Hindawi Publishing Corporation
Advances in Multimedia
Volume 2015, Article ID 439020, 10 pages
http://dx.doi.org/10.1155/2015/439020

Research Article

Supporting Image Search with Tag Clouds: A Preliminary Approach

Francesco Guerra, Giovanni Simonini, and Maurizio Vincini

DIEF, University of Modena and Reggio Emilia, Via Vivarelli 10, 41125 Modena, Italy

Correspondence should be addressed to Maurizio Vincini; maurizio.vincini@unimore.it

Received 9 September 2014; Accepted 11 December 2014

Academic Editor: Seungmin Rho

Algorithms and techniques for searching in collections of data address a challenging task, since they have to bridge the gap between the ways in which users express their interests, through natural language expressions or keywords, and the ways in which data is represented and indexed. When the collections of data include images, the task becomes harder, mainly for two reasons. From one side the user expresses his needs through one medium (text) and he will obtain results via another medium (some images). From the other side, it can be difficult for a user to understand the results retrieved; that is why a particular image is part of the result set. In this case, some techniques for analyzing the query results and giving to the users some insight into the content retrieved are needed. In this paper, we propose to address this problem by coupling the image result set with a tag cloud of words describing it. Some techniques for building the tag cloud are introduced and two application scenarios are discussed.

1. Introduction

The numbers around Twitter are impressive: the official statistics (https://about.twitter.com/company, August 2014) state that 500 million messages are sent per day by 271 million monthly active users. Tweets are now considered as an important medium adopted by governments and enterprises for their direct communications. Therefore, Twitter constitutes a large and authoritative information repository: to be able to gain any insight into it is of paramount importance.

In this field, we are not at step zero. Finding tweets containing related information, discovering trends, and analyzing impacts/influences of users and messages are still hot, open, and challenging research topics, although effective and efficient solutions have been developed and implemented in commercial tools. Applications, platforms, and appliances performing analytics on tweets are now commercialized and adopted in real business scenarios. These tools represent the outcomes achieved by the application of big data analysis (BDA) techniques to tweets. In this area, the academic and enterprise research communities have jointly put a large effort in the development of techniques for discovering unknown correlations and hidden patterns from huge amounts of data.

These software applications are typically oriented to the analysis of text and metadata which constitute a tweet. Nevertheless, tweets can include also images that can convey interesting information for the users too. Techniques and tools for the analysis of images associated with tweets are less advanced and some research work is still needed. If we consider, for example, three of the most used applications for searching images published in Twitter, that is, the search engine available in the official website, TwiPho (http://twipho.net/), and Topsy (http://topsy.com/), users can formulate query through a simple user interface which allows them to express their interests by means of keywords. The tools return a list of images which are related to the keyword queries according to some algorithm. Nevertheless, to understand why an image has been included in the result list for a specific query is often a mystery to the user. Let us suppose, for example, that a user is querying the Twitter interface looking for images about Modena and expecting as a result photos about this Italian city. It is not easy for a user to understand why some of the top images retrieved and shown in Figure 1 are connected to the Modena city. We can justify the presence of vinegar bottles and Ferrari cars in the results since Modena is the city where they are produced, but

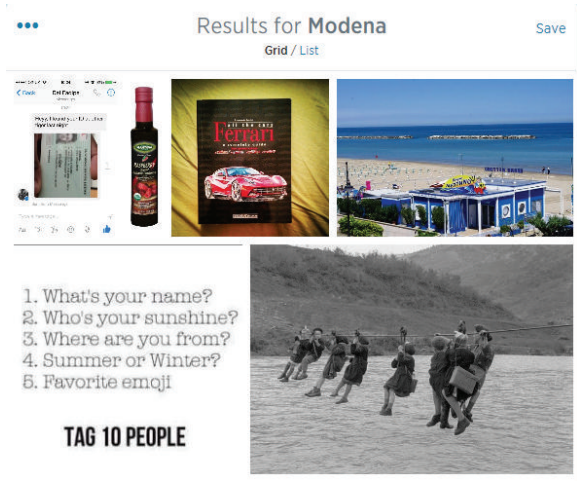

Figure 1: The top images resulting by the query "Modena" in Twitter.

the connection with Modena and the beach (last image in the first row) is hard to justify, since Modena is not close to the sea. The absence of effective techniques for analyzing the images in Twitter constitutes a tremendous gap since it has been estimated that around 36% tweets include images.

Therefore, we think that providing users with some techniques for analyzing the images retrieved by a user query is needed. Informal user studies at Yahoo!, as reported in [1], indicate that augmenting related suggestions with concrete explanations would significantly increase the relevance of the suggestions and increase user engagement. According to these studies, we claim that users would have a great benefit if they are able to couple the image result set with some justification concerning its relevance to the user query. In this paper, we discuss the abstract problem of generating tag clouds describing result sets and we propose techniques for generating tag clouds of words summarizing and explaining the images retrieved by a query. A tag cloud is a visual representation of text data associated with the images where the importance of each tag is shown with font size or color. The use of tag clouds for summarizing the results of a user query has been studied in the literature even if not frequently adopted in the image search, where we devise a two-step process for (1) tokenizing—the goal of this step is to associate images with some representative words—and (2) tag cloud generation—this step provides an effective tag cloud by using the words identified by the first step.

