Image Selection for Text Illustration

Η
ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

Υποβάλλεται στην

ορισθείσα από την Γενική Συνέλευση Ειδικής Σύνθεσης
του Τμήματος Μηχανικών Η/Υ & Πληροφορικής
Εξεταστική Επιτροπή

από τον

Νικόλαο Δ. Χαλιάσο

ως μέρος των Υποχρεώσεων

για τη λήψη

του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ

ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΟ ΛΟΓΙΣΜΙΚΟ

Ιούνιος 2015

AKNOWLEDGEMENTS

I would like to thank my supervisor Assistant Professor Panayotis Tsaparas for his invaluable help, support, and precious time spent during the elaboration of this thesis. Most of all, I would like to thank him for the patience until it was completed.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Nikolaos Chaliasos.
MSc, Dept. of Computer Science & Engineering, University of Ioannina, Greece.
June, 2015.
Image Selection for Text Illustration.
Thesis Supervisor: Panayotis Tsaparas.

A common problem when constructing a web page, writing text, or creating slides is to find a set of images that will *illustrate the text*. As the already considerable sizes of digital image libraries expand, so does the time for a user to search and construct a set of proper images. Thus, the need for tools that will assist the search and selection process is pressing. In this work, we present TEXTILLE (TEXT ILLustration Engine), an end-to-end system for text illustration. Our system takes as input a set of *topics* from the text and produces a *relevant* and *homogeneous* set of images as output that illustrate the topics in the text. In our approach, we assume that images are associated with tags, and we build a search engine over the image tags. Using the topics as queries, we can retrieve images that are related to the topics. Given an initial pool of relevant images we compute the pairwise similarity among all available images, across different topics, using both textual (tags) and visual (color histogram) features. Our goal is to select a subset of these images that have high relevance and are also highly homogeneous, that is, they are highly similar across topics. We use a score function that captures both relevance and homogeneity, and we seek the set that maximizes this score. We show that our problem is NP-hard. Using a connection between our problem and the Densest K-Subgraph problem, we propose a series of algorithms for solving our problem. We evaluate our system algorithms on a large collection of images collected from Flickr, using travel-related query-topics.

Experiments with professional users in the fields of branding/corporate identity and graphic arts demonstrate that our algorithms improve the performance of relevance-based baselines. In many cases, the selection process corrects errors related to the ambiguity or broadness of the query terms.

ΠΕΡΙΛΗΨΗ

Νικόλαος Χαλιάσος του Δημητρίου και της Όλγας.
MSc, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πανεπιστήμιο Ιωαννίνων.
Ιούνιος, 2015.
Image Selection for Text Illustration.
Επιβλέποντας: Παναγιώτης Τσαπάρας.

Ένα συνηθισμένο πρόβλημα όταν κατασκευάζεται μια ιστοσελίδα, γράφεται ένα κείμενο ή φτιάχνονται παρουσιάσεις είναι η εύρεση ενός συνόλου από εικόνες οι οποίες θα *εικονογραφήσουν το κείμενο.* Τα μεγέθη των ήδη μεγάλων συλλογών από εικόνες συνεχώς αυξάνονται. Μαζί τους αυξάνεται και ο απαιτούμενος χρόνος για την αναζήτηση εικόνων επ' αυτών από έναν χρήστη. Έτσι, αναδεικνύεται μια άμεση ανάγκη για εργαλεία που μπορούν να βοηθήσουν τη διαδικασία αναζήτησης και επιλογής εικόνων. Σε αυτή τη δουλειά, παρουσιάζουμε το TEXTILLE (TEXT ILLustration Engine), ένα ολοκληρωμένο σύστημα για την εικονογράφηση κειμένου. Το σύστημά μας δέχεται ως είσοδο ένα σύνολο από *θέματα* του κειμένου και παράγει ως έξοδο ένα σύνολο από *σχετικές* και *ομοιογενείς* εικόνες που εικονογραφούν το κείμενο. Στην προσέγγισή μας, υποθέτουμε ότι οι εικόνες φέρουν επισυνάψεις (tags), πάνω στις οποίες χτίζουμε μια μηχανή αναζήτησης. Χρησιμοποιώντας τα θέματα ως ερωτήματα, δύναται να ανακτηθούν εικόνες που σχετίζονται με τα θέματα. Δοθέντος του αρχικού συνόλου από σχετικές εικόνες, υπολογίζουμε την ομοιότητα μεταξύ εικόνων σε διαφορετικά θέματα, χρησιμοποιώντας χαρακτηριστικά που προκύπτουν από ψηφιακή επεξεργασία της εικόνας (ιστόγραμμα χρώματος) και τις επισυνάψεις. Στόχος μας είναι να επιλέξουμε ένα υποσύνολο από αυτές τις εικόνες ώστε να είναι σε μεγάλο βαθμό σχετικές και ομοιογενείς. Για το σκοπό αυτό, ορίζουμε μια συνάρτηση αξιολόγησης η οποία αναθέτει στο σύνολο αυτών των εικόνων ένα ορισμένο σκορ το οποίο συνδυάζει τη σχετικότητα και την ομοιογένεια του συνόλου.

Δοθείσας της συνάρτησης ψάχνουμε για το σύνολο από εικόνες που μεγιστοποιεί το σκορ. Δείχνουμε ότι το πρόβλημά μας είναι NP-hard. Χρησιμοποιώντας μια συσχέτιση μεταξύ του προβλήματός μας και του Densest k-Subgraph προβλήματος, προτείνουμε μια σειρά από αλγορίθμους για την επίλυση του. Αξιολογούμε το σύστημά μας χρησιμοποιώντας ερωτήματα-θέματα σχετικά με ταξίδια και τουριστικούς προορισμούς από μια μεγάλη συλλογή εικόνων του ιστότοπου Flickr. Πειράματα με εξειδικευμένους αξιολογητές στο χώρο της εταιρικής ταυτότητας και της γραφιστικής δείχνουν ότι οι αλγόριθμοί μας έχουν βελτιωμένη απόδοση έναντι αλγορίθμων που στηρίζονται μόνο στη σχετικότητα. Σε κάποιες περιπτώσεις, η διαδικασία επιλογής διορθώνει σφάλματα που προκαλούνται από ασάφεια ή ευρύτητα των ερωτημάτων-θεμάτων.

CHAPTER 1. INTRODUCTION

1.1. Motivation

A routine task for professionals in the area of corporate identity, branding and graphic arts is finding proper images to illustrate a project, according to their needs. Usually, these needs are described in a text that accompanies the project, or is a part of it. In a typical scenario, they will spend valuable working hours searching in order to find a suitable set of images to illustrate each text. Besides the vast size of digital image libraries, and the number of topics in the text, another important requirement contributes to this lengthy task: the set of images besides relevant should be highly homogeneous as well, in order to fit the user needs. For one topic, it seems straightforward to choose proper images, but what happens when a text has multiple topics? Such questions raise the need for such a system that would automate this process.

To address this need, we propose *TEXTILLE* (TEXT ILLUSTRATION ENGINE). The goal is to provide an end-to-end system for illustrating text with images. In a typical usage scenario the user gives a text as input to the system, covering a set of topics. For example, in a tourist guide the text could be the description of a city, and the topics the main attractions. The topics are used as queries to retrieve a collection of relevant images, from a database of images. Given the pool of images, the system it produces a set of images that best illustrate the topics of the text. . The selected set should have images that are *relevant* to the topics and at the same time homogeneous. In the tourist guide example, this would mean that we want images that truly depict the given attractions and at the same time they are similar in theme. E.g, they are all taken at the same season, or at the same time, and have similar colors.

To the best of our knowledge this is a novel application problem, which also raises interesting research questions. Related work on this problem (R. Agrawal, 2011) does not consider the text as a whole but rather treats concepts (or topics) that appear in it independently. As we show in our experiments, using only the notion of relevance is not enough, and similarity plays a significant role in the selection process performance.

1.2. Contributions of the Thesis

In this Thesis we propose an automated tool that can assist the search and selection process of images, for a given text. We describe the design of a system for this problem, and we identify an new research problem in the image selection process, where the goal is to select a set of images that are both relevant and homogeneous. We provide algorithms for image selection and test them experimentally.

In this Thesis we make the following contributions:

- We consider the novel application problem of text illustration, and we provide an end-to-end system design for search and selection of images to illustrate a given text.
- We define the image selection problem as a combinatorial optimization problem, where the goal is to select a set of images that maximizes a score that combines image relevance with set homogeneity. Using a connection with graph theoretic formulation we link our problem we prove that is NP-HARD. We exploit the connection with the graph theoretic problem to propose algorithmic solutions.
- The proposed algorithms perform better that relevance and density-based baselines.
- We study our algorithms experimentally with a real world scenario, using a real-world image database from Flickr, and professional users in the areas of corporate identity / branding / graphic arts as evaluators. We demonstrate that our algorithms work better than baselines that use only relevance or homogeneity.

1.3. Thesis Outline

Chapter 2 describes the related work in areas related to our problem. Chapter 3 provides a high level description of the system. Chapter 4 provides the problem definition, problem complexity and the algorithms for solving our problem. Chapter 5 outlines and explains the experimental evaluation of our system. Chapter 6 concludes this thesis with a summary of our contributions and directions for future work and extensions.

CHAPTER 2. RELATED WORK

In this chapter we present related work. First, we present image search technology and retrieval approaches, then we present image diversity in search results and image search results. We also discuss how other research efforts in areas like clustering are related to our work. Finally, we present some more topics related to our work, such as densest k-subgraphs and quasi-cliques.

2.1. Image retrieval / search

Image search is a long standing research problem. The last decade has witnessed an advance of image search technology (W.H. Hsu, 2006) (Li J., 2008) (K. Yang, 2010). The majority of work efforts on image search mainly falls into the category of content-based image retrieval. Many of the content-based image search systems use for image indexing not only the visual information (color, texture, shape, etc.), but also its combination with textual information (tags, meta-information).

Different from general images with no associated meta-information, annotated images that have a set of user-provided textual descriptors (tags, meta-information), and thus tag-based image search can be easily accomplished by using them as index terms.

The authors of (Li J., 2008) proposed a tag relevance learning method which is able to assign each tag a relevance score, and they have shown its application in tag-based image search (Li X.R., 2008).

The authors of (L. Kennedy, 2009) proposed a method to establish reliable tags by investigating highly similar images that are annotated by different photographers.
The authors of (Liu D., 2009) proposed an optimization scheme for tag refinement based on the visual and semantic connection between images.

Also, among the content-based image retrieval systems that have been built are: (A. W. Smeulders, 2000), (J.Z. Wang, 2001). The first of the last two works presents a extensive review in a series of visual content image retrieval approaches and concludes by putting forth its view on: “the driving force of the field, the heritage from computer vision, the influence on computer vision, the role of similarity and of interaction, the need for databases, the problem of evaluation and the role of the semantic gap”. The second one, presents SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), an image database retrieval system, using high-level semantics classification and integrated region matching based upon image segmentation. It represents an image by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. Based on segmented regions, the system classifies images into categories which are intended to distinguish semantically meaningful differences. A measure for the overall similarity between images is defined by a region-matching scheme that integrates properties of all the regions in the images.

Furthermore, many new systems perform feature extraction as a preprocessing step, in order to obtain global image features like color histogram or local descriptors like shape and texture. In (E. Hadjidemetriou, 2004) the authors propose a multi-resolution histogram for capturing spatial image information. In (S. Jeong, 2004) the authors propose a Gaussian mixture vector quantization (GMVQ) in order to extract color histograms, and shows to provide better retrieval than uniform quantization and vector quantization with square error.

2.2. Diversity in search results

The problem of increasing the diversity in search results has been recognized as an important problem for search engines in order to better satisfy the different needs and intentions of their users. So, much work has been done trying to deal with it. A criterion that combines query relevance and novelty of information has been pointed out by the authors of (Goldstein, 1998), in particular by measuring the dissimilarity of a search result with respect to the ones before it in the ranked list. This criterion, referred to as Maximal Marginal Relevance, is then applied for re-ranking the query results.

Another work dealing with the issue of diversity is presented in (J. Wang, 2009), which addresses the problem of ranking search results adopting the idea of Modern Portfolio Theory from the field of finance. The authors argue that ranking under uncertainty in not just about picking individual relevant documents, but about choosing the right combination of relevant documents. The main idea lies in considering documents not individually but in combination with other documents, formulating the problem as a portfolio selection problem. “The selected documents should maximize the relevance, while minimizing the variance (i.e., the risk), where the notion of variance corresponds, inversely, to that of diversity”. They show that an optimal rank order is the one that balances the overall relevance of the ranked list against its risk level.

2.3. Diversity in image search results and clustering

Currently there are two popular approaches for enhancing the diversity in image search: search results clustering and duplicates removing. When performing search results clustering, a representative image can be selected from each cluster. Then only these representatives can be presented or other images can be put behind them in the ranking list.

The problem has also been studied for multimedia search, such as diversifying image search results by clustering images according to visual features (R. H. van Leuken, 2009). The authors of this work investigate three methods for visual diversification of image search results. The methods that they present 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 retrieved. A representative image is selected from each cluster, which together form a diverse result set.

The authors of (Y. Chen, 2004) have approached image retrieval by using spectral graph clustering. They introduce a new technique, cluster-based-retrieval of images by unsupervised learning (CLUE), for improving user interaction with image retrieval systems by fully exploiting the similarity information. “It retrieves image clusters by applying a graph-theoretic clustering algorithm to a collection of images in the vicinity of the query”. Clustering in CLUE is dynamic. In particular, clusters formed depend on which images are retrieved in response to the query. According to the authors, it can be combined with any real-valued symmetric similarity measure (metric or nonmetric). Thus, it may be embedded in many current CBIR systems, including relevance feedback systems.

Also, in (D. Cai, 2004), the authors propose a method to cluster web image search results into different semantic clusters to facilitate the user’s browsing. It is a hierarchical clustering method using visual, textual and link analysis. They are using a vision-based page segmentation algorithm, which separates a web page into blocks. The textual and link information of an image can be accurately extracted from the block containing that image. After that they construct a graph by using block-level link analysis techniques. Then, they apply spectral techniques to “find a Euclidean embedding of the images which respects the graph structure”. For each image, they have three kinds of representations: visual feature based representation, textual feature based representation and graph based representation. By using spectral clustering techniques, they authors claim that they can cluster the search results into different semantic clusters.

Furthermore, choosing the best set of images from an image database to illustrate a piece of text has been studied by (R. Agrawal, 2011), where the authors propose techniques for finding images from the web that are most relevant for augmenting a section of the textbook that they are trying to illustrate with images. They break their process in three steps. The first step is image assignment: Given a set of candidate images relevant to the various sections of a chapter and their relevance scores, the goal of the image assignment component is to allocate to each section the most relevant images, while respecting the constraints that each section is not augmented with too many images and that each image is used no more than once in a chapter. The second step is image mining: Their two algorithms AFFINITY and COMITY are used for obtaining the ranked list of top k images along with their relevance scores for a given section. The third and last step is image ensembling: an algorithm named ENSEMBLE that combines the different image assignments is deployed.

