Geo-Temporal Distribution of Tag Terms for Event-Related Image Retrieval

Massimiliano Ruocco, Heri Ramampiaro

Norwegian University of Science and Technology
Dept. of Computer and Information Science
Trondheim, Norway
***Email**: {ruocco,heri}@idi.ntnu.no*

Abstract

Media sharing applications, such as Flickr and Panoramio, contain a large amount of pictures related to real life events. For this reason, the development of effective methods to retrieve these pictures is important, but still a challenging task. Recognizing this importance, and to improve the retrieval effectiveness of tag-based event retrieval systems, we propose a new method to extract a set of geographical tag features from raw geo-spatial profiles of user tags. The main idea is to use these features to select the best expansion terms in a machine learning-based query expansion approach. Specifically, we apply rigorous statistical exploratory analysis of spatial point patterns to extract the geo-spatial features. We use the features both to summarize the spatial characteristics of the spatial distribution of a single term, and to determine the similarity between the spatial profiles of two terms – i.e., term-to-term spatial similarity. To further improve our approach, we investigate the effect of combining our geo-spatial features with temporal features on choosing the expansion terms. To evaluate our method, we perform several experiments, including well-known feature analyses. Such analyses show how much our proposed geo-spatial features contribute to improve the overall retrieval performance. The results from our experiments demonstrate the effectiveness and viability of our method.

Keywords: Information Retrieval, Spatial Profile, Tag Relatedness, Query Expansion, Event Retrieval, Social Media Retrieval

1. Introduction

The proliferation of web and social media-based photo sharing has not only opened many possibilities but also resulted in new needs and new challenges. Despite recent developments and technological advances within – e.g., web-based media sharing applications, the continuously increasing amount of available information has made the access to these photos still a demanding task. In general, we can address this challenge by allowing the photo collections to be organized

and browsed through the concept of event [1, 2]. Also, most users are generally familiar with searching photo collections using events as starting points. Thus, aiming at supporting the detection and search of event-related photos, we propose an event retrieval framework to improve the state-of-the-art in real-life event retrieval systems in term of retrieval effectiveness.

Focusing on media sharing applications, an *event* refers to "something happening in a *specific* place at a *specific* time, and tagged with a *specific* term" [2]. With an event-retrieval system, we can assume two types of scenarios: (1) A user directly retrieves media resources related to a particular event; and (2) a user uses a given tagged photo representing an event to retrieve other photos related to any similar events from a large image collection. In this work, we mainly focus on scenario (2). Due to their characteristics, pictures in photo sharing applications such as *Flickr*[1] and *Panoramio*[2] are particularly interesting. For example, most of such pictures are accompanied with contextual metadata and other related information added by users, such as *Title*, *Tags*, *Description*, temporal information represented by the picture capture and upload times, and geolocation. Hence, with photo sharing applications in mind, we study how we can exploit contextual metadata to retrieve event-related pictures.

The main goals of this work are (1) to build a framework to extract a set of geographical features from geographical raw data of documents or pictures, and (2) to develop an approach to allow effective retrieval of event-based images. Specifically, we develop a set of geographical features that can capture the characteristics of the geographical distributions of social (or user) tags. Further, we investigate how we can combine these features with the state-of-the-art temporal features to improve the retrieval performance of an event-based image retrieval system. Finally, we explore integrating a machine-learning-based approach with our retrieval system. We study how these features can be used in a query expansion framework. Here, we are especially interested in the contributions of the features on the selection of expansion terms from feedback documents.

To this end, we propose a novel framework that improves the retrieval effectiveness of tag-based image search by including the geographical profile of terms. We have developed a new method for extracting spatial features using information about the geographical distribution of tags. Our main idea is to use such features to characterize the clustering tendency of tag terms and the geographical correlation between two geographical distributions of two tags. Spatio-temporal information retrieval is an established field already. However, existing approaches have mainly been concerned with point-of-interests (POI) extraction [3] and trajectory mining [4]. With the constantly increasing number of geotagged pictures – e.g., in Flickr[3], exploring the raw geographical metadata has become increasingly important.

In summary, the main contributions of this paper are as follows. First, we

---

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

[2]See http://www.panoramio.com/

[3]Around 220M of Flickr pictures are geotagged. See also http://www.flickr.com/map/

2

propose a new robust set of geographical features that can be used (1) to determine the clustering tendency of tags by analysing the geographical structure of their geographical distribution, and (2) to analyse the tag-relatedness between two tags by exploring the correlation between the geographical distributions. To do this, we have developed new measures derived from a well-founded Exploratory Analysis theory from Statistics. More specifically, we adapt the *Ripley's K-function* and *Ripley Cross-K function* (K-function and cross-K function for short) [5] as part of our approach to extract the geographical features. Second, we show how our features can be incorporated in a machine learning-based query expansion model to improve the ability to select good expansion terms. In addition, we demonstrate how these features can be combined with existing document-based approaches and temporal features to achieve improved retrieval performance. Third, through our experimental evaluation we show the effectiveness and practical feasibility of our approach. This includes comparing with both baseline retrieval models and baseline approaches for geo-temporal tag-relatedness. Fourth, we perform a thorough analysis to show the effectiveness of our proposed geographical features – in the afore-mentioned machine learning-based query expansion process.