In some cases, the first step can be trivial: the words associated with the images are manually provided by the users publishing the image. This is the case of Flickr, one of the most used websites for hosting and sharing images and videos, which allows users to include several tags for each shared image and provides a system for searching images related to some specific user-provided tag. When you perform this kind of search in Flickr (https://www.flickr.com/photos/tags), the images returned to the users are shown along with some other tags which represent elements "related" to the tags searched. Even if a tag cloud is shown, this kind of information does not provide any insight into the images included in the result set. Also Twitter allows users to tag messages and these words (or, in case of their absence, some representative words in the message can be extracted by means of some NLP technique) can be used as input for a tag cloud generator system.

The second step needs to address two main issues: the development of techniques for the selection of which words to visualize in the tag cloud (typically a tag cloud is composed of around 40 words and the union of the tags associated with the images belonging to a result set can have a higher cardinality) and the definition of the shape (e.g., size, font, and color) to assign to each word. We claim that the process for generating tag clouds is the result of the application of three fundamental operations: (a) the selection; that is, we need functions that can reduce the total number of tags which are associated with a result set; (b) the ranking; that is, we need techniques able to order tags with respect to some metrics; and (c) the partition, that is, techniques for grouping tags. We conceive these operations as "abstract procedures": obviously there are several possible implementations for each operation, each one addressing a specific goal. For example, ranking can take into account the publication time or the frequency of a tag for ordering.

In this paper, we introduce a novel tag-based system supporting the search of images published with some textual information. Two particular use cases where the system has been implemented will be shown: MediaPresenter, a system for the creation of multimedia presentations, and an image search engine for Twitter. Our goal is to provide a tool for analyzing an image result set: for this reason our proposal can be conceived as an add-on to be coupled with the image search engine. For the sake of simplicity, we consider only images having some descriptive hashtag in the tweet. This will be used by our system to generate the tag cloud. Nevertheless, NLP techniques for extracting the most representative words of a tweet can be used for this purpose. Our idea, which extends our previous proposal [2], is to model hashtags as a network, where two tags are linked if they are associated with the same image, and to provide users with specific implementations of the fundamental operations. Our system will allow users to select and combine these operations and their specific implementations to generate a tag cloud (or a set of tag clouds) representing the images of interests. With reference to the previous example, the user can apply a partition function to the hashtags associated with the images of Modena and discover that images are related to 3 main topics: cars, vinegar, and monuments. Moreover, he can apply a ranking function and creating a tag cloud where the elements are ordered on the basis of their frequency.

The rest of the paper is structured as follows. The next section introduces some related work. Section 3 describes the model underlying the proposal and Section 4 introduces the functional architecture of the system and two use cases where the application has been implemented. Finally, Section 5 introduces some conclusions and future work.

2. Related Work

The problem of selecting a limited number of tags representing a result set has been studied in the literature and in [3]

a model and some metrics for generating tag clouds from a set of labels have been proposed. Tag clouds generated according to the proposed techniques were implemented as part of CourseRank [4]: a social tool to access official university information and statistics.

Other interesting approaches include PubCloud [5], a project that uses tag clouds for summarizing query result of PubMed biomedical literature database. The tag clouds are generated from words extracted from the paper abstracts of the query results. The font size of the words in the cloud is calculated just using terms frequency, and the set of visualized tags are obtained by using the tags having frequency higher than 10% of the best frequency: both the score and the visualized set computation are very simple with respect to other approaches, such as our technique. In [6] three different approaches to determine word cloud generation from web search results are used: full-text, query biased, and anchor text based clouds. They define a specific model to score the terms and a greedy algorithm to select best tags: preliminary results are obtained by using 2009 TREC Web Track as documents set evaluation. Moreover, the Rex system [1] is somewhat related to the motivation of our research. In the paper the need of providing explanation for keyword queries result is highlighted. We think that tag cloud can provide this information. Finally, concerning keyword search over tags, the proposed technique takes some inspiration from our previous work on keyword search over relational databases where two prototypes have been proposed [7, 8].

3. The Model

In this section we describe the data representation model enabling the generation of tag clouds through the primitive operations. We consider a set of images C and the set of tags T associated with it. In our model, tags associated with images are represented as a graph, where the nodes are the tags and an edge exists in tags cooccurring in the same image. Inspired by [3], we propose and model some measures to be applied to our graph model for the generation of effective tag clouds. Our goal is to provide a technique, based on the selection, ranking, and partitioning operations, to generate and evaluate tag clouds that summarize the results retrieved by the image search engine. We are not interested in how these results are generated or ranked by the search application: our work is independent of the way the result set is computed. In this way our proposal is applicable to any system for image search. Moreover, in this section we intend to describe a general model and the operations that can be performed exploiting it. The actual implementation of the system is based on this model and it employs only a subset of the proposed measures.

Let us assume that a query q is a subset of T: this is a reasonable assumption since, typically, from a query q, an image search engine returns a collection of images $C_q \subseteq C$ and the set of tags T_q associated with at least one image $c \in C_q$. We denote by $A(t)$ the association set of the images that are tagged with a tag t and belong to C.

Definition 1 (association set of t under q). One defines the association set $A_q(t)$ of a query q as $A_q(t) = A(t) \bigcap C_q$.

