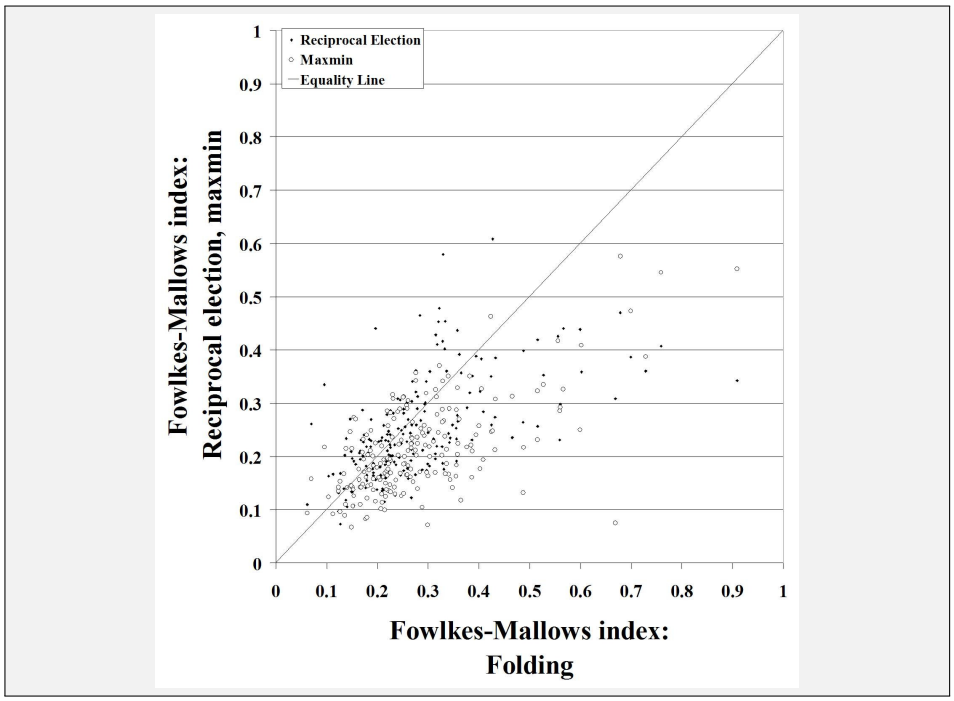



Figure 6.4: Performance evaluation per topic comparison. The plots show the performance of folding on the $x$-axis for each clustering comparison, with respect to the performance of the reciprocal election and maxmin on the $y$-axis. For every topic under the line, folding outperforms the corresponding other method.

this measure, reciprocal election outperforms folding and maxmin. Mean, first and third quartile performance is given in Figure 6.5, while Table 6.1 presents the performance averaged over all topics. In this case, a lower variation of information indicates a better performance. It denotes that there is less change in cluster membership while going from one clustering to the other.

Significance tests support the superiority of reciprocal election. The null hypothesis of all methods performing equally well is rejected with $p = 0.002$ for folding and with $p = 7.3 \times 10^{-7}$ for maxmin. Figure 6.6 presents relative performance comparisons per topic. It shows that the majority of folding and maxmin clusterings have a larger variation of information coefficient than reciprocal election, respectively 63% and 70%. In this figure, for every