Moreover, automatic text to scene conversion using computer graphics techniques has been studied by (D. C. Brown, 1981), (S. R. Clay, 1996), (R. Lu, 2002), (B. Coyne, 2001). All the approaches described in these works receive a text as their input and try to produce scenes representing it as output.

Lastly, a work that is close to the notion of our system behavior is the WordsEye system developed by researchers at the AT&T Labs (B. Coyne, 2001) that receives English natural language as input and produces 3D scenes that represent the text, as its output. It relies on a large database of 3D models and poses to depict entities and actions. Every 3D model can have associated shape displacements, spatial tags, and functional properties to be used in the depiction process. The authors describe the linguistic analysis and depiction techniques used by their system along with some general strategies by which more abstract concepts are made depictable.

2.4. Densest k-subgraphs / Cliques / Quasi-cliques

The authors of (U. Feige, 2001) have studied the dense k-subgraph maximization problem, of computing the dense k-vertex subgraph of a given graph. That is, on input a graph G and a parameter k, the authors are interested in finding a set of k vertices with maximum average degree in the subgraph induced by this set. They prove that this problem is NP-hard (by reduction from Clique), and give approximation algorithms for the problem. They manage to obtain a polynomial time algorithm that on any input (G, k) returns a subgraph of size k whose average degree is within a factor of at most n^{δ}, where n is the number of vertices in the input graph G, and $\delta < 1/3$ is some universal constant. We will explain thoroughly how their problem is related to ours in section 4.6, where we provide a greedy algorithm for finding the k-densest subgraph, based on one of their approximation algorithms.

The authors of (V. E. Lee, 2010) present an extended survey on algorithms for dense subgraph discovery on single and multiple graphs. In a sense, all dense components of a graph are either cliques, which represent the ideal, or some relaxation of the ideal. The authors explore algorithmic approaches such as quasi-clique and densest subgraph. They look at basic algorithms for finding cliques and quasi-cliques and comment on their time complexity. Because the clique problem is NP-hard, they consider some more time efficient solutions.

In another work by (A. Bhaskara, 2010), the authors present an algorithm that for every $\varepsilon > 0$ approximates the densest k-Subgraph problem within a ratio of $n^{1/4+\varepsilon}$ in time $O(1/\varepsilon)$. Their algorithm, as they mention, is inspired by studying an average-case version of the problem where the goal is to distinguish random graphs from random graphs with planted dense subgraphs.

The authors of (R. Andersen, 2009), consider the problem of finding dense subgraphs with specified upper or lower bound on the number of vertices. They introduce two optimization problems. The first one is the densest at-least-*k*-subgraph problem (dalks), which is to find an induced subgraph of highest average degree among all subgraphs with at least *k* vertices. The second one, is the densest at-most-*k*-subgraph problem (damks), which is to find an induced subgraph of highest average degree among all subgraphs with at most *k* vertices. These problems are relaxed versions of the well-known densest k-subgraph problem. Their main result is that *dalks* can be approximated efficiently, even for web-scale graphs, and they give a (1/3)-approximation algorithm for *dalks* that is based on the core decomposition of a graph and runs in $O(m + n)$, where n is the number of nodes and m is the number of edges.

Also, they show that *damks* is nearly as hard to approximate as the densest k-subgraph problem.

The authors of (J. Pattillo, 2013) investigate the maximum γ-clique problem, $\gamma \in (0,1)$, from the mathematical perspective. The problem consists of finding a γ-clique of largest cardinality in the graph. According to the authors, they establish a series of fundamental properties of the maximum γ-clique problem, including the NP-completeness of its decision version for any fixed γ satisfying $0 < \gamma < 1$, the quasi-heredity property, and analytical upper bounds on the size of a maximum γ-clique.

Finally, the authors of (P. Rozenshtein, 2014) consider the problem of mining activity networks in order to identify interesting events, such as a big concert or a demonstration in a city, or a trending keyword in a user community in a social network. They define an event to be a subset of nodes in the network that are close to each other and have high activity levels, and they formalize the problem of event detection using two graph-theoretic formulations. They propose greedy approaches and they prove performance guarantees for one of them. Their results show that their methods are able to detect meaningful events.

CHAPTER 3. THE TEXTILLE SYSTEM

In this chapter we will give a high level description of the text illustration system and the different components.

3.1. The TEXTILLE System

Given a text as input, the goal of the system is to produce a *relevant* and *homogeneous* set of images that illustrate the topics in the text, as output.

In order to achieve this, we propose the TEXTILLE (TEXT ILLustration Engine) system. The system will provide an end-to-end process for text illustration with images.

In a typical usage scenario the user would give a text as input to the system, and it would respond with a set of images that would cover important topics of the text with relevant and highly homogeneous images. In figure 3.1 we can see a basic flow chart that will help us better explain our system's working cycle.

Firstly, the user provides a text as input to the system. The system analyzes the text and extracts the most important topics from it. The topic extraction process is beyond the scope of this Thesis (focused on the other ones), so we assume that the topics have been extracted either manually or automatically.

Secondly, with the extracted topics a search is performed in an image search engine (for each one of the topics), and an initial pool of images (associated with tags) is constructed from all the images of each query. A pairwise similarity measure, based on visual and textual features is formed between all the retrieved images, for image pairs across topics.

Then, using this similarity measure, a graph is constructed. Images are nodes and weights of the edges (if they exist) are similarities. Given this graph, the image selection process of TEXTILLE, selects a proper set of images, based on a scoring function that we will define later on. Finally, the selection process returns the set of images that it selected to the user. The main characteristic of this set, is that images in it are highly homogeneous and relevant.

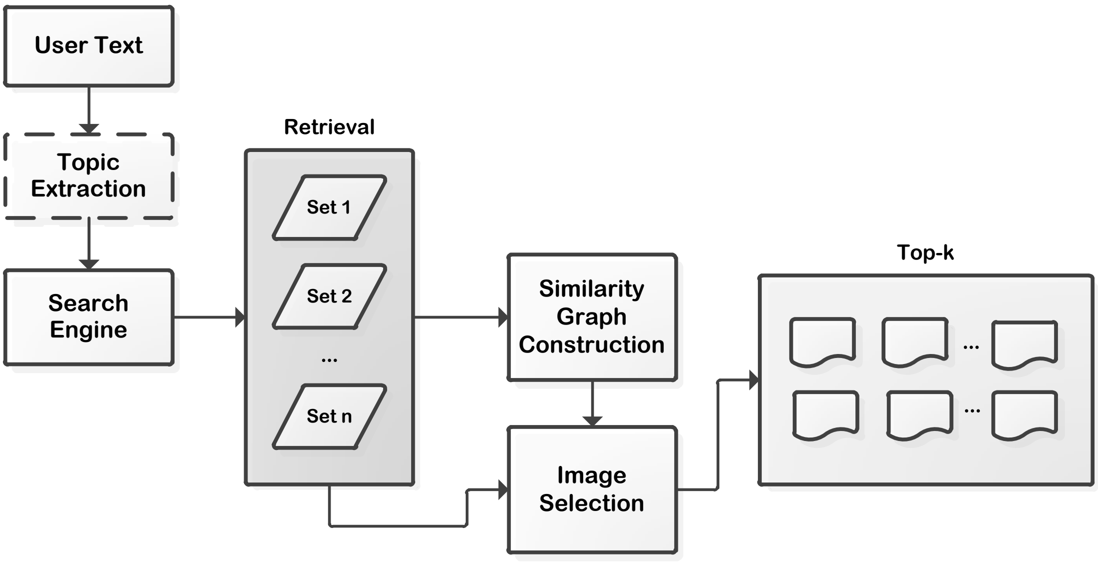

Figure 3.1. TEXTILLE System basic flow chart.

We describe the image search and graph construction in chapter 5. We will go into details with the image selection component in the next chapter.

CHAPTER 4. PROBLEM FORMULATION AND ALGORITHMS

In this chapter we consider in depth the image selection component. We will formally define the image selection problem, and study its complexity. We show a connection between our problem and the well-known k-densest subgraph problem, which motivates the algorithms that we consider in the rest sections of this chapter.

4.1. The Image Selection Problem Formulation

We now go into detail for the image selection. We will present the image selection problem formally and study its complexity.

Given a set $I = \{I_1, \dots, I_N\}$ of images and a set $Q = \{q_1, \dots, q_n\}$ of n topic-queries, we denote by C_i the set of relevant images for topic q_i, and as $C = C_1 \cup \cdots \cup C_n$ the complete pool of images retrieved for all topics. We provide the specifics of our image search engine in Chapter 5. We assume the existence of a relevance function

$rel: I \times Q \rightarrow [0,1]$, which given an image I and a query q it produces a relevance score $rel(I, q)$ between 0 and 1. For simplicity, when the query is known, we will use $rel(I)$ to denote the relevance of an image $I \in C_i$. The relevance score is provided by the search engine, as we explain in Chapter 5.

We also assume the existence of a similarity function $sim: I \times I \rightarrow [0,1]$ which given a pair of images I_i, I_j, $sim(I_i, I_j)$ is a value between 0 and 1 that captures the similarity between the two images. The similarity function we will consider combines the similarity between the tag annotations of the images, and the visual similarity between the images. We discuss the definition of similarity in detail in Chapter 5.

Given the pool of relevant images $C = C_1 \cup \cdots \cup C_n$ for the queries in Q, we want to select a small subset of them such that they are both relevant, and homogeneous. More specifically, we assume a parameter k and for each C_i we want to select a subset $S = \{S_1, \dots, S_M\} \subseteq C$ such that $S_i \subseteq C_i$ and $|S_i| = k$. We also want the images within each S_i to have high relevance score, while the pairwise combinations of images across different S_i sets have high similarity score.

To capture the quality of a set S in terms of both the relevance and homogeneity of the set we define the following score function:

$$score(S) = \sum_{\substack{S_i, S_j \in S \\ j > i}} \sum_{X \in S_i} \sum_{Y \in S_j} sim(\mathrm{X}, \mathrm{Y}) + \sum_{S_i \in S} \sum_{X \in S_i} rel(X)$$

Note that $score(S) \geq 0 \; \forall \; S \subseteq C$. The higher the score, the better the set S. Our goal is to find the set S that maximizes this score. We thus have the following definition of the Image Selection Problem.

Problem 4.1 [ImageSelection] Given pool of relevant images $C = C_1 \cup \cdots \cup C_M$ for a set of topic-queries $Q = \{q_1, \dots, q_M\}$ and a value k select a set of images $S = \{S_1, \dots, S_M\} \subseteq C$ such that $S_i \subseteq C_i$ and $|S_i| = k$ that maximizes the $score(S)$ function.

4.2. Problem complexity

Consider a graph $G = (V, E)$ and let nodes of G denote images, and edge weights denote similarities between images, with $sim: I \times I \rightarrow [0,1]$ the similarity function. We construct the graph as follows: Nodes are images. The addition of an edge connecting two nodes $(i, j) \in V,$ is dictated by the $sim(i, j)$, after checking if the similarity between these two nodes is above a certain threshold $t \in [0,1]$.

$$Given\, t \in [0,1]\ and\ sim(i, \mathrm{j})\ \forall\ (i, j) \in V, if\ sim(i, j) \geq t\ then\ \{i, j\} \in E$$

If an edge is formed between $i, j \in V$ on graph G, its weight is set to the value of $sim\left(I_i, I_j\right)$. For each node $i \in V$, we set the weight of i at $rel(I)$ of the corresponding image $I \in C_i$.

Proposition 4.1 *The ImageSelection problem is NP-HARD.*

Proof

We will now show that the ImageSelection problem is *NP-HARD* by reducing the maximum cardinality balanced bipartite clique problem to it. The decision version of the maximum cardinality balanced bipartite clique problem is defined as follows:

Instance: A graph $G = (V, E)$ and an integer k.

Question: Given G and a positive integer k, does there exist a maximum cardinality balanced bipartite clique with at least k nodes?

The problem is clearly in NP: given a graph G and a set of its nodes W, one can verify in linear time all nodes in W are connected and that the number of nodes in W is greater than or equal to k and that each side of the bipartition is of the same size. Given the input graph $G = (V, E)$ to the maximum cardinality balanced bipartite clique, we can form an instance of the ImageSelection problem, by considering the weights of all edges equal and 1 (pairwise image similarity is 1) and considering the weights of all nodes equal (all images have the same relevance). We ask if there is a solution of the ImageSelection problem of size at least k, with $score(S) \geq \binom{k}{2}$. If there is a such set of nodes $S \subseteq V$, then there is a maximum cardinality balanced bipartite clique in the graph with at least k nodes. So, there exists a set $S \subseteq V$ of size k that maximizes the $score(S)$, if and only if, there exists a maximum cardinality balanced bipartite clique of size greater than or equal to k.

In function maximization problems, proving that a function is *submodular* gives it some good properties.

Let S be a finite set. A function $f: 2^S \to R$ is submodular if for any $A \subseteq B \subseteq S \; and \; x \in S \backslash B,$

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B) \qquad \text{Eq. 3.3}$$

In our case, the function is supermodular. This can be easily proven, because as we add elements, that are connected with others already in the set, the relative difference is increasing.

Proposition 4.2 *The* score *function is supermodular.*

Proof

Our function is:

$$f(X) = \sum_{\kappa \in X} \sum_{t \in X} sim(I_k, I_t) + \sum_{\kappa \in X} rel(I_k), X \subseteq S. \qquad \text{Eq. 3.4}$$

$sim(I_i, I_j) \geq 0 \; \forall \; I_i, I_j \in S, \; rel(I_i) > 0 \forall \; I_i \in S, \; so \; f(X) > 0, \forall \; X \subseteq \; S.$

$$f(\emptyset) = 0 \qquad \text{Eq. 3.5}$$

$$f(A \cup \{x\}) - f(A) = \sum_{\kappa \in A \cup \{x\}} \sum_{t \in A \cup \{x\}} sim(I_k, I_t) + \sum_{\kappa \in \{x\}} rel(I_k) \qquad \text{Eq. 3.6}$$

$$f(B \cup \{x\}) - f(B) = \sum_{\kappa \in B \cup \{x\}} \sum_{t \in B \cup \{x\}} sim(I_k, I_t) + \sum_{\kappa \in \{x\}} rel(I_k) \qquad \text{Eq. 3.7}$$

If we replace the right hand side of Eq.6 and Eq.7 on the left hand side and right hand side of Eq. 3.3, after the cancellations we get:

$$\sum_{\kappa \in A \cup \{x\}} \sum_{t \in A \cup \{x\}} sim(I_k, I_t) \leq \sum_{\kappa \in B \cup \{x\}} \sum_{t \in B \cup \{x\}} sim(I_k, I_t) \Rightarrow$$

$$0 \leq \sum_{\kappa \in (B \backslash A) \cup \{x\}} \sum_{t \in (B \backslash A) \cup \{x\}} sim(I_k, I_t), \; which \; is \; always \; true.$$

Also, in our case the supermodular function has maximization constrains on the set size. Constrained supermodular function maximization is analogous to constrained submodular function minimization (K. Nagano, 2011). Constrained submodular function minimization problems are very difficult (Z. Svitkina, 2008), (S.Iwata, 2009), (G. Goel, 2009). In this work (K. Nagano, 2011), the authors discuss the following size-constrained submodular minimization (SSM): given a nonnegative integer $k \leq n$ the SSM problem asks for a subset $S \subseteq V\ with\ |S| = k\ that\ minimizes\ f(S)$. According to the authors, “This problem is NP-hard, and generalizes the densest-k-subgraph problem (U. Feige, 2001) and the graph partitioning problem (M. Garey, 1979) both of which are also NP-hard.” (K. Nagano, 2011).