In order to provide a flexible model to generate tag clouds, we consider the general definition of a scoring function:

$$s: C_q \longrightarrow [0, 1]. \tag{1}$$

A straightforward scoring function is the frequency of an image in $c \in C_q$. Employing it, all images have the same score equal to $1/|C_q|$, where $|C_q|$ is the cardinality of C_q. We call this function frequency function $f(\cdot)$. Other scoring functions can be exploited in order to give more importance to some images rather than others, for instance, assigning a higher score on the basis of the timestamp of the image: the more a photo is recent the more it is relevant, while older photos could also be not considered under a certain threshold. Other scoring functions can be taken into account: spatial proximity (with respect to the user's position) and the "social level" of the results (measured, e.g., by the number of retweets in Twitter).

Given a scoring function $s(\cdot)$, we want the space of tags T_q as a graph of tags $G_q(V, E, \Pi, \Gamma)|_s$ where V represents the vertices and each vertex is a tag $t_i \in T_q$; E is an edge between two tags $t_i, t_j \in T_q$ with $i \neq j$ that cooccurs in an image $c \in C_q$; Π and Γ are, respectively, the set of weights of the vertices and the set of weights of the edges. The weight of a node is defined as follows:

$$w_{t_i} = \sum_{c_i \in A_q(t)} s(c_i). \tag{2}$$

Intuitively, if $s(\cdot) = f(\cdot)$, the weight of a node t_i represents the fraction of images that belong to C_q and are tagged with t_i. Then, we define the weight w' of a pair of tags and use it to define afterwards the weighs of the edges:

$$w'_{t_i,t_j} = \sum_{c_k \in A_q(t_i) \bigcap A_q(t_j)} s(c_k). \tag{3}$$

Once again, if $s(\cdot) = f(\cdot)$, then w'_{t_i,t_j} represents the cooccurrence of the two tags t_i and t_j, normalised for the total number of images $|C_q|$. The w'_{t_i,t_j} can be directly used as weight w_{t_i,t_j} of the edge between two tags in G_q; alternatively, it allows computing other measures to be exploited as weights of edges in G_q. In fact, the computation of Dice, Jaccard, and Cosine coefficients becomes straightforward:

$$\begin{aligned} \text{Dice}_{t_i,t_j} &= 2 * \frac{w'_{t_i,t_j}}{w_{t_i} + w_{t_j}} \\ \text{Jaccard}_{t_i,t_j} &= \frac{w'_{t_i,t_j}}{w_{t_i} + w_{t_j} - w'_{t_i,t_j}} \\ \text{Cosine}_{t_i,t_j} &= \frac{w'_{t_i,t_j}}{\sqrt{w_{t_i} * w_{t_j}}}. \end{aligned} \tag{4}$$

Formally, a tag cloud S is a subset of T_q, represented in our model as a subgraph G_S of the graph G_q of tags under a query

q that summarizes the query results and aims to help the users in the navigation of them. In order to quantitatively evaluate the goodness of a tag cloud we define below two measures: the coverage and the overlap of S.

3.1. Evaluation. The coverage $\mathrm{cov}(S)$ of a tag cloud S is a measure of the fraction of images belonging to C_q that are associated with tags of S:

$$\mathrm{cov}(S) = \frac{W_S - I_S}{W_{C_q} - I_{C_q}}, \tag{5}$$

where I_P and W_P are, respectively, the weight of the inner edges and the weight of the inner nodes of a set of vertices P:

$$I_P = \sum_{\substack{t_i,t_j \in P \\ t_i \neq t_j}} w_{t_i,t_j}, \qquad W_P = \sum_{t_i \in P} w_{t_i}. \tag{6}$$

The coverage of a tag cloud S depends on the number of images $c_i \in C_q$ that are associated with tags belonging to S and their score $s(c_i)$. If $s(\cdot) = f(\cdot)$ and $w = w'$, then the coverage of S represents the exact fraction of images of C_q associated with at least one tag of S.

Another measure we take into account evaluating a tag cloud is the overlap $\mathrm{overlap}(S)$ of S that can be seen as a measure of redundancy, that is, how many images associated with a tag $t_i \in S$ are associated also with another tag $t_j \in S$:

$$\mathrm{overlap}(S) = \frac{I_S}{W_S}. \tag{7}$$

Generally, a desirable tag cloud should have a low overlap and a high coverage. Below we define and discuss about some metrics useful to perform the selection and ranking of T_q in order to build S.

3.2. Ranking and Selection. To perform the ranking and the selection of nodes belonging to the graph of tags $G_q(V, E, \Pi, \Gamma)|_s$, we need to determine the importance of a node within the network. If the importance can be measured, both selection and ranking of nodes can be performed basing on it.

We can consider nodes with a high number of connections (also considering the weight of those connections) to be important. A measure that captures this idea is the centrality degree $D_c(t_i) = \sum_{t_j \in T_q} w_{t_i,t_j}$ that can be normalised by the maximum degree d_j in the network, or by the degree sum I_c.

Another possible approach is to compute the PageRank of the nodes in the graph, relying on the idea that if an important node is connected to another node, the latter should be important too.

All these measures lack one aspect: they tend to consider all nodes belonging to a unique cluster that have to be summarized. This is a quite contrived assumption that rarely fits for real cases. In fact, generally the results can be classified in different clusters due to the intrinsic ambiguity of the tags. Hence, besides allowing a partitioning and a ranking task, we need a further step: the partitioning.