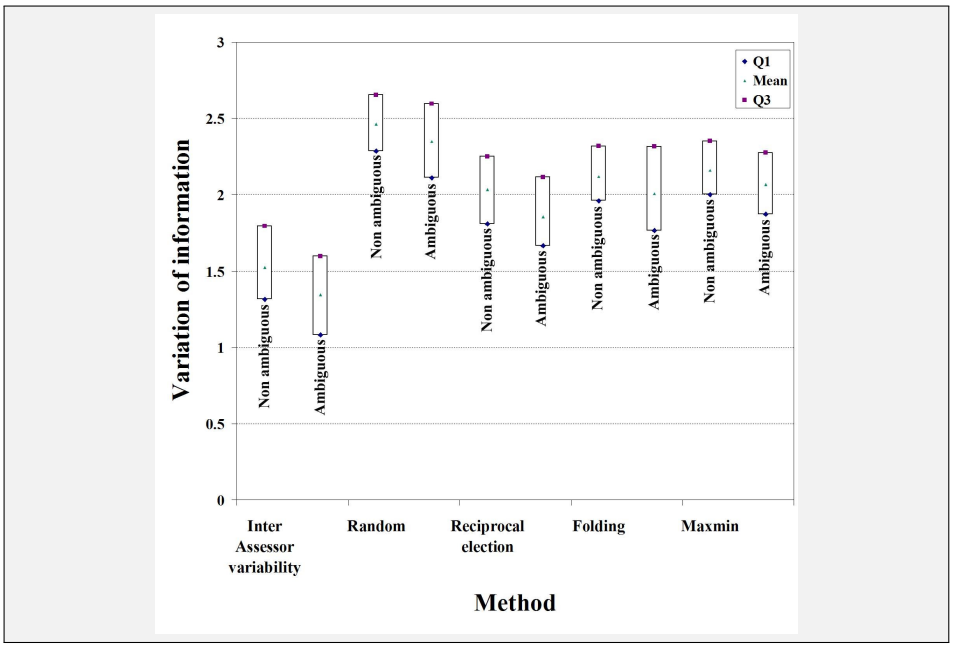



Figure 6.5: Performance of the three methods on the variation of information metric, compared to human assessments and the random baseline.

topic under the line reciprocal election achieves a better performance.

Rather than counting image pairs that fall in the same cluster under both clusterings, variation of information focuses on the relationship between an image and its cluster. It measures the difference in this relationship, averaged over all images, between the two clusterings. As this is a more general than counting successfully clustered image pairs, reciprocal election has a better overall performance. This might be due to how the approach follows the intuition behind a cluster. Images in a cluster should all be well represented by that cluster, a notion that translates directly to how the reciprocal ranks are used as votes.

**Ambiguous Topics vs. Non-ambiguous Topics.** Finally, by analyzing the results presented on Figures 6.5 and 6.3, it is possible to observe that assessors agree more on ambiguous topics than on non ambiguous topics. This is probably due to the fact that a more generally accepted clustering exists for topics that produce semantically different clusters. On non am-

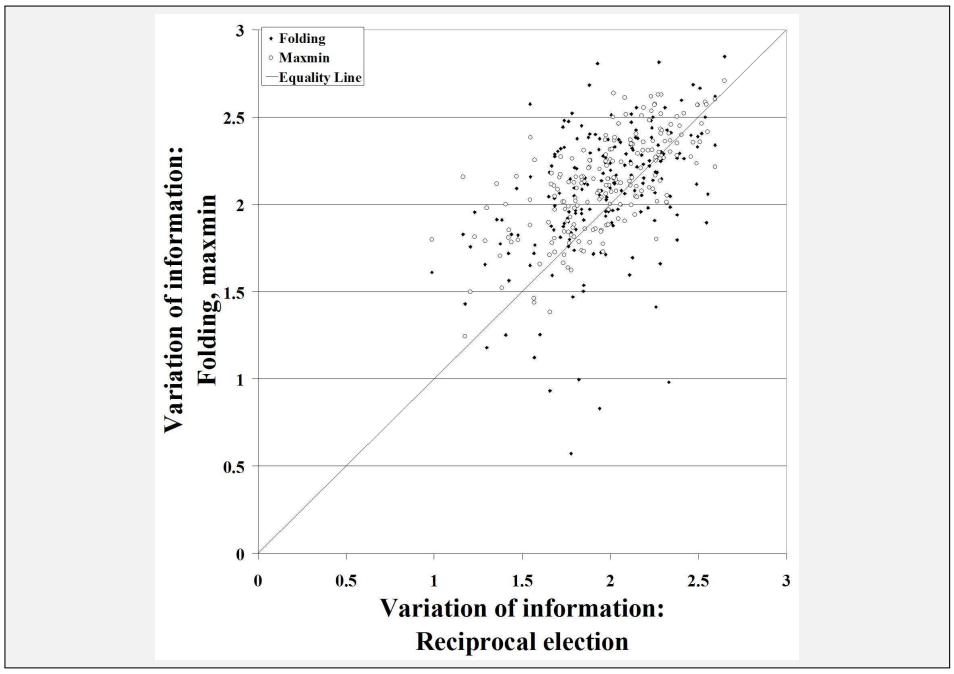



Figure 6.6: Performance evaluation per topic comparison. The plots show the performance of reciprocal election on the $x$-axis for each clustering comparison, with respect to the performance of the folding and maxmin on the $y$-axis. For every topic above the line, reciprocal election outperforms the corresponding other method.

biguous topics the assessors may choose more different criteria to base their clustering on. This behavior is also visible in the performance of the methods; the performance on ambiguous topics is significantly better than on non ambiguous topics. This indicates the existence of clear visual dissimilarity between semantically different images.