4.3. Greedy Algorithm

In order to provide the solution for the problem described in section 3.5, we use a greedy algorithm that is subject to the following constraint: Maximize the score of the size-k set, but do not check the same image again in a future step. It receives a set of images and an integer k as input and produces a subset of the input of size $n * k$, as output, where n is the number of topics.

```
Algorithm 1: Greedy
    Input: Set (of images) $C\ with\ relevance$, integer $k > 0$, graph $G$.
    Output: Set (of images) $S \subseteq C, with\ |S| = n * k$.
for $i\ in\ 1..k$ do
    $S \leftarrow S \cup$ {select $x \in C \backslash S : f(S \cup x) - f(S)$ is max, each topic set contributes $k_t$ images}
return $S$
```

We want to produce a set $S \subseteq C, |S| = n * k$ that is of large score. Intuitively, we should promote image pairs with high similarity and images with high relevance in order to produce such a set. In each iteration the algorithm chooses the element $x \in C \backslash S$ that maximizes the score of set S and moves on to the next iteration until it adds $n * k$ elements in S.

4.4. Layered Greedy Algorithm

In order to provide the solution for the problem described in section 3.5, we use a variation of the Greedy algorithm described in the previous section. This algorithm receives a set of images and an integer k as input and produces a subset of the input of size $n * k$, as output, where n is the number of topics.

```
Algorithm 2: Layered Greedy
    Input: Set (of images) $C\ with\ relevance$, integer $k > 0$, graph $G$.
    Output: Set (of images) $S \subseteq C, with\ |S| = n * k$.
$graph\_layers \leftarrow find\_graph\_layers(G, k)$.
$\max\_score\_layer \leftarrow find\_layer\_with\_maximum\_score(graph\_layers, similarity, relevance)$.
$S \leftarrow S \cup \{\max\_score\_layer\}$
$remove\ \max\_score\_layer\ from\ graph\_layers$.
repeat until { $|S| = k$ }
        $S \leftarrow S \cup \{$ x $\in graph\_layers \backslash \max\_score\_layer$: the $score(S)$ is maximized and $k$ images per topic are selected $\}$.
return $S$
```

Given the set C we want to find a set of nodes in G that maximizes the $score(S)$ that we defined in section 3.4. According to our setup, this algorithm will first insert a number of nodes that is equal to the number of topics at one step, such that the score of this initial set is maximized. If we can see the graph as a set of layers, it will choose one image from each layer (or topic) and the whole set will have the maximum score. Then, the rest of the nodes until $|S| = n * k$, are selected by trying to maximize the score of set S every time a new node is selected.

4.5. Densest k-Subgraph Greedy Algorithm

In order to provide the solution for the problem described in section 3.5, we use an algorithm that is a biased version of the dense k-subgraph greedy algorithm described in (U. Feige, 2001). This algorithm receives a set of images and an integer k as input and produces a subset of the input of size $n * k$, as output, where n is the number of topics.

Algorithm 3: Densest k-Subgraph Greedy

```
    Input: Set (of images) $C$, integer $k > 0$, graph $G$.
    Output: Set (of images) $S \subseteq C, with\ |S| = n * k$
Sort the vertices of $G$ by order of their score.
$H \leftarrow n * k/2\ vertices\ with\ highest\ score\ in\ G$: all sets contribute $k$ images
Sort the remaining vertices by the set score they form with neighbors in $H$.
L $\leftarrow n * k/2\ vertices\ in\ G \backslash H$ with the max score they form with neighbors in $H$: each topic set contributes $k$ images.
return $T \leftarrow H \cup L$
```

The algorithm sorts all vertices by order of their score. Let H denote the $k/2$ vertices with highest scores in G. After that, it sorts the remaining vertices by the set score that they form with their neighbors in H. Let L denote the $k/2$ vertices in G/H with the maximum score formed with their neighbors in H. Finally, it returns $H \cup L$, that is of size n$*$ k, as it selects k images per topic.

4.6. HITS with Relevance Algorithm

In order to provide the solution for the problem described in section 3.5, we use an algorithm based on a biased version of the HITS algorithm (Kleinberg, 1998), that we augmented with the addition of the relevance score according to the work done by (L. Li, 2002). If s_i is the relevance score of an image I_i and h_i the hub value, $s_i * h_i$ instead of h_i is used to compute the authority values of images it points to. Similarly, if a_i is its authority value, $s_i * a_i$ instead of a_i is used to compute the hub values of images that point to it. This algorithm receives a set of images and an integer k as input and produces a subset of the input of size k, as output. The authority and hub score of each image is calculated and then we select the top-$k * n$ images with the maximum authority rank, where n is the number of topics.

```
**Algorithm 4**: HITS R
    Input: Set (of images) $C\ with\ relevance$, integer $k > 0$, graph $G$.
    Output: Set (of images) $S \subseteq C, with\ |S| = n * k$.
Initialize authority and hub values.
**while {** $change > 0.00005$ **}**
    **for each** element $i$ of $C$, **do**
        $auth[i] = 0$
    **for each** element $j$ of $node\ i\ inbound\ neighbours$, **do**
        $auth[i] += hub[j] * similarity(i, j) * relevance[j]$
    Normalize the authority values.
    **for each** element $i$ of $C$, **do**
        $hub[i] = 0$
        **for each** element $j$ of $node\ i\ outbound\ neighbours$, **do**
            $hub[i] += auth[j] * similarity(i, j) * relevance[j]$
    Normalize the hub values.
    Update $change$
end while
**for** $i\ in\ 1..k$ **do**
    $S \leftarrow S \cup \{x \in auth \backslash S: f(S \cup \{x\} - f(S)\ is\ max)$, each topic set contributes $k$ images}
return $S$
```

In the case of HITS shown above, we produce a set $S, with\ |S| = n * k$ from the $n * k$-most authoritative nodes, with the constraint that each topic provides k images. It uses the relevance score as a bias, when authorities and hubs collect and distribute their scores.

4.7. Densest Subgraph Greedy Algorithm

In order to provide the solution for the problem described in section 3.5, we use an algorithm that is a biased version of the Densest Subgraph Greedy algorithm. The difference from other algorithms is that this one builds a homogeneous and relevant set, without taking k into account.

Algorithm 7 Densest Subgraph Greedy Algorithm

```
    Input: Set (of images) $C$, graph $G$
    Output: Set (of images) $T \subseteq C$.
$S \leftarrow nodes\ of\ G$
while {$S \neq \emptyset$} do
    for each element $i \in C$, do
        $score(i) \leftarrow$ score that node $i$ has with all its neighbors in $G$
    $set\_scores \leftarrow set\_scores \cup S$
    $min \leftarrow find\_element\_with\_minimum\_score(score)$
    $remove\_one\_element(S, min)$

return $T \subseteq set\_scores$: $f(T)\ is\ max$, where $f$ is our scoring function
```

We want to produce a subgraph of G of large average score. Intuitively, we should throw away vertices that contribute poorly in the overall score, in order to produce such a subgraph.

The algorithm maintains a subset S of vertices. Initially $\leftarrow V$. In each iteration, the algorithm identifies i_{min}, the vertex of minimum score in the subgraph induced by S. The algorithm removes i_{min} from the set S and moves on to the next iteration. The algorithm stops when the set S is empty. Of all the sets S constructed during the execution of the algorithm, the set S maximizing $score(S)$ (i.e. the set of maximum average score, or the sum of similarities and relevance) is returned as the output of the algorithm.

4.8. Density Baseline Algorithms

Here we present two algorithms that are variations of existing ones, which in this case do not use relevance in the score, or as a bias. This is done, in order to provide baseline algorithms that use only the similarity value in the scoring function, and not the relevance value.

The first one is the *k-Densest Subgraph Greedy DB* Algorithm. The major difference with Algorithm 4 described in section 4.5 is that it does not use the relevance of an element in the scoring function, but only the similarity.

The second one is the *HITS* Algorithm. The major difference with the HITS R algorithm described previously in section 4.6, is the following: in *step 5* and in *step 12,* it does not use the relevance of an element as a bias for the scores of hubs and authorities. Instead it uses only the similarity.

CHAPTER 5. EXPERIMENTS

5.1. Introduction

In this chapter we describe the implementation and experimental evaluation of the *TEXTILLE* system. We then evaluate the image selection algorithms described in the previous chapter, and test them with a number of values for the input parameters α and k. We conduct the experiments with real world and user annotated images directly from the popular image platform Flickr (see section 5.3). To evaluate the quality of the constructed sets we used professional users in imagery industry to evaluate the quality of the constructed sets. In our experiment we study the tradeoff between textual and visual similarity, and the effect of parameter k for the size of the set. We also perform a comparative evaluation between the different algorithms we consider, and compare against simple baselines that consider only relevance, or only similarity as the criterion for selecting the images. Our results indicate that our algorithms work well in practice compared to these baselines, often improving on the relevance of the results by correcting issues related to the ambiguity & broadness of query terms. We present anecdotal results to give intuition about our results.

5.2. Implementation

In this section, we present the specifics our system design and implementation. We provide the specifics of our image search engine, the graph construction, the calculation of similarity, and the set selection process.

TEXTILLE receives a specific text as input from the user and returns a set of homogeneous and relevant images to his screen as output. Specifically, *TEXTILLE* consists of four components:

- The topic extraction from user text
- The image search and retrieval
- The graph construction
- The selection process

User Text

This step is omitted and out of the scope of our work. We consider the text given by the user as a series of high level given topics from which we extract our queries. Each topic, can be a series of one or more tags, that act as queries in the image search engine. We will now describe the process of this step briefly. At this step, the user provides the system with a piece of text. For this text, the system automatically extracts a series of representative topics that describe it. Each topic can be one or more terms that, and each one of these terms form the queries for the image search engine.

Search / Retrieval

This part receives the set of topics extracted from the text as input, and queries the image search engine, in order to form an initial pool of images from the images returned by each topic-query.

In order to implement an image search engine for our system we used Apache Solr (Targett, 2013), the popular and fast open-source enterprise search platform from the

Apache Lucene project (A. Bialecki, 2012), (Targett, 2013). In our implementation, it runs as a standalone full-text search server within the Tomcat servlet container. Solr as we mentioned is powered by Lucene, a powerful open-source full-text search library, under the hood. The relationship between Solr and Lucene, is like that of the relationship between a car and its engine. Solr is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead. Solr represents data as *Documents*, where a Document is the unit of search and index. An index consists of one or more Documents, and a Document consists of one or more Fields. It's scoring system is based on the tf-idf (A. Bialecki, 2012) formula. We provide more details on the platform's specifics and scoring system in the appendix. In the *schema* that we used for our implementation we declared (only the non-default are mentioned):

- what kinds of fields there are (tags, title, histogram_ch, histogram_edh, views, comments).
- which field should be used as the unique/primary key (id).
- which fields are required.
- how to index and search each field.
- tokenizers, analyzers, stopwords filter (for tags, title).
- boost certain documents (number of views, number of comments). We used the log value of the views as a boost for each document. We used the same approach for boosting a document according to the number of comments it has.

As we can see each document, which is an image in our case, has a set of tags, a title, two different image analysis features (described section 5.2), the number of views (on Flickr website) and the number of comments (on Flickr website).

Each query at the image search engine returns N=100 images (or as many as possible found). All the returned images from each topic-query, are ranked by the relevance score assigned by the search engine. Relevance is the quality of results returned from a query, encompassing both what documents are found and their relative ranking (the order that they are returned to the user.) By default, a "TF-IDF" (Targett, 2013) based Scoring Model is used. The basic scoring factors are described in the Appendix. We use a boosted value of the relevance score per image, that is given by our search engine's ranking system. Each document is boosted on indexing time according to its views and the number of comments that it has received on the Flickr website. If the initial relevance score for an image I_i, was $rel(I_i)$, the new boosted relevance is: $rel(I_i) = rel(I_i) * log(comments_i) * log(views_i)$. We believe that an image with a lot of views and comments has more accurate and robust associated tags, that's why we choose to boost its relevancy score a priori. The boosted relevance of each image is normalized in the range of [0,1].

Calculation of similarity

We use the combination of two similarity measures in order to form the similarity for each pair of images. Firstly, we calculate a visual feature similarity which is based on low level features between images. It is formed after combining *ch*, a 64-D color histogram (LAB) and *edh*, a 73-D edge direction histogram to produce one image similarity measure. We denote this visual similarity measure by $vsim = \frac{ch+edh}{2}, vsim \in [0,1]$. Both *ch* and *edh* are calculated using the *L2 norm*. We can see the details of 64-D color histogram (LAB) and 73-D edge direction histogram in section 5.3.

Secondly, for each pair of images, we calculate a textual similarity which is based on the cosine similarity of their tag vectors. As a document we consider the query and as a corpus of documents we consider the whole dataset. This method promotes rare tags but also weakens unimportant tags. First we weight the each tag with tf-idf: $\forall x \in I$, $T_x = set\ of\ tags\ for\ image\ x$, and $tf(g) = \#$ of appearances of tag in T_i. Given a set of images $R = R_1 \cup \ldots \cup R_n$ for topics $T_1, \ldots, T_n$, for some image $x \in R_i$, $T_{t_i} = \bigcup_{x \in R_i} T_x$.

$idf(g) = \log(1/fraction\ of\ images(overall)\ in\ which\ g\ appears)$.

So, $F_x(g) = tf(g) * idf(g)$.

That is, if T_1 and T_2 are two image vectors consisting of the image tags, their cosine similarity or $tsim$ as we name it, is defined as: $tsim = \frac{T_1 {T_2}^T}{|T_1||T_2|}, tsim \in [0,1]$.

The two kinds of similarities $vsim, tsim$ are combined to form a unified similarity measure for pairs of images that belong to C. For $a \in [0,1]$, $sim(I_i, I_j)$ between images I_i and I_j is defined as follows:

$$sim(I_i, I_j) = a * vsim(I_i, I_j) + (1 - a) \cdot tsim(I_i, I_j), i, j \in \{1, \dots, n\}, i \neq j \qquad \text{Eq. 3.1}$$

We should mention that parameter α, acts as a balance between the participation of $vsim$ and $tsim$ in the value of the similarity between each pair of images. Also, $sim(I_i, I_j) \geq 0 \ \forall \ i, j \in \{1, \dots, n\}$ and $sim(I_i, I_j) = sim(I_j, I_i)$ according to our definition. The textual information of an image is subject to human intervention and captures the effort of a person to describe an image efficiently. The visual information is a matter of choosing an image analysis method that captures the user's visual similarity needs. So, we decided to combine them in order to form a similarity measure that promotes both visually similar images and textually similar images.