3.3. Partitioning. Many algorithms can be employed to partition a graph [9]. The goal of this task is to minimise the overlap among different clusters and at the same time maximise the strength of the association of the members inside a cluster. To give a quantitative measure, according to the previous notation, we define the group clustering coefficient as

$$\mathrm{cluster}(S) = \frac{I_S - \beta B_S}{W_S}, \tag{8}$$

where $B_S = I_C - I_S$ is the sum of the weights of the edges that link a node inside to one outside the cluster.

Similarly, our model allows to exploit community detection algorithms, in particular [10] that proposes a technique to identify automatically the optimal number of communities, as well generating hierarchical or overlapping ones.

4. The System

Our application is conceived as two separate components that work coupled: a search engine retrieving images and related tags and a cloud generator that analyzes the results provided by the first tool to generate a summarized view in the form of a tag cloud of the contents. The components are independent of each other: the generation of the tag clouds does not rely on the technique adopted for retrieving the images, but it only takes into account the images and tags retrieved and, vice versa, the image searching is not influenced by the tag cloud generation process. Moreover, working with images and related tags is not a limitation, since there are applications working in real scenarios managing and retrieving images and related words.

Figure 2 shows the functional architecture of the searcher and the tag cloud generator components. In our preliminary prototype, the searcher component actually returns images on the basis of the associated descriptions. As shown in the figure, the first operation performed by the component is the collection of images and related descriptions. The goal of this step is to generate tags which are representative of the images starting from their descriptions. The component implements a simple information retrieval technique, based in particular on vector spaces [11], for retrieving and ranking images on the basis of their descriptions with respect to user keyword queries.

The input of the second component is the result of the user keyword query computed by the searcher, with the aim of generating its representation in the form of tag cloud. The technique proposed models the result set as a graph where nodes are the tags and edges exist between tags associated with the same image. Moreover, the nodes in the graph are weighted on the basis of their frequency. Starting from this graph representation of the result a tag cloud is generated by the application of the three fundamental operations of selection (to reduce the number of tags), ranking (to discover the most important ones and to visualize them accordingly), and partition (to cluster tags which are in some way related). There are a number of possible implementations of these operations, which are based on and try to maximize different

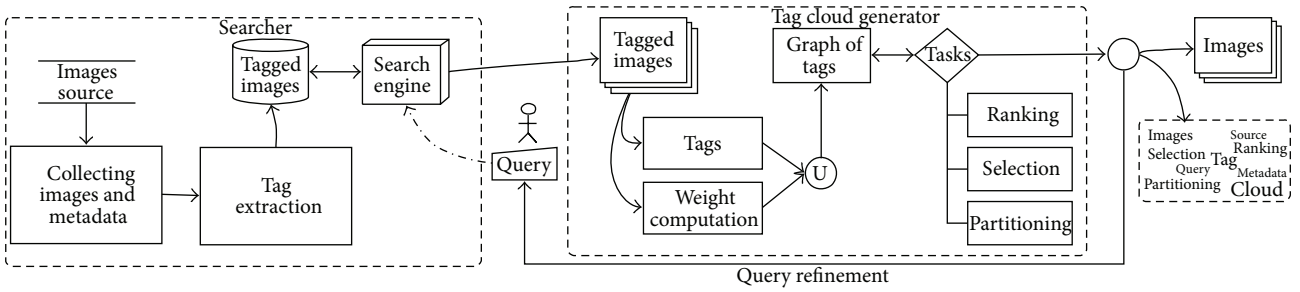

Figure 2: The functional architecture of our application.

measures. In the following, we will propose some proposals as they have been implemented in our prototypes. Moreover, we think that it is a responsibility of the user to choose which operation to apply (only one operation or a series of them), the order of their execution, and the specific implementation.

A first implementation of our idea has been proposed and tested in MediaPresenter, a tool developed through a join collaboration between the DBGroup at University of Modena and Reggio Emilia and Addiction Creation Media Lab, an Italian SME. Part of the activity has been funded by Italian Emilia Romagna region, within the LISEA laboratory (http://spring.bologna.enea.it/lisealab/). As described in Section 4.1, the tag cloud generated in this project is obtained as the result of a selection process only and by using a measure. In Section 4.2, we show our ideas implemented in MediaPresenter can be extended for dealing with the retrieval of images in Twitter.

4.1. MediaPresenter. MediaPresenter is an online cross-platform application, which offers a large number of services for sharing and managing digital archives. The main goal of MediaPresenter is to produce multimedia presentations and this could be done by combining multimedia resources available in a repository called MediaBank. We experimented our ideas for supporting image search with tag cloud in MediaPresenter by implementing a service supporting users in retrieving interesting contents from the MediaBank repository. Other services are offered by MediaPresenter and they include the concurrent access to data by multiple users with different roles, the possibility of importing multiple types of digital resources (3D representations, videos, images, etc.) and exporting them in various formats (swf, pptx, png, and pdf), and a keyword-based search engine for retrieving digital contents from MediaBank. The process for generating presentations consists of four steps, namely, selection, assembling, transformation, and presentation.

The Selection Step. In this step, the multimedia resources contained in the MediaBank are retrieved using a information retrieval technique based on vector space. Each multimedia resource in the repository is identified by a unique code (typically the name of the file) and has associated a series of metadata which specify contents and the properties that characterize such element. The metadata constitute the search space where the IR technique looks for the keywords provided by the user as an input.