## 6.4 Discussion

Image search engines on the Web still rely heavily on textual metadata, causing a lack of visual diversity in image search results. Still, diversity is a highly desired feature of search results, no less in image search than in other search applications. This section presents new methods to visually diversify image search results that deploy lightweight clustering techniques.

These methods are effective, efficient and require no training nor parameter tuning. Given a user query, they adapt automatically to the set of image search results. The weights for visual features in a dynamic ranking function are computed on the fly to emphasize highly discriminant features for this set of results, and the number of clusters is adaptive as well.

The folding strategy respects the ranking order and picks the cluster representatives accordingly, while reciprocal election aims to optimize the clustering and the (s)election of cluster representatives by a voting strategy where each image determines a list of candidate images that it would be best represented by. Using a large user-study to establish a ground truth and a baseline, it was possible to evaluate the performance of each method.

Folding shows a better performance according to the Fowlkes-Mallows index, a performance measure that focuses on image pairs that can be formed with images from the same cluster. This indicates that the folding approach benefits from its strong mechanism to automatically and dynamically select a proper number of clusters. On the other hand, reciprocal election significantly outperforms the other methods in terms of variation of information, a more general performance measure. The selection of candidates and the decision on cluster membership both follow an intuitive notion behind a clustering. This might be explained by means of a low variation of information, and therefore conclude that reciprocal election achieves the strongest overall performance.

Chapter 7

# Conclusions

As stated in the title of this thesis, "Large Scale Image Retrieval based on User Generated Content", the main focus of the research presented is in the use of user generated content combined with the visual content of images, to improve the retrieval of images in large scale collections. Different forms of user generated content are explored, such as textual metadata (tags, title, description), visual annotation, and click-through-data, as well as, different techniques to combine these data were applied obtaining promising results.

CBIR is a complex task, and even though the research is extensive in this area, not too many methodologies has been applied on a broad domain such as the Web. Chapter 3 presents how the usage of visual annotations combined with aggregation techniques, over a diverse image collection, can significatively boost the performance of the retrieval task. Another observation is that aggregating annotations always brings an improvement, and then adding textual information, such as tags, provide an even bigger boost.

This thesis presents a novel approach on the use of rank aggregation applied to keyword-based image retrieval. Moreover, it presents different aggregation techniques that can be used in different stages of the image processing. Promising results are obtained from novel ideas, such as the pre-retrieval aggregation presented in Section 3.2, which creates a different representation of the original text query by combining the visual words, thus creating a synthetic query, which is then used in the retrieval task.

The problem of determining the quality of the existent annotations, as well as how many annotations are required to obtain good results in the retrieval

task, is addressed using annotation selection, presented in Section 3.3. Using different measures, it is possible to select the annotations that provide better results, and the minimum number annotations required to obtain satisfactory results. For example, for a medium-size collection of 12K images, only a small number of annotations (∼5 annotations) are required to obtain a stable precision. This can give the intuition that for larger collections it will not be necessary to have a large number of annotations.

Using the previous results as the proof of concept that combining UGC (visual annotations and tags) with visual information leads to better retrieval performance, Chapter 4 presents an scalable way of processing images to include their visual information in the retrieval task. This chapter describes a "media extractor" capable of processing large amounts of media information, allowing to combine the extraction of visual and textual features in a parallel fashion. This media extractor is used in two applications, which are described in the thesis.

In Chapter 5 a new form of UGC is included, the use of click through data as an implicit form of user assessment. Due that this data is available in vast amounts, it can provide relevant knowledge by associating the queries with the visual information.

Lastly, Chapter 6 shows how the use of visual features can provide diversification of results in an efficient fashion.

In summary, the objectives and hypotheses formulated in Chapterr̃efch:introduction were satisfied as:

**Objective 1: Use UGC to Improve Image Retrieval.** Boost the performance of the image retrieval process, in terms of precision, by combining different types of user generated content (UGC). Usually, UGC is considered to cover textual information provided by the user (e.g. title, description), but it can be extended to the visual annotations provided by the users.

*Hypothesis 1 (H1)* The use of UGC and visual information, specifically visual annotations, will improve the results of an image retrieval system in terms of precision.

*Validation of H1* Results from Chapter 3, shown in table 3.3, indicate that using a combination of visual annotations and tags improves the retrieval performance, in comparison to only using the tags information (system S1 vs. S4).