The Graph Construction

Consider a graph $G = (V, E)$ and let nodes of G denote images, and edge weights denote similarities between images. We construct the graph as follows: Nodes are images. The addition of an edge connecting two nodes $(i, j) \in V$, is dictated by the $sim(i, j)$, after checking if the similarity between these two nodes is above a certain threshold $t \in [0,1]$.

$$Given\ t \in [0,1]\ and\ sim(i, \mathrm{j})\ \forall\ (i, j) \in V, if\ sim(i, j) \geq t\ then\ \{i, j\} \in E$$

If an edge between is formed between $i, j \in V$ on graph G, its weight is set to the value of $sim(I_i, I_j)$. For each node $i \in V$, we set the weight of u at $rel(I_i)$ of the corresponding image $I_i \in C$. For our system implementation, t was set to 0.1, so images with a similarity of less than 0.1 don't form an edge in the graph.

The set selection process

This process is the implementation of each one of the algorithms described in chapter 4. According to the selected algorithm, the process receives the pool of images with their assigned pairwise similarities as input, and produces a subset of k images as output. This set is presented to the user. Each algorithm, selects k images per topic.

5.3. Image Dataset

In this section, we present the characteristics of the real world image dataset that we used in order to experiment on, and evaluate our system.

We used the NUS-WIDE (ARe15) web image dataset created by the Lab for Media Search of the National University of Singapore (NUS) (T. Chua, 2009). To our knowledge, this is the largest real-world web image dataset comprising almost 270,000 images. All the characteristics of the dataset are summarized in Table 5.1 Image dataset characteristics. Specifically, it consists of 269,648 images and the associated tags from Flickr (Fli15), with a total of 5,018 unique tags. Also, we removed noise like camera model variations, comprising of about 300 tags of the original set of tags. The average number of tags per image is 18.35. The most tags on an image are 632. In Table 5.2 Some of the most frequent tags we can see a set of the most frequent tags of the image dataset and in we can see some of the least frequent tags.

Finally, we used two types of low-level features extracted from the dataset images, namely the 64-D color histogram and 73-D edge direction histogram in combination to produce one image similarity metric.

Table 5.1 Image dataset characteristics.

Images	**Unique tags**	**Avg tags per image**	**Most tags on an image**	**2 Low level features**	**Others per image**
269.648	5.018	18.32	632	Color Histogram, Edge direction histogram	Title # Comments # Views

64-D color histogram (LAB) (L. G. Shapiro, 2003) : The LAB color space image histogram represents the color content of an image. “It is defined as the distribution of the number of pixels for each bin“ (T. Chua, 2009). LAB stands for lightness (L), and color components (A,B). It is a linear color space, so the authors of (T. Chua, 2009) quantized each component of LAB color space uniformly into four bins.

73-D edge direction histogram (D. K. Park, 2000) : The edge direction histogram encodes the distribution of the direction of edges. “It comprises a total of 73 bins, in which the first 72 bins are the count of edges with directions quantized at five degrees interval, and the last bin is the count of number of pixels that do not contribute to an edge.” (T. Chua, 2009).

Table 5.2 Some of the most frequent tags.

Nature	19657	**Clouds**	14201	**Sunset**	10195
Sky	17329	**Red**	13172	**Light**	10115
Blue	16519	**Green**	12262	**White**	9444
Water	17646	**Bravo**	11871	**People**	7437

Table 5.3. The frequencies of some of the tags that we removed.

Abigfave	25218	**Anawesomeshot**	16519	**Soe**	9301
Aplusphoto	17122	**Bravo**	11871	**Bw**	8788
DiamondClassPhotographer	17052	**Flickrdiamond**	10725	**Goldstaraward**	6564

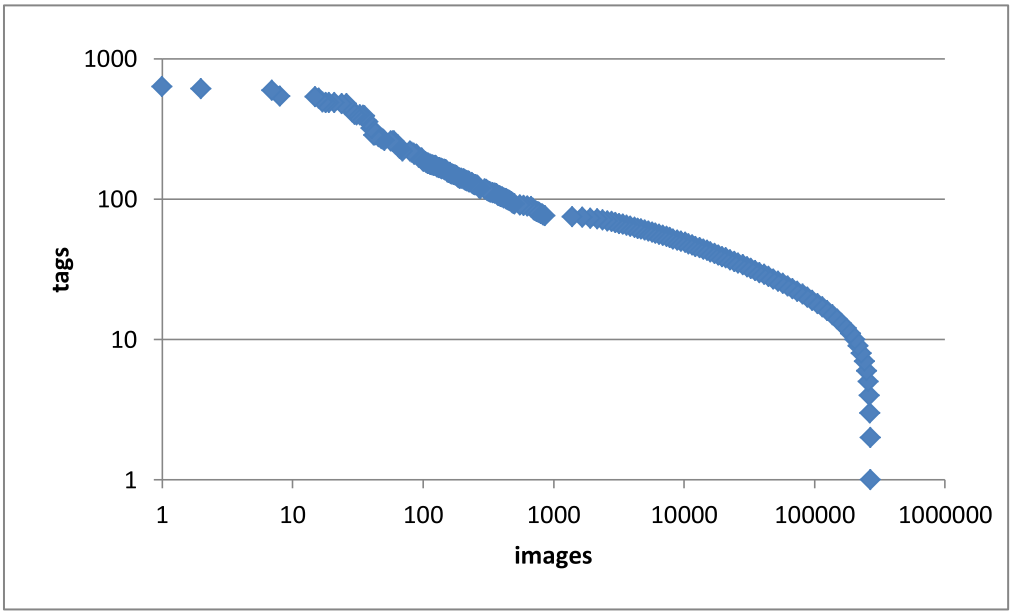

Figure 5.1 Distribution of the number of at least k tags per image, with log log scale.

As we can see in Figure 5.1 Distribution of the number of at least k tags per image, with log log scale., very few images have the most tags and the most images have a smaller number of tags.

Image dataset processing

First of all we removed about 300 tags that were plain noise in each image's tag list, such as camera model variations and photography techniques specifics. Also, we extended the NUS-WIDE (ARe15) dataset by adding the title, number of views and number of comments for each image. In order to accomplish this task, we used the Flickr API (Fli15) and specifically the *flickr.photos.getInfo* method.

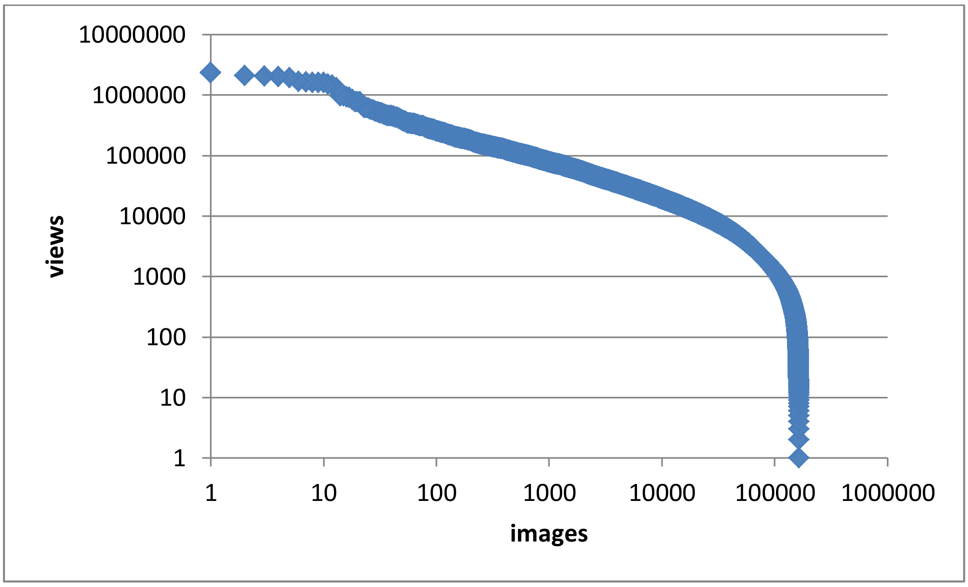

Figure 5.2 Distribution of the number of at least k views per image, with log log scale.

The title was used for image retrieval through Solr. Also, as we already mentioned the comments and views of each image in Flickr, were used as a boost for the relevance score of each image. We can see the distributions of comments and views along the dataset images in figure 5.4 and 5.3.

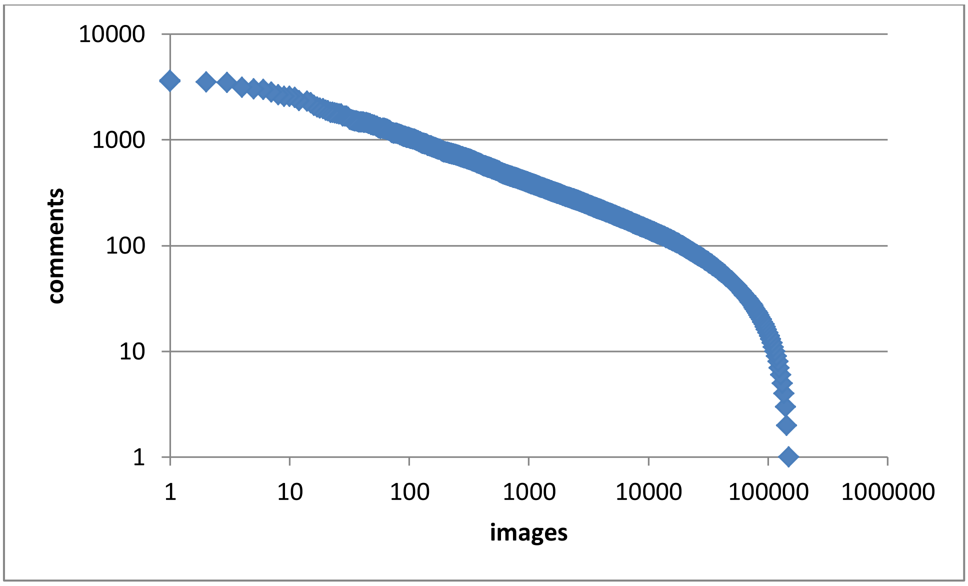

Figure 5.3 Distribution of the number of at least k comments per image, with double log scale.

5.4. Experimental Setup

In order to evaluate the results of our algorithms in generating size-k sets of homogeneous and relevant images, we used a method based on human judges or evaluators as we will call them, that was driven by a "set of topics search" use case. The goals of the evaluations included:

- Determining if our algorithms perform better than the trivial (relevance-based) baseline algorithm.
- Determining the impact of low-level visual features (color histogram) versus textual features (tags) on the performance of our algorithms by trying different values for parameter α.
- Assessing the performance of our algorithms for various sizes of $k, k \in \{1,2,3\}$.

We should mention that it is quite challenging to quantify the quality (or difference of performance) of sets of image search results from our algorithms for a series of good reasons. First of all, a user's preference towards a certain image is profoundly biased by his personal tastes and influences. Secondly, asking a user to compare the quality of a set of images is a difficult task. For example a user may find it hard to choose between set A, which has some "appropriate" images and set B, which is mixed with "appropriate" and "bad" images.

Queries that we used

Typical queries included travel-related destinations as they are presented in Table 5.4. All the queries were considered to be travel-related popular destinations and they were chosen carefully through Wikipedia (Wik15). During the query selection process we went on the Wikipedia culture/tourism/travel text segment of each destination, which is as a whole text is a very good description of travel-related information about each destination. From there, we extracted three or four representative topics. For each destination, the list of three or four topics that was selected formed the query that we used in our experiments. It was rather difficult to build twenty queries, as for each topic per query we had to make sure that there are enough images in our dataset. In the next table, we can see the twenty Text Themes that we managed to construct. From every text theme of a topic we construct a query. For each query, we made sure that there are enough images in our digital image dataset from Flickr.

Table 5.4 The topics that formed each query.

Query		Topic 1	Topic 2	Topic 3	Topic 4
1	**Paris**	eiffel tower	louvre	notre dame	arc triomphe
2	**Cairo**	river nile	citadel cairo	pyramids	-
3	**Rome**	st peters square	colosseum	pantheon	-
4	**Santorini**	fira	caldera	oia	-
5	**Istanbul**	aya sofia	grand bazaar	blue mosque	-
6	**New York**	times square	brooklyn bridge	statue liberty	empire state building
7	**Madrid**	palacio real	plaza mayor	almudena cathedral	debod
8	**London**	london tower	bus	Big ben	london eye
9	**Delhi, India**	taj mahal	humayun	jama masjid	-
10	**Rome**	vatican museum	trevi	catacombs	-
11	**Barcelona**	sagrada familia	guell	torre agbar	-
12	**San Francisco**	golden gate bridge	alcatraz	transamerica pyramid	-
13	**Los Angeles**	walt disney concert hall	walk of fame	chinatown	-
14	**Washington**	capitol	white house	washington Monument	-
15	**Moscow**	christ cathedral	basil	redsquare	-
16	**Tokyo**	sensoji	sibuya	tokyo palace	tokyo tower

17	**Sydney**	harbor bridge	opera house	chinatown	-
18	**Beijing**	forbidden city	summer palace	greatwall	tiananmen
19	**Florence**	tower pisa	duomo	palazzo vecchio	-
20	**Berlin**	reichstag	berlin gate	fernsehturm	-

5.4.1. User study on mixed set of images per algorithm

This study is designed to measure the success rate for each one of our algorithms separately. For this experiment, we mixed the results returned from each algorithm, with the top relevant results of the trivial (baseline) algorithm described in section 4.2, in one set. The top relevant results are obtained from ranked by relevance images returned by the baseline algorithm without any further intervention. For each query described in section 6.1, we presented to the user a corresponding piece of text. Then, we asked him to (1) read it carefully and for each on of the topics that it refers to, (2) choose k images, with $k \in \{1,2,3\}$. One basic guideline for choosing was that the final set of chosen images should be homogeneous in appearance (colors, textures) and content (is relevant and describes the actual topic). In this experiment, ten professional users in the areas of branding/corporate identity and graphic arts participated. They can be marked as experts, according to their experience in the field of imagery. Each one did a total of twenty evaluations for each $k, k \in \{1,2,3\}$. They did one per each query as they are described in table 6.2., so the total number of evaluations per user is sixty.

There are two interesting points about this study. First, we did not ask the users to compare two sets of images since, as we mentioned earlier, this is an arduous task. Instead, the user was asked to examine each image per topic individually and the final set of chosen images as a whole. Second, we did not give any indication of ranking or category (which algorithm produced each image) to the user, thereby alleviating the burden of analyzing image ordering or biased labeling. The position of each image in the sets of images that we showed per experiment was shuffled randomly for every user.