The Assembling Step. In this step, the user assembles the final presentation starting from the single resources retrieved in the previous step. Users can also assemble different types of elements depending on their role. For instance, one user can be a slide-maker and therefore he can assemble just slides; meanwhile a presentation-maker can assemble only complete presentations.

The Transformation Step. Once the user has repeated iteratively the previous steps and the complete presentation has been created, the transformation step allows him to save the final product in different formats according to the user needs.

The Presentation Step. MediaPresenter client has been developed as a Flex application running inside a browser; hence a presentation can be potentially shown on whatever device having a browser and an Internet connection. Nevertheless, a presentation can be published in different formats, and consequently it can be shown through other applications.

4.1.1. The MediaPresenter Tag Cloud. The main goal of MediaPresenter is to support users in publishing multimedia presentations. To reach this goal, the system provides the user with all the available information about the topic which can be briefly summarized in presentations, unpacked slides, or series of slides and digital resources already used in the past or available as digital content of the enterprise. During the creation process, the typical action performed by users is searching resources stored into the MediaBank using name, dimension, type, and date of creation as search criteria. The results from these searches are often unsatisfactory, mainly due to the lack of database structure knowledge and experience of the user. Furthermore, database search results are often considered as database tuples, whereas nontechnical people think in terms of entities, not tuples. To overcome this issue, we devised a particular search method which allows users to perform a keyword search over the MediaBank database using terms that can be found in different fields (name, title, description, etc.) and in multiple relations, hiding the data structure to the user. In order to make this approach even more effective, the system enables users to

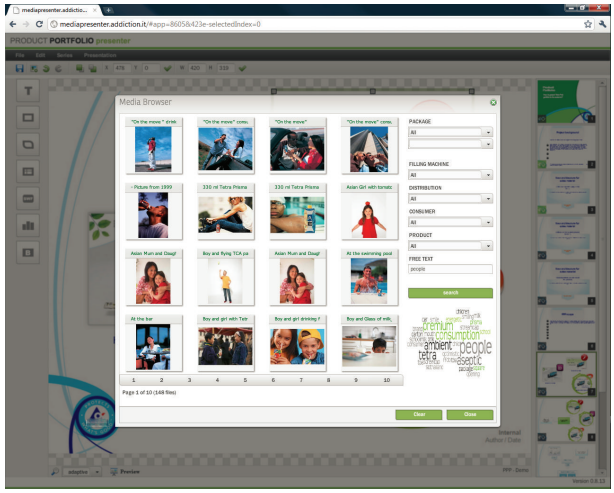

Figure 3: The MediaPresenter GUI.

associate words to each digital resource: these tags that can be manually specified by the users choosing the correct ones among a predefined set of proposals or creating new tags ex novo at run-time. The action of organizing resources by adding metadata is called "tagging" and it is gaining popularity on the web in these years [12]. Using tags to annotate resources allows the system not only to specify the keyword search over the stored assets, but also to create tag clouds and to consider each term in the cloud as a hyperlink that can be used to refine the search results, dynamically guiding users in the hidden relationship among contents and eventually leading to serendipitous discoveries of interesting results.

Figure 3 shows the MediaPresenter interface used for keyword searching supported by tag clouds. In this case the user was looking for all the multimedia resources containing the term "people." The user can select the preferred resource on the left, and the tag cloud at the bottom right of the image supports the user in browsing all the related resources.

4.1.2. Keyword Search over the MediaBank and Tag Cloud Generation. In this section, the process adopted in MediaPresenter for keyword search and tag cloud generation is introduced. Let us denote by D the MediaBank database part where the search process is applied and by R_i the ith relation stored in D. Each relation contains a set C of columns. We consider $R_i \cdot C_j$ as the jth column of the ith relation. By t we denote a generic tuple in D. Given R_i and R_j and a primary-foreign key relationship between R_i and R_j, we consider TD as the tuple graph of the database D, where for nodes we consider each tuple in D and given two tuples $t_i \in R_i$ and $t_j \in R_j$, an edge exists among them if $(t_i \bowtie t_j) \in (R_i \bowtie R_j)$. D can be modeled as a collection V of search entities. In our case V corresponds to the set of digital assets that can be returned by the search method, providing, thus, a sort of unit of representation for returned entities. For each entity $v \in V$ we consider C_i as the ith attribute describing the entity. C_i can be seen both as a one-to-one mapping to a particular column in the database

Table 1: The MediaBank fragment containing data searchable by the users.

(a) `DAM_object`

Asset_ID	Asset_name	Folder_name	Asset_file_name
DA001	asset_001	Products	box_front.jpg
DA639	opening_3	Packages	B03240.jpg
DA640	opening_4	Packages	ustraws.jpg

(b) `tag`

Tag_ID	Term
T45	Yellow
T10	Box machine
T55	Products
T2	Umbrella
T19	Aseptic
T31	Juice
T50	Square
T87	Operator

(c) `serie_slide`

Slide_ID	Title	Source
S33	Overview	<mx:Application> ...
S101	Special box	<mx:Application> ...
S6	Conclusions	<mx:Application> ...
S869	Updates	<mx:Application> ...

(d) `presentation_serie_template`

Object_ID	Name	Description
P50	The new box machine	In this presentation...
P13	About MediaPresenter	MediaPresenter is a...