*Hypothesis 2 (H2)* Aggregating the results of different visual annotations,for the same topic, will significantly improve the retrieval performance in terms of precision. The agreement between the different result sets of the partial searches will lead to a more focused result set for the aggregated result set, with a higher precision at the top of the ranking.

*Validation H2* Results of the retrieval performance experiment, presented in Section 3.1 of Chapter 3, clearly show how the quality of the results significantly improve when applying rank aggregation over the results obtained with the content-based image retrieval system. In addition, Section 3.2 studies different forms of aggregation, such as rank aggregation and pre-retrieval aggregation. In every case, the use of aggregation outperforms the results obtained by only combining text and visual information. Moreover, Figure 3.14 shows how the pre-retrieval aggregation approach performs even better than S5, and at the same time, reducing to only one the number of queries required.

**Objective 2: Investigate Scalability of using Visual Information.** Investigate scalable solutions for processing and analyzing visual content. Computing visual features is an expensive computational process, especially for Web-size collections.

*Hypothesis 3 (H3)* Media content is available in large quantities, mostly unannotated. Image analysis and retrieval techniques do not live up to their textual counterparts in terms of retrieval performance. However large-scale applications of the techniques will improve performance.

*Validation H3* Chapter 4 presented technologies and methodologies that can be applied for the processing of large scale image collections using parallel computation. The approach introduced shows how image processing, that on a non-parallel fashion would take days, are performed in few hours, considerably reducing the time and making the inclusion of visual features feasible in the image retrieval context. For example, building a visual vocabulary is done in significantly less time. Moreover the research presented in Chapter 5 shows how retrieval performance is improved by incorporating the visual signals extracted from 3.5M images.

**Objective 3: Use Click-through Data to Improve Image Retrieval.** Research how using other forms of UGC, such as click-through data, that

has been successfully used in Web text retrieval can improve image retrieval performance.

*Hypothesis 4 - (H4)* Click feedback, provided by users searching for images on the Web, can be used as an assessment of the image, providing strong signal of the quality and type of images a user is really looking for, besides its topical relevance.

*Validation H4* Table 5.1, from Chapter 5, shows the Accuracy and Mean Reciprocal Rank of the proposed system. It shows that using the result of combining text and visual features is significantly better for the prediction of clicked images than using only the visual or text features. By considering the click as an indicator of the relevance of the image, it is possible to successfully predict if an image will be clicked or not. At this point, is important to distinguish between textual and visual relevance. Although state of the art methodologies in text retrieval has been used to ensure that the results retrieved are relevant from a textual point of view, is important to include visual features to rank higher the images that are of better visual quality, or that actually depict the object of interest.

**Objective 4: Use Visual Content to Improve Retrieval in Terms of Diversity.** Though powerful in its simplicity, keyword-based query formulation does not allow a user to fully express the visual characteristics of their information need. Therefore alternative query formulations have been proposed. In this thesis, this issue is addressed by investigating how search result sets can be diversified to better address the various users needs.

*Hypothesis 5 - (H5)* Based on the visual analysis of the content is possible to provide a meaningful clustering of the images and provide diverse search results.

*Validation H5* Due to the reliance on the textual information associated with an image, image search engines on the Web lack the discriminative power to deliver visually diverse search results. The textual descriptions are key to retrieve relevant results for a given user query, but at the same time provide little information about the rich image content.

Three methods for visual diversification of image search results have been investigated. The methods deploy lightweight clustering techniques in combination with a dynamic weighting function of the visual features, to best capture the discriminative aspects of the resulting set of images that is re-

trieved. A representative image is selected from each cluster, which together form a diverse result set.

Based on a performance evaluation, is observed that the outcome of the methods closely resembles human perception of diversity, which was established in an extensive clustering experiment carried out by human assessors.

The research presented in this work was largely driven by the problems of Yahoo!, a Web scale search engine. The work on diversification of image search results has led to some internal changes, including measuring search result diversity and ensuring that for some queries the results presented are in fact diverse to better addresses the user's needs.

## 7.1 Future Work

For future work, it would be interesting to extend some of the sections presented in this thesis.

Section 3.3 introduced how to select the best visual annotations for a given query based on the footrule distance between each result ranking or using the reciprocal rank score of each result ranking. This can be used to explore the diversity in the results given by each annotation. If one of the results list is not related with the other ones, perhaps the visual annotation is referring to another object with the same tag value. This can be used to show diverse results for a given query if the set of results from each annotation indicate that they can be partitioned in a certain way.