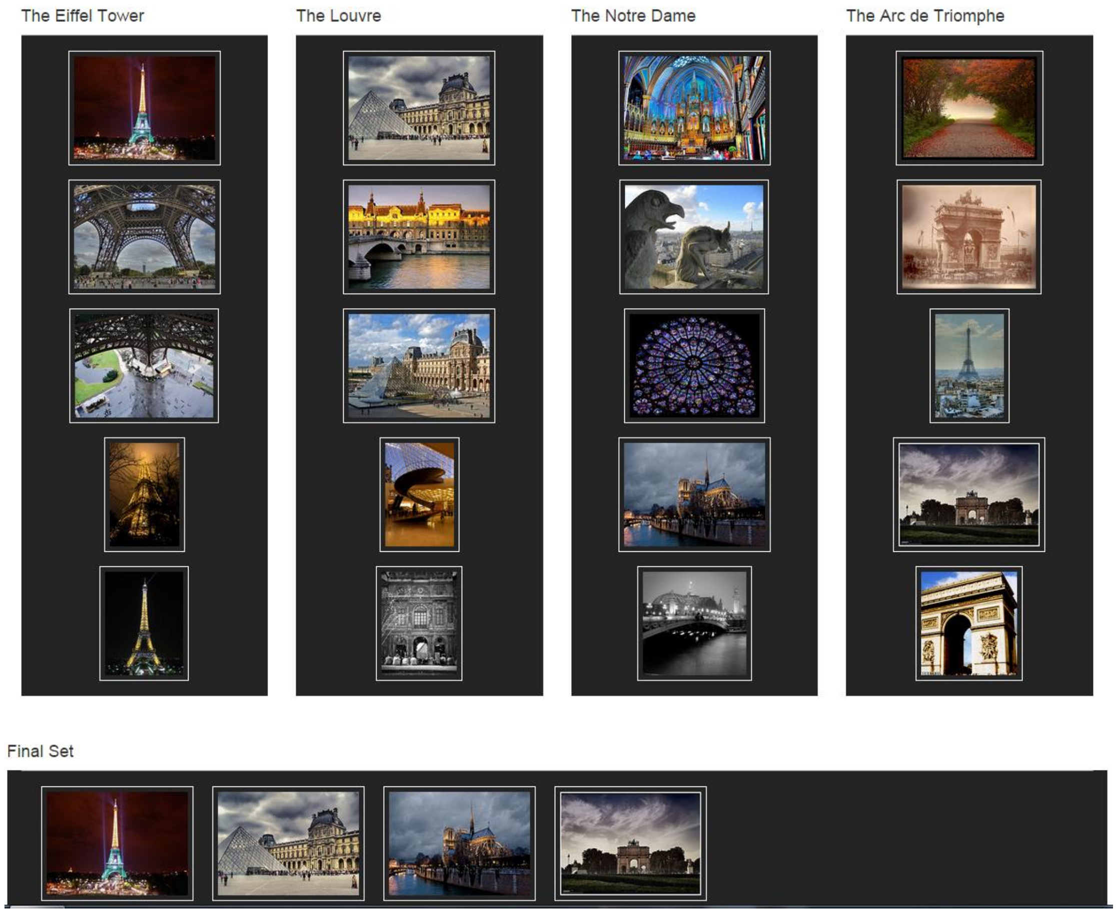

Figure 5.4 A part of the evaluation contents.

As we can see in the previous figure, we present four sets of images, according to the topic to the evaluator, and he chooses one image per set ($k = 1$). The Final set of images is submitted as the result of his evaluation. Before this segment of the evaluation there are specific guidelines and the topics accompanying text and description.

The evaluation guidelines were:

- Read the given piece of text carefully.
- For each topic (topics are bolded in the text), choose <k > image that you believe is (1) relevant to it and (2) homogeneous (with the already selected) as a set.
- The goal is to produce a set of <$n \cdot k$ > relevant and homogeneous images. The final set of four images should be relevant to each topic and homogeneous as a set towards the appearance(colors) and content(keywords). The chosen image per topic should show the actual content of the topic it belongs to.

- If you are not familiar with any of the topics or content of an image, please read the tags by hovering on the image or follow the links to Google Images and Wikipedia for further consultation.
- You can select an image by clicking on it.
- You can un-select an image by clicking on it.
- Image ordering of the set of images that you choose from is random.

If you don't understand any of the above guidelines please do not submit the evaluation and contact : nchalias@cs.uoi.gr

5.4.2. Comparison with a Gold Standard Dataset

For this study, we chose to measure the success rate of our algorithms with a *gold standard set* as a *benchmark*. For our case, this set of images created by users, can be referred to as the best available under reasonable conditions, such as the subjectivity factor and the experience of the person that builds it.

This study was designed to measure the success rate for each one of our algorithms separately. We asked from professional users to construct a set that would stand as a golden standard benchmarking set for each one of the queries. The process of building the baseline set for each query can be summarized as follows:

- The users performed the query on the search engine and it returned a set of 100 images (at most) for each one of the topics.
- Then, they had to choose $k, k \in \{1,2,3\}$ images per topic so that the final set of images would be homogeneous and relevant.
- They worked as a team and jointly agreed for each image of the final set.

The three users that participated are professionals in the areas of branding/corporate identity and graphic arts. They can be marked as experts, according to their experience in the field of imagery.

Success Rate metric

In order to quantify and compare the performance of our algorithms for each experiment, we calculated the *Success Rate* metric. It is a good metric in order to have a comparison measure between them. For each query $q = \{T_1, \dots, T_n\}$, every algorithm A produces a set $R = \{R_1, \dots R_n,\}$, where $|R_i| = k$, and $R_A = R_1 \cup \dots \cup R_n$. Each user selects a set $U = U_1 \cup \dots \cup U_n$ of images. The *Success Rate* for algorithm A and query

q is defined as: $Success\_Rate(A, q) = \frac{R_A \cap U}{n \cdot k}$. This is done for every user that participated in the experiment, so we calculate the average: *Success Rate* = $\sum_{i=1}^{N_u} \frac{Success\_Rate(A,q)}{N_u}$, where N_u is the number of users.

Baselines that we used

We used three different approaches as baselines. Firstly we used the *Relevance Baseline*, or RB, which for a given k and a query $q = \{T_1, \ldots, T_n\}$ as input, returns the k most relevant images per topic $T_i \in q$.

Secondly, we experimented with two *density baselines*, that do not take the relevance score into account. The first one, *k-Densest Greedy (DB) is* a version of k-Densest GD algorithm that is described in section 4.8. The second one is an unbiased by the relevance version of the HITS R algorithm, the HITS algorithm that is described in section 4.8. This algorithm, doesn’t use relevance as a bias when it’s hubs and authorities pass and collect their scores.

5.5. Experimental Results

In this section we present the experimental results, we indicate the worth mentioning observations and points of interest. Also, we point out which algorithm behaves better and for what values of parameters.

For briefness and avoiding the use of long names, we provide a name mapping in table 5.4, that we will use for the rest of this chapter, for shortening the lengthy names of our algorithms.

Table 5.5. Name mapping of algorithms used in chapter 6.

Algorithm Name	Brief Description
Relevance	Relevance Based Baseline Algorithm
k-Densest Greedy (DB)	k-Densest Subgraph Greedy, Density Based
HITS	HITS Algorithm
Greedy	Greedy Algorithm
k-Densest Greedy	k-Densest Subgraph Greedy Algorithm
Layered Greedy	Layered Greedy Algorithm
HITS-R	Biased HITS Algorithm
Densest Greedy	Densest Subgraph Greedy Algorithm

5.5.1. Parameter α calibration

In this section we explain how we calibrated the parameters a for our system. As we mentioned in chapter 4, parameter $\alpha, \alpha \in [0,1]$ balances the visual and textual participation in the similarity measure between images.

We run our algorithms with five values for $a, \alpha = \{0, 0.25, 0.5, 0.75, 1\}$, and documented the average success rate for each one of them. As we observe, for all values of k, the algorithms seem to behave well for $\alpha = 0.5$ and for $\alpha = 0.25$. Both of these values produce good success rates. As we observe in figures 5.6-5.8, the performance of the k-Densest Greedy algorithm is better than all others for $k = 1$, and similar with HITS-R for $k = 2$ and $k = 3$. Thus, we decided to calibrate the value of a to $a = 0.5$. For all the comparisons from now on, we will use this value for our algorithms.

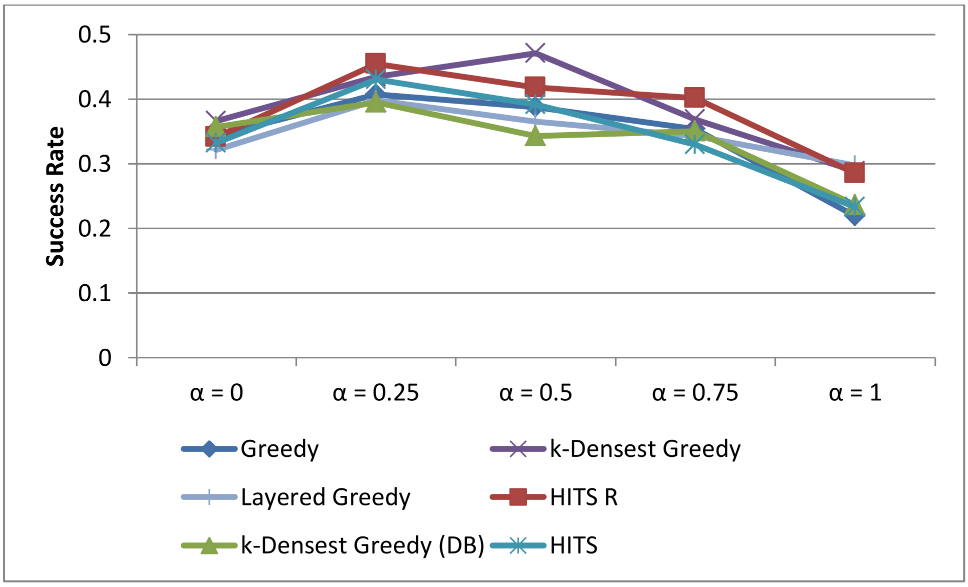

Figure 5.5. The success rate for all algorithms and values of α for k=1.

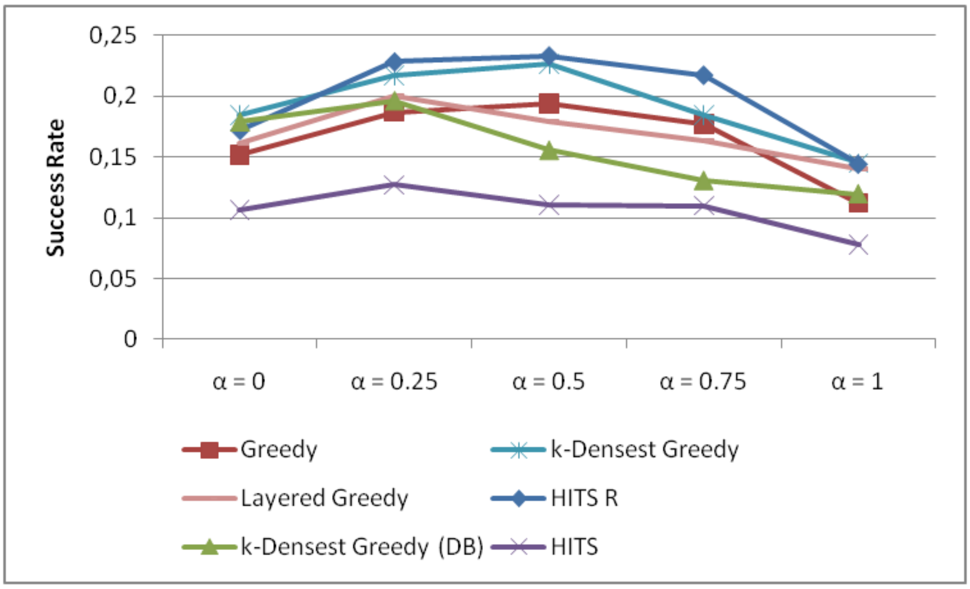

Figure 5.6. The success rate for all algorithms and values of α for k=2.

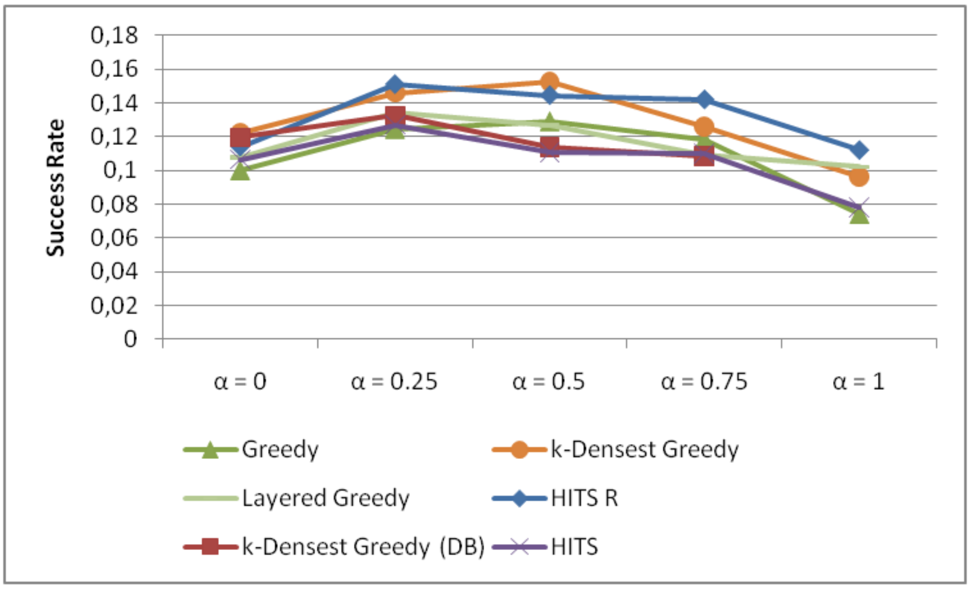

Figure 5.7. The success rate for all algorithms and values of α for k=3.

5.5.2. Correcting the ambiguity & broadness of query terms

In many cases, the selection process corrects errors related to the ambiguity or broadness of the query terms. As we mentioned in the introduction, the time for a user to search and construct a set of proper images for a text is expanding along with the sizes of digital image libraries. So, removing such errors from the results is very important when a user is trying to find images that illustrate a text. Also, this correction of errors from our algorithms indicates visually the need for a method that doesn't use only the relevance score of an image and supports the experimental results that showed a major improvement in the success rate between our algorithm and relevance-based baselines.

We present an example with query 4 of table 5.5 (Santorini { fira, caldera, oia }), for $k = 2$ where the trivial algorithm returns the results seen on the next figure [6.2]. As we can see in the results, it picks fira and oia correctly, but the two caldera images that it chooses are from the national park of Oregon. This makes them irrelevant to the text of query 4.