(e.g., the resource name) and also as a many-to-one mapping, therefore grouping several information in one search entity attribute (e.g., the set of tags specified for a digital asset can be thought as an attribute of the asset entity). In particular, we consider C_0 as the identifier of each specific search entity. An entity ID, hence, is a mapping to the primary key of the relation R_0, where R_0 is called primary entity relation. In our context, we recognize as primary entity relation the table `DAM_object`, which provides all the digital resource IDs. On the other hand, we call all the other relations that join directly or indirectly with R_0 and provide additional information to v secondary entity relation. We identify tag, `presentation_serie_template`, and `serie_slide` tables as secondary entity relations. Table 1 depicts the database relations we took into account for the keyword search.

Summarizing, the keyword search engine we developed returns digital resources identified by the ID stored in the `DAM_object` primary relation. Each resource contains attributes directly related to the primary relation (name, file name, folder name) but also attributes grouping information belonging to secondary relations. Each entity gets information about the slide(s) it belongs to, thanks to the join with the `serie_slide` relation. In addition, information

about the presentation (series, template) is added joining the relation `serie_slide` with `presentation_serie_template`. At the end the relation tag provides information about the set of assets tags as well as information about the set of tags related to the presentation (series, template) containing the asset entity.

The input of our search function is a query q, and we assume that q is composed of a certain number of keyword terms. We assume that given a keyword term k and a search entity v identified by the ID stored in the tuple t of the primary relation R_0 and v contains k if one of the following statements holds: (a) one of the attributes values of t contains k; (b) TD contains a tuple t_i stored in the relation R_i that contains an attribute value equal to k, and a path in the tuple graph connecting the tuple t_i to t exists. Given a query q, the set of resulting entities is denoted by $V_q \subseteq V$, and it contains the set of search entities related at least to a keyword term k contained in q.

In this application, we proposed the selection operation for generating the cloud from the tags associated with the multimedia resources. For this purpose, let us consider the set L of all tags. These tags are textual labels (words) assigned to a resource; thus each resource $v \in V$ is associated with a set of tags, denoted by L_v. We denote by L_q the set of tags related to the entities contained in V_q; similarly we consider $V_q(l) \subseteq V_q$ as the set of objects associated with the tag $l \in L_q$. Following the coverage measure introduced in [3] we implemented a tag selection algorithm to maximize the number of entities in V_q that are covered by the set of tags $S \subseteq L_q$ resulting as output. In addition, since the user creates tags in a context of a group and each group has a label identifying its generic topic, we can consider that given a certain search result, the sum of all the groups' labels is itself a tag cloud summarizing all the topics of the results. In order to sum up the tag clouds resulting from the tags and the one resulting from the group labels, we use colors to identify for each tag the group it belongs to and an index showing all the groups related to the search. In this way the user is able to perform a refinement over the results using two different levels of granularity: by generic topic, selecting the group of interest on the index, and one more detailed using the tags from the cloud.

4.2. Searching for Images in Twitter. Twitter offers a second use case of where to apply our ideas for supporting search for images by means of tag clouds. Our work in this scenario is still preliminary: for the moment we have developed a software prototype and we are now evaluating the user experience in using our techniques.

Twitter already provides a search engine; nevertheless, as observed in the introduction, in some cases it is not clear in which way images are related to the user keyword queries. We claim that a tag cloud can support the user in this process, by providing some insight into the contents of the image result sets. This knowledge can furthermore be exploited for refining the keyword queries, by adding terms better reflecting the intended meaning of the user and improving the quality of the results. Our idea is to implement our technique as an add-ons application that analyzes the result obtained by the Twitter search engine. With reference to the functional architecture in Figure 2, the first component is already implemented by Twitter and we have focused our effort on the second component. There are two main issues that have been addressed for the development of the system: (1) the definition of the "candidate tags" for a tweet and (2) the development of a GUI for managing the user interaction and some implementation of the three fundamental operations to be used in this context.

Concerning the first point, the issue is to define which tags are associated with an image. Twitter allows users to define specific tags, called "hashtags"; nevertheless in our experience the use of only this metadata is limiting since it is not adopted by users. In our testing dataset (obtained by a random selection by real posts published on Twitter), only 23% of tweets with an associated image have also some hashtags. For this reason, our choice was to consider as tags all the words part of the tweet, after the removal of stop words and a stemming process. In this way, the number of tags associated with each image increased. If needed (this was not the case for our experiment), to reduce the noise generated by irrelevant words, it is possible to reduce the number of tags by considering, in the graph generations, only terms that appear as more than a specified threshold.

Figure 4 shows the GUI supporting users in the tag cloud generation. The interface is divided into two parts. In the main part the images retrieved by the Twitter search engine, as an answer to the user keyword query formulated through the input text box at the top, are shown along with their tag cloud. A simple dashboard allows users to select which operations, which implementation of them, and in which order they are to be executed for the generation of the cloud. In the right part of the GUI, the system shows what happens, in terms of images retrieved and tag cloud, with a number of automatic computed refinements of the user query. The suggestion of new possible keywords directly derives the process for generating tag clouds, where the best results (according to the metrics selected by the user) can be added to the original ones. In this case the application shows three refinements, but the number can be specified by the user. In this way, we devise an iterative process, where the user can manually refine his queries or be supported by the application in finding what he is looking for.