Another interesting enhancement would be to apply the object image retrieval system on a larger collection of images. Initially it could be bounded to a specific type of objects, for example, a collection of logos. The techniques and tools described in this thesis will allow to search for logos in a broad image collection, such as Web images, in a scalable and efficient way. Furthermore, click-through data can be used to provide an automatic assessment of the system.

Since one of the biggest motivations of this work is "How can we include the visual content of images in the retrieval process?", it is obvious that the scalability issue must be addressed. Since image analysis is a resource consuming task, is important to keep these processes to a minimum. Based on the promising findings obtained in our research, the pre-retrieval aggrega-

tion with the construction of visual concept models leads to the conclusion that the future work should focus on this.

Another use of click-through data is to associate queries with the clicked images, and consider them as visual annotations. This will allow to represent the queries with visual words, then apply the *bag-of-visual-word* approach and the aggregation techniques to obtain the set of results. Regarding the machine learning framework, the inclusion of local features seems a reasonable further step.

Finally, based on the diversity of results shown in Chapter 6, it would be interesting to include the social aspect on the result. For instance, given a query show the results from users that are socially connected with the user performing the query, and to allow diversity, limit them to only one result per user. Also, it would be interesting to include the geographic localization of the images displayed, hence diversify by location.

# Bibliography


[1] M. Ames and M. Naaman. Why we tag: Motivations for annotation in mobile and online media. In *CHI '07: Proceedings of the SIGCHI conference on Human Factors in computing systems*, New York, NY, USA, 2007. ACM Press.

[2] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 276–284, New York, NY, USA, 2001. ACM.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by probability distribution. *Bull. Calcutta Math. Soc.*, 35:99–109, 1493.

[4] C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, UK, 1995.

[5] J. C. Borda. Memoire sur les elections au scrutin. In *Histoire de l'Academie Royale des Sciences*, 1781.

[6] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32, New York, NY, USA, 2004. ACM.

[7] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959, 2004.

[8] O. Chapelle, P. Haffner, and V. Vapnik. SVMs for histogram-based image classification, 1999.

[9] S. A. Chatzichristofis and Y. S. Boutalis. CEDD: Color and Edge Directivity Descriptor: A compact descriptor for image indexing and retrieval. In A. Gasteratos, M. Vincze, and J. K. Tsotsos, editors, *ICVS 2008: Proceedings of the 6th International Conference on Computer Vision Systems*, volume 5008 of *Lecture Notes in Computer Science*, pages 312–322. Springer, 2008.

[10] E. Cheng, F. Jing, L. Zhang, and H. Jin. Scalable relevance feedback using click-through data for web image retrieval. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 173–176, New York, NY, USA, 2006. ACM.

[11] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[12] M. Ciaramita, V. Murdock, and V. Plachouras. Online learning from click data for sponsored search. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 227–236, New York, NY, USA, 2008. ACM.

[13] Corel clipart & photos. http://www.corel.com/products/clipartandphotos/, 1999.

[14] G. Csurka, C. Dance, J. Willamowski, L. Fan, and C. Bray. Categorization in multiple category systems. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 745–752, New York, NY, USA, 2006. ACM Press.

[15] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, 2008.

[16] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing tags over time. *ACM Trans. Web*, 1(2):7, 2007.

[17] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *WWW*, pages 613–622, 2001.

[18] J. L. Elsas, V. R. Carvalho, and J. G. Carbonell. Fast learning of document ranking functions with the committee perceptron. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 55–64, New York, NY, USA, 2008. ACM.

[19] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson. Searching the workplace web. In *WWW*, pages 366–375, 2003.

[20] E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[21] J. Goldberger, S. Gordon, and H. Greenspan. Unsupervised image-set clustering using an information theoretic framework. *IEEE Transactions on Image Processing*, 15(2):449–458, 2006.

[22] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[23] Hadoop. http://hadoop.apache.org/.

[24] Z. Harchaoui and F. Bach. Image classification with segmentation graph kernels. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.

[25] A. Hauptmann, R. Yan, and W.-H. Lin. How many high-level concepts will fill the semantic gap in news video retrieval? In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 627–634, New York, NY, USA, 2007. ACM.

[26] J. Huang, S. R. Kumar, and R. Zabih. An automatic hierarchical image classification scheme. In *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, pages 219–228, 1998.