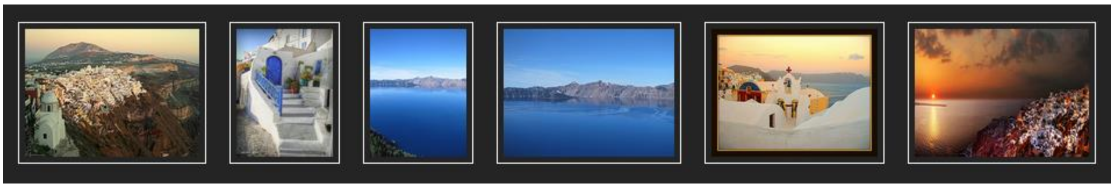

Figure 5.8 Baseline algorithm results for query 4 of table 5.5

On figure 6.12 we see the results of the Greedy algorithm for $k = 2, a = 0.5$. As we can see, it chooses images from the three topics {fira, caldera, oia} and among the choices is the correct caldera.

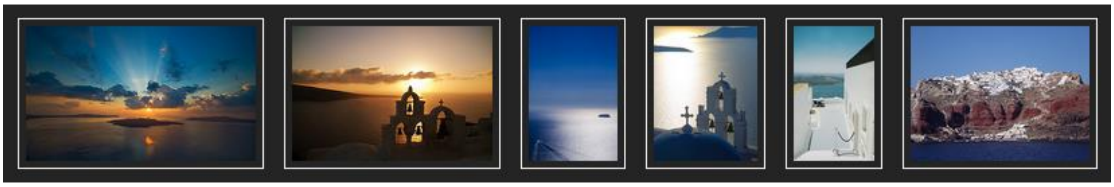

Figure 5.9 Greedy algorithm results for query 4 of table 5.5

Another such example can be seen on figure 6.4 for query 14 of table 6.2 for Washington {capitol, white house, Washington monument}, $k = 1$. As we can see, the algorithm chooses two images for the capitol and Washington monument, but it fails to choose a correct image for the white house.

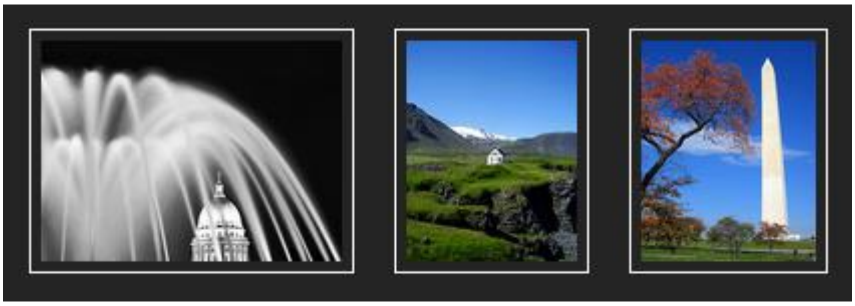

Figure 5.10 Baseline algorithm results for query 14 of table 5.5

On figure 6.14, we see the results of the HITS algorithm for $k = 1, a = 0.5$. For the white house topic, it chooses an image from the actual white house. We should mention here, that this query for the topic of white house was very noisy on its most relevant results, and the image selected by the HITS algorithm was ranked 44th from the image search engine.

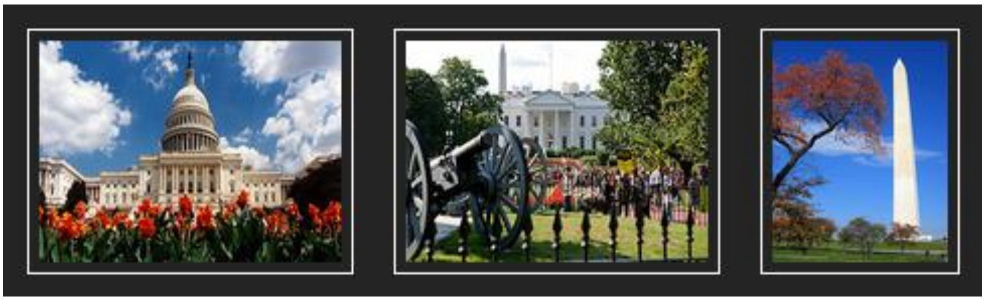

Figure 5.11 HITS algorithm results for query 14 of table 5.5

The pairwise similarity described in chapter 5 and uses both visual and textual features. So, images across topics that share common tags will have a higher textual similarity than other ones that don't share any common tags. For the example shown in figure 6.12, the images of the caldera at the national park of Oregon share only the word caldera with other images from the sets of Fira and Oia. The Santorini Caldera images that are ranked further lower in the results share many more tags with other images from the sets of Fira and Oia. Thus, the notion that these images belong to the text that describes the Fira, Caldera and Oia of Santorini is captured.

5.5.3. Algorithms performance for the user study on mixed set of images

In this section we will compare the performance of our algorithms for the calibrated parameters $a = 0.5$ and for various values of $k, k = \{1, 2, 3\}$ and the user study on mixed set of images experiment. In figure 5.12, we present the comparison of all algorithms for various values of k.

A first observation we can make from figure 5.12, is that as the size of k increases, the success rate of all algorithms falls. As the user can choose more than one image per topic, it seems more difficult for each algorithm to cover the second or third choice of the user. As the users that participated in the experiment told us, the set of images that they chose for $k = 2$, in some cases was different than the one for $k = 1$. So, they didn't just add one image per topic on their initial selection for $k = 1$. They pointed out that they found it more difficult to choose images when k increased to 3, especially in queries of four topics, where they had to build sets of twelve images.

Another observation that we can make, is that the k-Densest Greedy algorithm performs better over all other ones for $k = 1$, and $k = 3$. For $k = 2$, it is slightly worse than HITS-R but the difference is very small. The k-Densest Greedy algorithm captures the user choices on an average of 47% of the cases for $k = 1$. The Greedy and Layered Greedy algorithm perform worse than the k-Densest Greedy and HITS-R algorithms as k increases, and their results are rather similar.

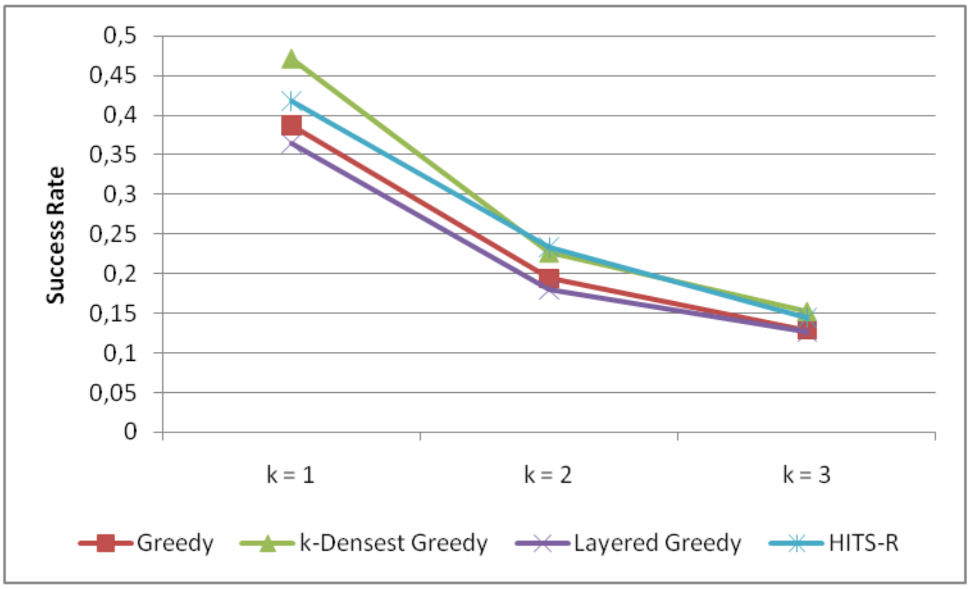

Figure 5.12. Algorithm comparison for various k and a=0.5.

In figure 5.13 we present the comparison of the k-Densest GD algorithm that performed better than our other algorithms with the algorithms that we used as baselines. The two algorithms that don't use the relevance in the score, and the algorithm that uses only the relevance. As we can see, the performance of the k-Densest GD algorithm is better than all other three for all values of k. The HITS algorithm performs rather well in comparison to the Relevance algorithm. The same is observed for the k-Densest Greedy (DB) algorithm, which is better than the Baseline algorithm for all values of k.

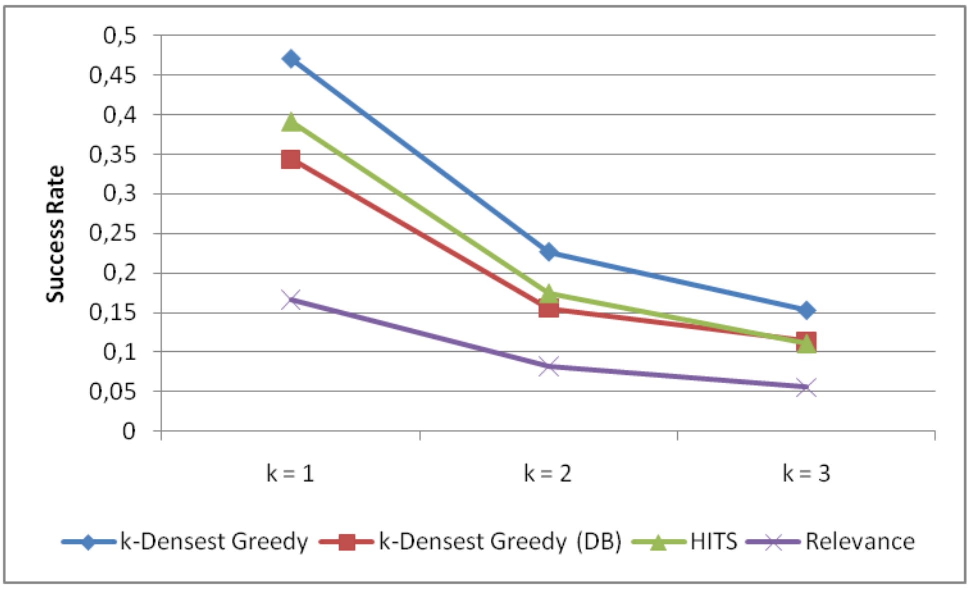

Figure 5.13. Success rate of Baseline algorithms and the k-Densest GD.

5.5.4. Algorithms performance for the comparison with a gold standard

In this section we will compare the performance of our algorithms with the gold standard set. In figure 5.11 we present the comparison of all algorithms for $a = 0$ and various values of k.

A first observation we can make, is that as the size of k increases, the success rate of all algorithms falls. The comment that we can make on this behavior is the same as for the previous experiment. We can see, the k-Densest Greedy algorithm stands out, having a better performance than all other three. Its performance for $k = 1$ and $k = 3$ is very close to that of HITS-R algorithm. The other two algorithms, Greedy and Layered Greedy perform worse, but rather close for $k = 2$, $k = 3$. For $k = 1$, the Layered Greedy is better than Greedy.

Another observation that we can make, is that the k-Densest Greedy algorithm performs better over all other ones for all values of k. The k-Densest GD algorithm best captures the user choices on an average of 29% of the case for $k = 1$.

In figure 6.14 we present the comparison of all algorithms for various values of k.

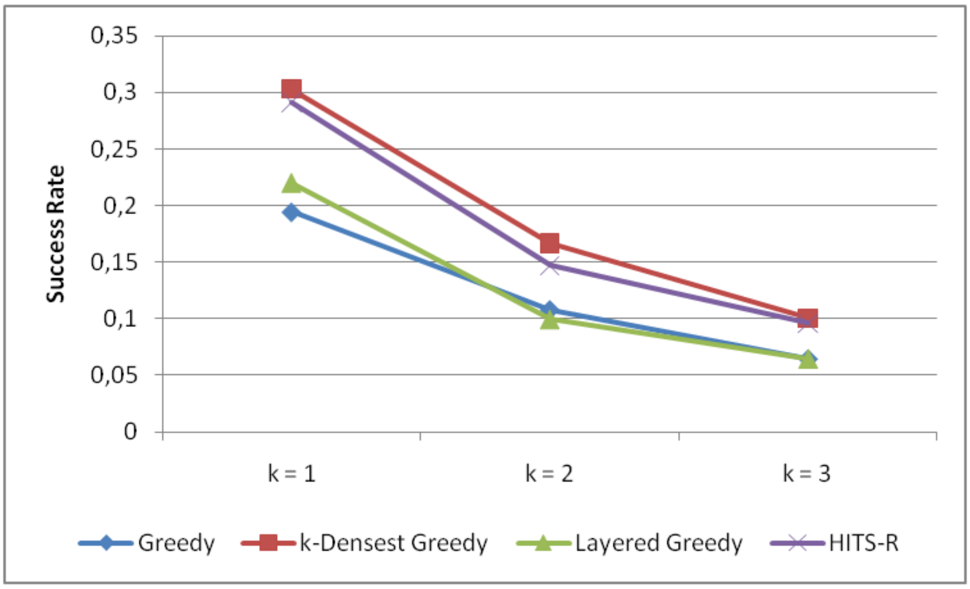

Figure 5.14. Algorithm comparison for various k, α=0.5.

In figure 5.15 we present the comparison of the k-Densest Greedy algorithm that performed better than our other algorithms with the algorithms that we used as baselines. The two algorithms that don't use the relevance in the score, and the algorithm that uses only the relevance. As we can see, the performance of the k-Densest Greedy algorithm is again better than all other three, for all values of k. The HITS algorithm and the k-Densest Greedy (BS) algorithm perform worse than the k-Densest Greedy, yet rather well in comparison to the Relevance algorithm.

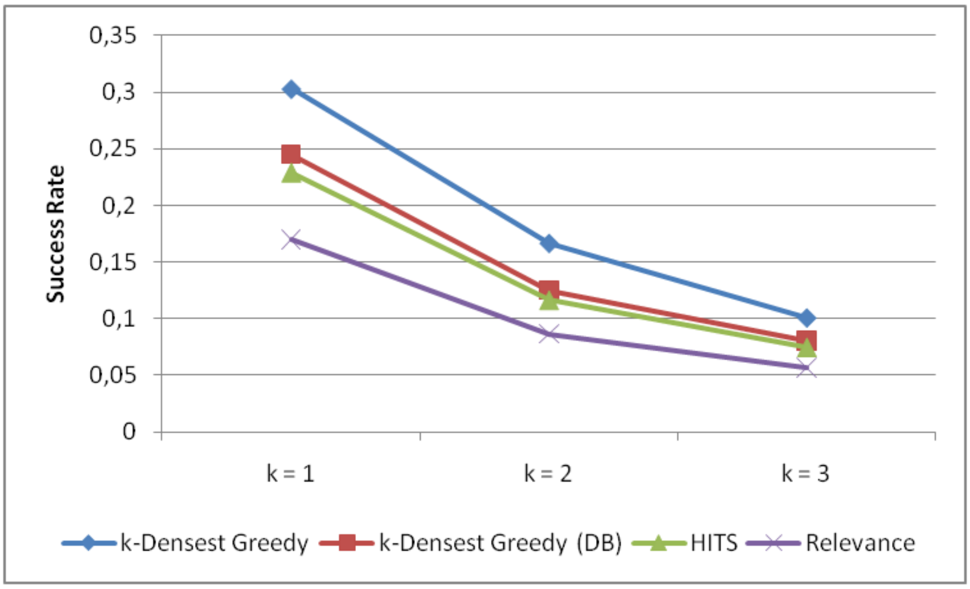

Figure 5.15. Success rate of Baseline algorithms and the k-Densest GD.