There are some possible metrics that can be used for the implementations of the selection, ranking, and partition operations. Table 2 shows some metrics that we have exploited to develop techniques dealing with Twitter images. In particular, techniques based on the analysis of space and time proximity can be exploited for implementing versions of all the operations. By using techniques based on these measures we can select, rank, and group tags on the basis of their spatial/temporal proximity. According to these measures, for example, we can assign high priority to the most recent tags. Techniques based on frequency can be used for selecting and ranking tags. For example, we can exploit the coverage degree (i.e., the amount of images which are associated with a subset of the tags [3]) and the overlap degree (see Section 3) for selecting the tags which better summarize the images to be inserted in the tag cloud. Since tags are

Figure 4: The GUI of the tag cloud generator adapted for Twitter.

Table 2: Some possible implementation of the fundamental operations.

Metrics	Selection	Ranking	Partition
Space/time proximity	X	X	X
Frequency	X	X	
Link analysis	X	X	X
Semantic analysis	X		X

Table 3: Selected tags associated with the query "Modena."

Selected tags
Agriculture, vehicles, autos, race, food, Italian, cities, Italy, football, match, ModenaFC, healthy_food, aceto, dome, Historic_Center, Maranello, team, grapes, people, university, places, Balsamico, sports, Sassuolo, pasta, restaurants, winery, wine, Ferrari, Ghirlandina, square, ham, stadium, landscape, Enzo_Ferrari, museum, countryside, world

modeled as a graph, we can exploit metrics which have been developed in the field of link analysis for all the operations. In Section 3, the centrality degree and PageRank have been proposed for analyzing the graph representation. Techniques based on these values can be exploited for the selection, ranking, and partitioning of the tags. Finally, semantics-based technique can be used for identifying tags which are similar, or representing more/less generic concepts than other ones.

4.2.1. Motivating Example. As a motivating example, we address the keyword query "Modena" to our system, and we show how the tag cloud associated with the results can be generated exploiting the proposed operations of selection, ranking, and partitioning. First of all the system retrieves all the images that are associated with the tag "Modena" and collects all the tags associated with them.

Starting from the list of tags, the system computes the cooccurrence and similarity metrics and builds the graph of tags. Finally, a selection operation is performed trying to minimize the overlap of the tags while selecting those with higher degree centrality. Table 3 shows the selected tags. All these tags are then ranked, on the basis of their importance, measured with the PageRank. Visually the notion of importance is expressed by means of the font size of tags shown to the user. To better summarize the content of the tag cloud, the set of tags are partitioned exploiting a community detection algorithm. This allows the tool to manage the intrinsic ambiguity of a query, since the retrieved images are about different topics. In our example, the community detection algorithm identifies four main topics: the food, the Modena football team, the Ferrari cars, and the city of Modena and its historical center (see Table 4).

The analysis of the search task highlighted by this motivating example shows that the combination of selection, ranking, and partitioning procedures provided by our proposal can really support users in effective and efficient keyword search in Twitter images.

To the best of our knowledge, the procedures provided by our approach and the support that it can provide in the search process are not available in any other tool. A qualitative comparison of our proposal with some of the available commercial keyword search systems for images is shown in Table 5. In particular, we formulated the same keyword query

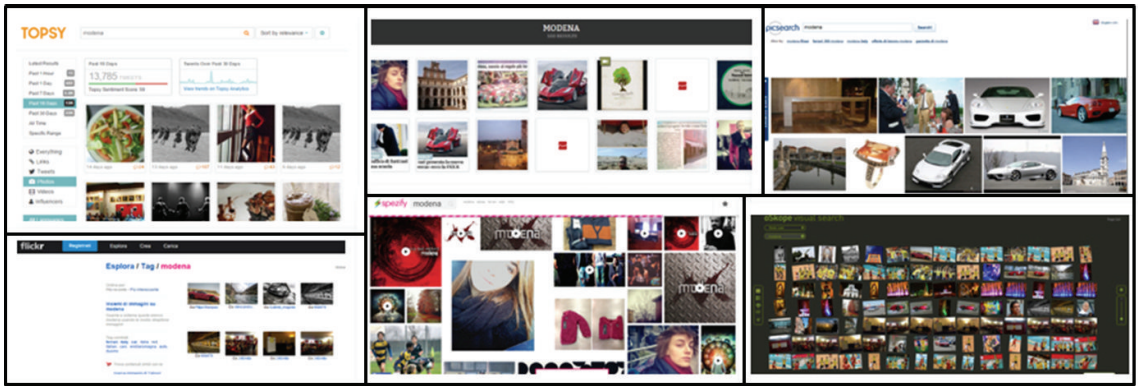

Figure 5: The six tools under evaluation.

Table 4: Partitions of the tag cloud.

Partitions	Tags
City	Italian, cities, Italy, dome, Historic_Center, people, university, Sassuolo, Ghirlandina, square, landscape, countryside, world places
Car	Vehicles, autos, race, Maranello, Ferrari, museum, Enzo_Ferrari
Football	Football, match, ModenaFC, team, stadium, sports
Food	Agriculture, healthy_food, aceto, grapes, Balsamico, pasta, restaurants, winery, wine, ham, food

Table 5: Qualitative comparison of well known tools.