[27] T. Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, New York, NY, USA, 2002. ACM.

[28] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of Computer Vision and pattern Recognition (CVPR2005)*, San Diego, USA, June 2005.

[29] R. Lienhart and M. Slaney. pLSA on large scale image databases. In *IEEE International Conference on Acoustics, Speech and Signal Processing 2007 (ICASSP 2007)*, 2007.

[30] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[31] Lucene. http://lucene.apache.org.

[32] Mahout. http://mahout.apache.org/.

[33] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.

[34] M. Meilă. Comparing clusterings: an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.

[35] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, October 2004.

[36] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[37] P.-A. Moëllic, J.-E. Haugeard, and G. Pitel. Image clustering based on a shared nearest neighbors approach for tagged collections. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 269–278, 2008.

[38] V. Murdock, R. van Zwol, L. Garcia, and X. Olivares. Image retrieval in a commercial setting. In Muller, editor, *ImageCLEF: Experimental evaluation in Image Retrieval*. Springer, 2010.

[39] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library: COIL, 1996.

[40] X. Olivares, M. Ciaramita, and R. van Zwol. Boosting image retrieval through aggregating search results based on visual annotations. In *MM'08: Proceeding of the 16th ACM international conference on Multimedia*, pages 189–198, New York, NY, USA, 2008. ACM.

[41] X. Olivares, R. van Zwol, and R. Baeza-Yates. The power of visual annotations in image retrieval. Submitted for publication.

[42] J. S. Pedro and S. Siersdorfer. Ranking and classifying attractiveness of photos in folksonomies. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 771–780, New York, NY, USA, 2009. ACM.

[43] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[44] P. Salembier and T. Sikora. *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.

[45] F. Schalekamp and A. van Zuylen. Rank aggregation: Together we're strong. In I. Finocchi and J. Hershberger, editors, *ALENEX*, pages 38–51. SIAM, 2009.

[46] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, editors, *Toward Category-Level Object Recognition*, volume 4170 of *LNCS*, pages 127–144. Springer, 2006.

[47] C. G. M. Snoek, B. Huurnink, L. Hollink, M. de Rijke, G. Schreiber, and M. Worring. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9(5):975–986, 2007.

[48] SOLR. http://lucene.apache.org/solr/.

[49] K. Song, Y. Tian, W. Gao, and T. Huang. Diversifying the image retrieval results. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 707–710, 2006.

[50] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6), 1978.

[51] Text REtrieval Conference homepage. http://trec.nist.gov/.

[52] Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 Workshop on Learning to Rank for Information Retrieval*, 2007.

[53] H. Tong, J. He, M. Li, W.-Y. Ma, H.-J. Zhang, and C. Zhang. Manifold-ranking-based keyword propagation for image retrieval. *EURASIP J. Appl. Signal Process.*, 2006:190–190, uary.

[54] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM.

[55] R. H. van Leuken, L. Garcia, X. Olivares, and R. van Zwol. Visual diversification of image search results. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 341–350, New York, NY, USA, 2009. ACM.

[56] R. van Zwol. Multimedia strategies for B3-SDR, based on Principle Component Analysis. In *Advances in XML Information Retrieval*, Lecture Notes in Computer Science. Springer, 2006.

[57] R. van Zwol, V. Murdock, L. Garcia, and G. Ramirez. Diversifying image search with user generated content. In *Proceedings of the International ACM Conference on Multimedia Information Retrieval (MIR 2008)*, Vancouver, Canada, October 2008.

[58] E. Vee, U. Srivastava, J. Shanmugasundaram, P. Bhat, and S. Amer-Yahia. Efficient online computation of diverse query results. In *ICDE '08: Proceedings of the 24th International Conference on Data Engineering*, pages 228–236, 2007.

[59] S. Wang, F. Jing, J. He, Q. Du, and L. Zhang. IGroup: presenting web image search results in semantic clusters. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 587–596, 2007.

[60] K. Weinberger, M. Slaney, and R. van Zwol. Resolving tag ambiguity. In *MULTIMEDIA '08: Proceedings of the 16th International ACM Conference on Multimedia (MM 2008)*, Vancouver, Canada, November 2008.

[61] P. Wilkins, P. Ferguson, and A. F. Smeaton. Using score distributions for query-time fusion in multimediaretrieval. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 51–60, 2006.

[62] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 504–511, 2005.

[63] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, 2005.