5.6. Additional Experiments with an Interactive User Selection Simulation

For this study, we chose to measure the success rate of the Greedy algorithm by comparing the sets of images that it would build by using the first image choice of a user, instead of the first one that it chooses. In order to conduct this experiment we did the following:

- Given the first image that a user selected in experiment 1 as input to the GD algorithm, and force it to insert it as the first image in its set of images.
- The rest of the selection process for the Greedy algorithm remains intact, as it is described in section 4.3.
- The Greedy algorithm returns a set of images as its output.

If we have N_u users, and each user did a series of $|E|$ evaluations, with $E = \{E_1, \dots, E_Q\}$, then Greedy algorithm run $N_u \cdot |E|$ times and produced the same number of sets. The characteristic of each one of these sets is that the first image that was inserted by the Greedy algorithm, was the first choice of each user for that experiment. The value of parameter α, was calibrated to α=0.5.

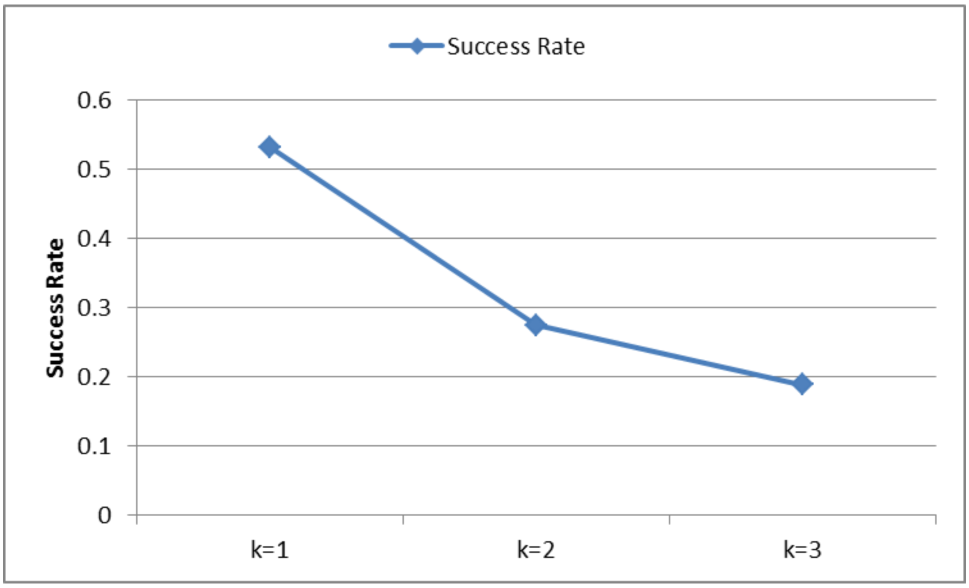

Figure 5.16 Success Rate of the Greedy Algorithm with an Interactive User Selection Simulation

As we can see on figure 5.16, the success rate for k=1 is 0.53 which means that on average this process finds half of the images that a user found. As the value of k grows, the success rate falls. This happens because the range of available images for a user to choose is larger.

5.7. Additional Experiment with the Densest Subgraph Greedy Algorithm

This experiment was used to observe the behavior of an algorithm that does not take k into account. This algorithm, which is described in section 4.7, produces a dense set with the maximum average score. We run the *Densest Subgraph Greedy* algorithm for all the available queries and gathered the results. Then, we measured the average uniformity of distribution of its results across topics. In order to do this, we calculated the average Shannon Entropy for each value α and for *text themes* with three or four topics.

If the algorithm assigns images to each topic, then we say that the probabilities of occurrence for each one of the topics are $p_1, p_2, \dots, p_n$.

$$H(X) = \sum_{x \in X} p(x) \cdot log\, p(x) \qquad \text{Eq. 5.1}$$

In equation 5.1, we set the base of the logarithm to 2.

That may give us an insight of the uniformity of distribution, or how uniformly the algorithm assigns images across topics. The uniform distribution maximizes the entropy, so the maximum entropy is $log_2(k)$ for our case.

Table 5.6. Densest Subgraph Greedy Algorithm observations.

Number of topics	**Normalized Avg. Entropy**	**Max. images of a topic**	**Min. images of a topic**	**Max. images (total)**	**Min images (total)**
n **= 3**	0.902407	78	0	163	12
n **= 4**	0.496834	89	0	220	11

The normalized average entropy for thee topics indicates that the distribution of images per topic is quite uniform. Instead, the normalized average entropy for four topics, indicates that the distribution is not uniform. Also, there was a case that it chose 163/300 images from all topics, for a query with three topics and many cases where it didn’t choose any images at all from a topic. Furthermore, for the four topics, there was a case that it chose 220/400 images in total, and a case where it chose 89 images from a single topic.

A conclusion that we can make from this observation, is that the Densest Subgraph Greedy algorithm for α=0.5 distributes the images per topic quite uniformly on average. Also, as the number of images per topic increases to four, the distribution of images per topic becomes non-uniform, on average.

5.8. Summary

To ensure our algorithms work in practice, we conducted the experiments with real world and user annotated images directly from the popular image platform Flickr (see section 5.3), and we used professional users as judges and asked them to evaluate the results of our algorithms.

In order to measure the performance of our system algorithms, we defined a success rate metric that indicated the average approval of the users towards the sets of images that they produced.

Also, through the experimental evaluation, we managed to calibrate the parameter α and observed how our algorithms behave for various values of k. We compared our algorithms with relevance baselines and density baselines.

Our algorithms perform considerably better than the relevance baseline, indicating that using only the relevance score in not enough to produce acceptable results. Also, our algorithms perform better than the two density baselines that we tested, indicating that the relevance value participation in the scoring function is important.

The value of parameter α has a significant impact on the performance of algorithms. Various values of α produce different success rates for all algorithms in both experimental evaluations that we did.

Surprisingly, the HITS-R algorithm produced very good and acceptable results, in comparison to the k-Densest Subgraph Greedy algorithm.

CHAPTER 6. CONCLUSION

6.1. Summary

Text illustration with images is a problem of great importance that arises in several applications. In this Thesis we proposed the *TEXTILLE* system that aims to facilitate and automate this process, providing a complete solution to the problem. We proposed an end-to-end system design for the problem with several components. We formulated the image selection process as an optimization problem, where the goal is to select a set of images that maximize a score that combines relevance and homogeneity of the images. Based on a connection with graph theoretic problems that we recognized, we proved that our problem is NP-hard, and we proposed a series of algorithms.

We evaluated our system algorithms on a large collection of real world images collected from Flickr, using travel-related query-topics. We did experiments with professional users in the fields of branding/corporate identity and graphic arts. Experiments demonstrated that only taking the relevance into account, is not enough to produce good results. Also, not taking relevance into account at all, again is not enough to produce good results. The combination of both similarity and relevance produced the most acceptable results. Finally, in many cases the image selection process corrected errors related to the ambiguity or broadness of the query terms.

6.2. Future Work and Extensions

This work is a first step towards building automated tools that will assist the search and selection process. In the future we would be interested to pursue the following directions:

- It would be interesting to study our system for other application areas beyond tourism. For example using queries for animal species, or queries for branding and advertisements. Different application areas may behave in different ways.
- It would be interesting to evaluate our system and our algorithms in terms of efficiency. In very large image databases, some heuristics may be necessary for pruning the set of images to be considered. Our problem could have some interesting connections with rank aggregation algorithms.
- In order to complete the TEXTILLE design, we would like to investigate automated techniques for extracting topics from the user text to give as input to the topic retrieval process of the system.
- It would be interesting to consider other applications of the selection algorithms on application domains beyond images. For example our techniques could be used for products, where a user is searching for a particular set of products such as furniture for her house that are relevant to the queries (the type of furniture she is looking for), and at the same time they are homogeneous in style. We could also consider an online learning application where a student, that is searching for a set of courses to attend needs a course selection that is relevant to her needs, but also thematically related (e.g., programming assignments are all in the same programming language). In general our methodology can be applied to any domain where we have relevance in one space, and a notion of similarity in a different space.

BIBLIOGRAPHY

A Real-World Web Image Database from National University of Singapore. *NUS-WIDE.* [Online] [Cited: 03 05, 2015.] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.

A. Bhaskara, M., Charikar, E. Chlamtac, U. Feige. 2010. Detecting High Log-Densities – an O(n^1/4) Approximation for Densest k-Subgraph. *STOC '10 Proceedings of the forty-second ACM symposium on Theory of computing.* 2010, pp. 201-210.

A. Bialecki, R. Muir, G. Ingersoll. 2012. Apache Lucene 4. *SIGIR 2012 Workshop on Open Source Information Retrieval.* 2012.

A. Shrivastava, T. Malisiewicz, A. Gupta, and A. Efros. 2011. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics.* 2011, Vol. 30(6), pp. 54:1–154:10.

A. Vailaya, M.A.T. Figueiredo, A.K. Jain and H.J. Zhang. 2001. Image classification for Content-Based Indexing. *IEEE Trans. Image Processing.* 2001, pp. 10(1):117-130.

A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain. 2000. Content Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2000, Vol. 22, 12, pp. 1349-1380.

A. Yanagawa, S. Chang, L. Kennedy, and W. Hsu. 2007. Columbia University's Baseline Detectors for 374 LSCOM Semantic Visual Concepts. *Technical report, Columbia University*. 2007.

B. Coyne, R. Sproat. 2001. WordsEye: An automatic text-to-scene conversion system. *Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques.* 2001, pp. 487-496.

B. S. Manjunath, W. -Y. Ma. 1996. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* August 1996, Vol. 18, 8, pp. 834-842.

C. Frankel, M. Swain, V. Athitsos. 1996. WebSeer: An image search engine for the world wide web. *TR-96-14.* 1996.

Chang, D. Zhang and S. 2004. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. *In Proceedings of the 12th Annual ACM International Conference on Multimedia.* 2004, pp. 877–884.

D. C. Brown, B. Chandrasekaran. 1981. Design Considerations for Picture Production in a Natural Language Graphics System. *ACM SIGGRAPH Computer Graphics.* 1981, pp. 174-207.

D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. 2004. 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.* 2004, pp. 952-959.

D. K. Park, Y. S. Leon and C. S. Won. 2000. Efficient use of localedge histogram descriptor. *ACM Multimedia.* 2000.

D. Manning, P. Raghavan and H. Schütze. 2008. *Introduction to Information Retrieval.* Cambridge : Cambridge University Press, 2008.

E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. 2004. Multiresolution Histograms and their Use for Texture Classification. *IEEE Trans. Pattern Analysis and Machine Intelligence.* 2004, Vol. 26(7), pp. 831-847.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* MA : MIT Press, Cambridge, 1998.

Flickr Images. *Flickr.* [Online] [Cited: 03 01, 2015.] https://www.flickr.com/.

Foundation, Apache Software. 2013. Apache Lucene Class Similarity. [Online] Apache Software Foundation, 2013. http://lucene.apache.org/core/3_5_0/api/core/org/apache/lucene/search/Similarity.html.

G. Adorni, M. Di Manzo and F. Giunchiglia. 1984. Natural Language Driven Image Generation. *COLING 84.* 1984, pp. 495–500.

G. Goel, C. Karande, P. Tripathi, L. Wang. 2009. Approximability of combinatorial problems with multiagent submodular cost functions. *In Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science.* 2009, pp. 755–764.

G. Guo, G. Xu, H. Li and X. Cheng. 2008. A unified and discriminative model for query refinement. *SIGIR.* 2008.

G. Kumaran, V. R. Carvalho. 2009. Reducing long queries using query quality predictors. *SIGIR.* 2009.

Goldstein, J. G. Carbonell and J. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. *In SIGIR.* 1998, pp. 335-336.

J. Pattillo, A. Veremyevb, S. Butenkoc, V. Boginskib. 2013. On the maximum quasi-clique problem. *Discrete Applied Mathematics.* 2013, Vol. 161, 1-2, pp. 244-257.

J. Wang, J. Zhu. 2009. Portfolio theory of information retrieval. *In SIGIR.* 2009, pp. 115-122.

J.Z. Wang, J. Li and G. Wiederhold. 2001. SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. *IEEE Trans. Pattern Analysis and Machine Intelligence.* 2001, pp. 23(9), 947–963.

K. Barnard, P. Duygulu, D. Forsyth, N. de. Freitas, D. M. Blei and M. I. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research, volume 3.* 2003.

K. Nagano, Y. Kawahara , K. Aihara. 2011. Size-constrained Submodular Minimization through Minimum Norm Base. *In Proceedings of the 28 th International Conference on Machine Learning.* 2011.

K. Nagano, Y. Kawahara, K. Aihara. 2011. Size-constrained Minimization through Minimum Norm Base. *In Proc. 28th International Conference on Machine Learning.* 2011.

K. Yang, M. Wang, X.-S. Hua and H.-J Zhang. 2010. Social image search with diverse relevance ranking. *n: International MultiMedia Modeling Conference (MMM).* 2010.

Kleinberg, J. M. 1998. Authoritative Sources in a Hyperlinked Environment. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms.* 1998.

L. G. Shapiro, G.C. Stockman. 2003. *Computer Vision.* s.l. : Prentice Hall, 2003.

L. Kennedy, M. Slaney, K. Weinberger. 2009. Reliable tags using image similarity: mining specificity and expertise from large-scale multimedia databases. *In: WSMC '09: Proceedings of the 1st Workshop on Web-scale Multimedia Corpus.* 2009, pp. 17–24.

L. Li, Y. Shang, W. Zhang. 2002. Improvement of HITS-based Algorithms on Web Documents. *WWW2002.* 2002.

Li J., Wang J. 2008. Real-time computerized annotation of pictures. *IEEE Trans. Pattern Anal. Mach. Intell.* 2008, pp. 30(6), 985–1002.

Li X.R., Snoek C.G.M., Worring M. 2008. Learning tag relevance by neighbor voting for social image retrieval. *In: Proceedings of MIR.* 2008, pp. 180–187.

Liu D., Wang M., Yang L., Hua X.-S., Zhang H.-J. 2009. Tag quality improvement for social images. *In: Proceedings of ICME.* 2009, pp. 350–353.