Website	Images	Tags	Tags—images	Tag expl.
Topsy	2	n.a.	n.a.	n.a.
TwiPho	1	n.a.	n.a.	n.a.
Picsearch	2	1	2	2
Flickr	2	1	1	3
Spezify	3	2	2	n.a.
oSkope	1	n.a.	n.a.	n.a.
Our proposal	2	1	1	1

("Modena") to the tools: Topsy (http://topsy.com/), TwiPho (http://twipho.net/), Spezify (http://www.spezify.com/), Picsearch (http://www.picsearch.co.uk/), Flickr (https://www.flickr.com/photos/tags), and oSkope (http://www.oskope.com/). The results provided by the tools and shown in Figure 5 have been analyzed and compared with the ones obtained by our proposal according to four perspectives: quality of the images retrieved (i.e., we evaluated if the images provided as a result can be easily associated in some way with the subject of the query), quality of the tags retrieved (i.e., we evaluated if some tags are provided as a result and if they can be easily associated in some way with the images and the subject of the query), relationships between tags and images (i.e., we evaluated if images and tags retrieved are in some way related), tag explorations (i.e., we evaluated if and in which way the system supports search for related tags). For each perspective and each system, a rating (1—strongest, 3—weakest) is provided if the feature is implemented.

Topsy, oSkope and TwiPho are tools oriented only to image retrieval: they do not provide any tag or justification for the answers returned to the user. The evaluation reported in Table 5 represents the accuracy obtained by analyzing the relevance of the images in the result set. In our experiments, TwiPho and oSkope performed better than Topsy. Picsearch, Flickr, and Spezify associate tags with the images. In our experiments, Spezify was not able to retrieve accurate results both in terms of images and in terms of tags. Moreover, the tags proposed are lowly related to the query (i.e., the system generated the tags "Modena," "stamp," "Ferrari," "stati," and "1852." Among these tags, only Modena and Ferrari are related to the query). Flickr and Picsearch are the tools which performed better in our experiments; nevertheless, in both the tools, searching for tags produces new queries which are only poorly related to the initial user's request.

5. Conclusion and Future Work

In this paper, we have presented our approach for coupling result sets with tag clouds. We think that the knowledge conveyed by the tag cloud can be extremely useful to provide to the user a better insight into the relevance of the result set. This knowledge is really helpful when users are looking for images, where typically there is a mismatch between the way the users formulate queries (by text) and the results obtained (images). In this scenario, finding a connection between query and answer can be hard. We have introduced our preliminary ideas and we have shown how these ideas have been implemented in two systems. The next steps of our research are mainly two: (1) to develop new techniques for implementing the selection, ranking, and partition operations for the tag cloud generation and (2) to evaluate the user experience in using our approach, with particular reference to the Twitter scenario.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] L. Fang, A. D. Sarma, C. Yu, and P. Bohannon, "Rex: explaining relationships between entity pairs," *Proceedings of the VLDB Endowment*, vol. 5, no. 3, pp. 241–252, 2011.

[2] S. Bergamaschi, F. Ferrari, M. Interlandi, and M. Vincini, "Mediapresenter, a web platform for multimedia content management," in *Sistemi Evoluti per Basi di Dati—SEBD 2011, Proceedings of the Nineteenth Italian Symposium on Advanced Database Systems, Maratea, Italy, June 26-29, 2011*, G. Mecca and S. Greco, Eds., p. 437, 2011.

[3] P. Venetis, G. Koutrika, and H. Garcia-Molina, "On the selection of tags for tag clouds," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM '11)*, pp. 835–844, February 2011.

[4] B. Bercovitz, F. Kaliszan, G. Koutrika et al., "Social sites research through courserank," *SIGMOD Record*, vol. 38, no. 4, pp. 29–34, 2009.

[5] B. Y.-L. Kuo, T. Hentrich, B. M. Good, and M. D. Wilkinson, "Tag clouds for summarizing web search results," in *Proceedings of the World Wide Web Conference (WWW '07)*, pp. 1203–1204, 2007.

[6] R. Kaptein and J. Kamps, "Word clouds of multiple search results," in *Multidisciplinary Information Retrieval*, vol. 6653 of *Lecture Notes in Computer Science*, pp. 78–93, Springer, Berlin, Germany, 2011.

[7] S. Bergamaschi, E. Domnori, F. Guerra, M. Orsini, R. T. Lado, and Y. Velegrakis, "Keymantic: semantic keyword-based searching in data integration systems," *Proceedings of the VLDB Endowment*, vol. 3, no. 2, pp. 1637–1640, 2010, http://www.comp.nus.edu.sg/~vldb2010/proceedings/files/papers/D31.pdf.

[8] S. Bergamaschi, F. Guerra, M. Interlandi, R. T. Lado, and Y. Velegrakis, "QUEST: a keyword search system for re lational data based on semantic and machine learning techniques," *Proceedings of the VLDB*, vol. 6, no. 12, pp. 1222–1225, 2013.

[9] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3–5, pp. 75–174, 2010.

[10] Y.-y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 7307, pp. 761–764, 2010.

[11] W. B. Croft, D. Metzler, and T. Strohman, *Search Engines—Information Retrieval in Practice*, Pearson Education, 2009.

[12] S. A. Golder and B. A. Huberman, "The structure of collaborative tagging systems," http://arxiv.org/abs/cs/0508082.