Lovasz, L. 1983. Submodular functions and convexity. *Mathematical Programming The State of the Art.* s.l. : Springer Berlin Heidelberg, 1983, pp. 235-257.

M. Agosti, F. Crestani and G. Pasi. 2000. *Lectures on Information Retrieval, Lecture Notes in Computer Science.* Germany : Springer-Verlag, 2000.

M. Garey, D. Johnson. 1979. Computers and Intractability: A guide to the theory of NP-completeness. New York : W.H. Freeman & Co, 1979, 1-3.

Medelyan, O. 2009. *Human-competitive automatic topic indexing.* s.l. : University of Waikato, 2009.

N. Zhou, W. Cheung, G. Qiu, and X. Xue. 2011. A hybrid probabilistic model for unified collaborative and content-based image tagging. *IEEE Transactions on Pattern Analysis and Maching Intelligence.* 2011, 33(7), pp. 1281–1294.

P. Rozenshtein, A. Anagnostopoulos, A. Gionis, N. Tatti. 2014. Event detection in activity networks. *KDD '14 Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* 2014, pp. 1176-1185.

R. Agrawal, S Gollapudi, A. Kannan and K. Kenthapadi. 2011. Enriching Textbooks With Images. In Proceedings of the 20th ACM international conference on Information and knowledge management (CIKM '11). 2011.

R. Andersen, K. Chellapilla. 2009. Finding Dense Subgraphs with Size Bounds. *Algorithms and Models for the Web-Graph.* 2009, Vol. 5427, pp. 25-37.

R. Datta, D. Joshi, J. Li and J. Z. Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys.* 2008.

R. H. van Leuken, L. G. Pueyo, X. Olivares, and R. van Zwol. 2009. Visual diversification of image search results. *WWW.* 2009, pp. 341-350.

R. Lu, S. Zhang. 2002. Automatic Generation of Computer Animation. *Lecture Notes in Artificial Intelligence.* s.l. : Springer-Verlag, 2002, Vol. 2160.

R. Mihalcea, C. W. Leong. 2008. Toward communicating simple sentences using pictorial representations. *Machine Translation.* 2008, p. 22(3).

R. Simmons, G. Novak. 1975. Semantically Analyzing an English Subset for the CLOWNS Microworld. *American Journal of Computational Linguistics.* 1975.

R. Soffer, A. Lempel. 2001. PicASHOW: Pictorial authority search by hyperlinks on the web. *Proc. 10th Int. World Wide Web Conf.* 2001, pp. 438-448.

S. Dasgupta, C. Papadimitriou and U. Vazirani. 2006. Algorithms. New York : McGraw-Hill, 2006, 8.

S. Huston, W. B. Croft. 2010. Evaluating verbose query processing techniques. *SIGIR.* 2010.

S. Jeong, C. S. Won, and R.M. Gray. 2004. Image retrieval using color histogram generated by Gauss mixture vector quantization. *Computer Vision and Image Understanding.* 2004, 9(1-3), pp. 44-66.

S. Liu, P. Cui, H. Luan, W. Zhu, S. Yang and Q. Tian. 2013. *Advances in Multimedia Modeling.* s.l. : Springer Berlin Heidelberg, 2013. pp. 239-249.

S. R. Clay, J. Wilhelms. 1996. Put: Language-Based Interactive Manipulation of Objects. *IEEE Computer Graphics and Applications.* 1996, Vol. 15, 5, pp. 31-39.

S.Iwata, K. nagano. 2009. Submodular function minimization. In Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science. 2009, pp. 671-680.

Sun A., Bhowmick S. 2009. mage tag clarity: in search of visual-representative tags for social images. *In: WSM '09: Proceedings of the First SIGMM Workshop on Social Media.* 2009, pp. 19–26.

T. Chua, J. Tang, R. Hong, H. Zhiping, Y. Zheng. 2009. A real-world web image database from National University of Singapore. *ACM International Conference on Image and Video Retrieval.* 2009.

T. Cormen, C. Leiserson, R. Rivest and C. Stein. 2001. *Introduction to Algorithms.* s.l. : MIT Press: The Massachusetts Institute of Technology, 2001.

Targett, C. 2013. Apache Solr Reference Guide. [Online] Lucidworks, 2013. https://docs.lucidworks.com/display/solr/Apache+Solr+Reference+Guide.

—. 2013. Apache Solr Relevance. [Online] Lucidworks, 2013. https://docs.lucidworks.com/display/solr/Relevance.

U. Feige, G. Kortsarz and D. Peleg. 2001. The dense k-subgraph problem. *Algorithmica.* 2001, 29, pp. 410-421.

V. E. Lee, N. Ruan, R. Jin, C. Aggarwal. 2010. A Survey of Algorithms for Dense Subgraph Discovery. *Advances in Database Systems.* 2010, 10.

W.H. Hsu, L.S. Kennedy and S.F. Chang. 2006. Video search reranking via information bottleneck principle. *In: Proceedings of ACM Multimedia.* 2006, pp. 35–44.

Y. Chen, J. Z. Wang, R. Krovetz. 2004. CLUE: Cluster-based Retrieval of Images by Unsupervised Learning. *IEEE Transactions on Image Processing.* 2004, Vol. 13, 15.

Y. Rui, T.S. Huang. 1999. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Trans. Circuits Syst. Video Technol.* 1999, pp. 8(5), 644–655.

Z. Svitkina, L. Fleischer. 2008. Submodular approximation:sampling-based algorithms and lower bounds. *In Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science.* 2008, pp. 697-706.

Zauner, C. 2010. Implementation and Benchmarking of Perceptual Image Hash Functions. 2010.

APPENDIX

Table 7.1. The queries per each text theme.

Query		Topic 1	Topic 2	Topic 3	Topic 4
1	**Paris**	The Eiffel Tower	The Louvre Museum	The Notre Dame	The Arc de Triomphe
2	**Cairo**	The River Nile	The Citadel Of Cairo	The Pyramids of the ancient city Giza	-
3	**Rome**	St Peter’s Square	The Colosseum	Pantheon	-
4	**Santorini**	Fira	Caldera	Oia	-
5	**Istanbul**	Aya (Haghia) Sofia	The Grand Bazaar	The Blue Mosque	-
6	**New York**	The Times Square	The Brooklyn Bridge	The Statue of Liberty	The Empire State Building
7	**Madrid**	The Palacio Real	Plaza Mayor	The Almudena Cathedral	The Temple of Debod
8	**London**	The London Tower	The London Bus	Big Ben	London Eye
9	**Delhi, India**	The Taj Mahal in Agra	Humayun’s Tomb	The Jama Masjid Mosque	-
10	**Rome**	The Vatican Museum	Trevi Fountain	The Catacombs	-
11	**Barcelona**	The Sagrada Familia	Guell Park	Torre Agbar	-
12	**San Francisco**	The Golden Gate Bridge	The Alcatraz	Transamerica Pyramid	-
13	**Los Angeles**	Walt Disney Concert Hall	The Walk of Fame	The Chinatown	-
14	**Washington**	The Capitol	The White House	Washington Monument	-
15	**Moscow**	The Christ Cathedral	St Basil’s Cathedral	The Red Square	-
16	**Tokyo**	Senso Ji	The Shibuya Crossing	Tokyo Palace	Tokyo Tower
17	**Sydney**	The Harbour Bridge	The Opera House	Chinatown	-
18	**Beijing**	The Forbidden City	The Summer Palace	The Great Wall (Badaling Section)	Tienanmen Square
19	**Florence**	The Tower of Pisa	The Duomo	Palazzo Vecchio	-
20	**Berlin**	The Reichstag	The Berlin Gate	The Fernsehturm	-

Table 7.2. The most frequent tags in the dataset

nature	19657	**tree**	6262	**specanimal**	3894
sky	17329	**flower**	6239	**betterthangood**	3846
water	16586	**orange**	5941	**building**	3836
blue	16519	**usa**	5893	**cute**	3790
clouds	13201	**sun**	5882	**boat**	3737
red	12315	**street**	5831	**man**	3733
green	12262	**girl**	5807	**france**	3702
impressedbeauty	11211	**ocean**	5672	**magicdonkey**	3691
landscape	11131	**flowers**	5670	**grass**	3532
naturesfinest	11006	**searchthebest**	5531	**love**	3435
explore	10705	**winter**	5306	**asia**	3397
blueribbonwinner	10493	**colors**	5114	**sand**	3378
sunset	10195	**beautiful**	5084	**ysplix**	3367
light	10115	**snow**	5067	**spring**	3346
white	9444	**wildlife**	4690	**mountain**	3339
sea	8784	**summer**	4559	**autumn**	3334
art	7809	**pink**	4529	**golddragon**	3310
beach	7670	**urban**	4515	**cat**	3243
yellow	7652	**animals**	4369	**japan**	3227
night	7580	**photoshop**	4369	**germany**	3206
macro	7529	**river**	4309	**zoo**	3148
people	7437	**canada**	4239	**colour**	3143
portrait	7412	**uk**	4184	**shadow**	3106
architecture	7122	**lake**	4178	**dark**	3093
black	7022	**italy**	4152	**digital**	3066
trees	6934	**mountains**	4106	**sunrise**	3048
travel	6916	**europe**	4065	**london**	2993
color	6828	**england**	4059	**window**	2960
animal	6655	**woman**	4002	**fun**	2943
reflection	6588	**old**	3991	**silhouette**	2942
superaplus	6446	**film**	3956	**lights**	2939
city	6284	**park**	3940	**closeup**	2926
california	6283	**bird**	3921	**spain**	2924

Table 7.3. Some of the least frequent tags of the dataset.

cartolina	7	**panasoniclumixfz18**	7	**pamir**	7
glidden	7	**pandan**	7	**seagrass**	7

cartello	7	**aviationgreen**	7	**cargoramp**	7
panneau	7	**gladiolas**	7	**lmer**	7
cartas	7	**carniceros**	7	**cargoboat**	7
loggia	7	**panavia**	7	**abyssinia**	7
carryon	7	**treesinthemist**	7	**seafoam**	7
seattleflickrmeetups	7	**treffen**	7	**carey**	7
seattlelibrary	7	**searchandrescue**	7	**travelphotograpy**	7
carrizoplain	7	**glacierbay**	7	**girlphotographers**	7
seaswimming	7	**avi**	7	**travelnfotog**	7
glastonburyfestival	7	**carnage**	7	**travelphotographer**	7
avocadoface	7	**treerat**	7	**se17**	7
carrickfergus	7	**treebeard**	7	**cardamom**	7
lofts	7	**gla**	7	**sdcc2007**	7
carreta	7	**acadianationalpark**	7	**gipsies**	7
seastack	7	**30th**	7	**lk**	7
locomotora	7	**losroques**	7	**giovannipaoloii**	7
carrefour	7	**loboartico**	7	**llandyfriog**	7
aviopresscom	7	**trawling**	7	**carcar**	7
glassbricks	7	**30secondstomars**	7	**seaacape**	7
carrara	7	**carling**	7	**carbonbasedsentientlifeform**	7
carrally	7	**carlights**	7	**lizlieu**	7
carr	7	**traversecity**	7	**giornale**	7
seaspray	7	**lms**	7	**palloncini**	7
panicatthedisco	7	**giuliani**	7	**sd450**	7
carpodacus	7	**giuseppe**	7	**gio**	7
lockedup	7	**loaf**	7	**pallascat**	7
glasnevin	7	**carine**	7	**palladian**	7
treestump	7	**seaguls**	7	**sd10**	7
lochan	7	**seahawks**	7	**scurve**	7
searocket	7	**lmff7**	7	**livigno**	7
lochaber	7	gitanos	7	livia	7

Table 7.4. The list of tags that we removed from the dataset.

a	hdr	401s	85mmf14d
an	1d	longexposure	mm
and	dslr	213	1785
are	50mm	393	1000
as	10mm	xti	1
at	100mm	sb800	2

be	180mm	i500	3
but	1020mm	10faves	4
by	sigma1020mm	30faves30comments	5
for	sigma1020	3030300	6
if	55200mm	alf186000	7
in	2005	55200mm	8
into	2006	f456gvr	9
is	2007	a3b	10
it	2008	w80	18
no	2009	fuji9500	55
not	2010	1on1nightshots	nikonf3
of	2011	delete3	sr196
on	2012	delete4	slr
or	350d	delete5	s5
s	sd400	delete6	s5pro
such	topf25	delete7	fuji
t	100v10f	delete8	fujifilm
that	nikkor	delete9	s8000fd
the	nikkon	deleteme2	s5600
their	nikon	deleteme3	s5000
then	nikond50	deleteme4	400h
there	nikond70	deleteme5	s6500
these	nikond200	deleteme6	s9600
they	105mmf28gfisheye	deleteme7	s6500fd
this	nikkond80	deleteme8	pn400n
to	nikond3	deleteme9	400
was	nikoncoolpix8800	deleteme10	5000e
will	d3	save4	100
with	d40	save1	finepixf30
canon	d50	strobist	fp100c
5d	d70	topf25	f10
f20	d70s	topv111	400asa
v1400	d80	topv11	ee100
20d	d100	100v10f	hasselblad500cm
170500mm	d200	fv5	fp100b
1870mm	d300	lg	s3pro
eos10d	d460	tag1	fujis5500
eos400d	sonyalpha100	tag2	polaroid
30d	e500	tag3	18200
40d	mandj98	1on1	18200mm
400d	topv1000	75016	70200
o2a	1xp	75	sigma18200mm

canonpowershots3is	p1f1	topv333	25faves
a620	nikonf401s	18200mmf3556gvr	nikond100
3xp	potwkkc12	2star	selection1
photomatix	subtlehdr	theunforgettablepictures	75007
100vistas	onlyyourbestshots	flickrsbest	geolon2288638
photomatix	winner	blackandwhite	geolat48861962
3px	superbmasterpiece	geotagged	400iso
efs1022mm	exquisteshot	photography	sigma10mm
pro1	hdr*	colorphotoaward	megashot
canon30d	damniwishidtakenthat	outstandingshots	sigma10mm
canon2870mmf28usm	711149411	wow	exposure
abigfave	s2is	goldenphotographer	flickrgoldgroupaward
aplusphoto	geo:lon=229403	interestingness	wwwdgphotoscouk
diamondclassphotographer	geo:lat=488584	interestingness1	nikonstunninggallery
anawesomeshot	geo:tool=wikiworldflicksorg	interestingness2	1785mm
bravo	geo:lon=2298294	fv10	75001
flickrdiamond	geo:lat=48855835	topf100	303sph
soe	geo:tool=gmif	tophdr	selection1
bw	geo:lon=229403	geotagged	selection2
theperfectphotographer	geo:lat=488584	potwkkc10	id
goldstaraward	geo:tool=wikiworldflicksorg		

SHORT CV

Mr Nikolaos Chaliasos was born in 1986 in Ioannina, Greece. In 2004 he entered the undergraduate studies program of the Computer Science Department, University of Ioannina. He finished his studies in 2009 and in 2011 he was admitted to the graduate studies program of the same department.