The rest of this paper is organized as follows. To put our research in a perspective Section 2 provides an overview of approaches related to our work. Section 3 gives an overview of the preliminary theory underlying our approach and defines the problems addressed in this paper. Section 4 presents our proposed geographical features and explains how we extract them. Section 5 describes our framework applying these features in a learning-based re-weighting process for a query expansion model. Section 6 explains our experimental setup. Section 7 presents the results from our experiments. Finally, in Section 8 we conclude the paper and outline our future work.

2. Related Work

In the past decades, detection of events from textual document streams and databases has been treated extensively in the literature [6, 7]. However, although mining and retrieving pictures related to real-life events is an active field, it is still not a fully mature research domain [2, 8, 9]. Most related approaches have been aimed at *extracting* events from different types of datasets. To the best of our knowledge, only few works have addressed the problems of *retrieval* of events in connection to media sharing, and many of these approaches were presented in the Social Event Detection (SED) task at MediaEval [4] [1], where the main objective was to propose event retrieval systems for Flickr pictures.

A research area closely related to ours is pseudo-relevance feedback. Generally speaking, pseudo-relevance feedback refers to techniques to average top-retrieved documents to automatically expand an initial query. It has been

---

[4]http://www.multimediaeval.org/mediaeval2011/

3

studied widely in information retrieval both to extend existing retrieval models [10, 11, 12, 13], and as part of query expansion frameworks [14, 15]. Specifically, Lavrenko and Croft [10] and Zhai and Lafferty [11] propose two methods – the *Relevance Model* and the *Mixture Model*, respectively – to include feedback information in the Kullback-Leibler (KL) divergence retrieval model [16]. The idea is to estimate a new query model using terms in the top-k retrieved documents, also called pseudo-relevant feedback documents to update an existing query model. Experiments have shown that these approaches are indeed able to improve the standard retrieval models with respect to retrieval effectiveness [17]. This has also been the main motivation for including them in our study.

Cao et al. [13] present a classification approach to automatically select *good* expansion terms from a set of candidate terms from the pseudo-relevant documents. To do this, they train a classifier using a set of *good* and *bad* candidate expansion terms represented by feature vectors. Such feature vectors consist of traditional statistical features based on the distribution of the terms both in the whole collection, and the set of (pseudo) relevant documents. Lin et al. [18] propose an extension of this work by applying a learning-to-rank approach for training and classifying the candidate expansion terms. They show that they can improve the retrieval effectiveness by using social annotation from external tagged resources, such as the `de.li.cio.us`[5] social bookmarking web service, as a source for extracting useful expansion terms. The use of social annotation as source for improving the retrieval performance has also previously been investigated by Zhou et al. [19]. These approaches are related to ours in that we also use classification to select good expansion terms. Their main differences with our approach are that none of them applies either temporal, geo-spatial or geo-spatio-temporal features.

As discussed later in this paper, we are interested in investigating the contributions of the temporal characteristics of a term in a pseudo relevance feedback context. Within event retrieval, the usefulness of temporal information is evident. Also within general information retrieval, results from existing work have proven its usefulness. For example, Dakka et al. [20] and Jones and Diaz [21] show how the temporal profile of queries can be used to improve existing retrieval models; whereas Keikha et al. [22] and Whiting et al. [23] propose new temporal-based approaches to improve pseudo relevance feedback based models. Nevertheless, while existing approaches seem to have focused on the temporal aspects only, to fully support event retrieval, we stress the necessity of the spatial profile of social tags, as well as the temporal profile. To the best of our knowledge, the combination of both temporal and spatial features of social tags to improve the retrieval effectiveness has still not been sufficiently investigated. Only few methods – e.g., [24, 25], incorporate temporal and spatial correlation measures to compute term-to-term relatedness. Specifically, Radinsky et al. [24] propose a method to improve the semantic relatedness measure of two terms by capturing the correlation between the temporal profiles of tags and concepts

---

[5]`http://www.delicious.com/`

4

associated with the two terms. Zhang et al. [25], on the other hand, analyse the tag relatedness by using different correlation measures, based on spatial and temporal co-occurrence. In summary, although these approaches are related ours, the way we extract the spatial profiles of tags and apply them in combination with the temporal profile is different. Also, while these approaches were originally developed for textual documents containing much term redundancy that can normally carry the document semantics, image tags usually consist of few unique terms. This makes it more challenging to derive term-based semantic relatedness for image retrieval in general [26], thus further proving the usefulness of our approaches.

3. Preliminary

In this section, we first describe the data our approach is based on and define the problem we address. Thereafter, we give an overview the statistical method our approach are built on.

3.1. Data and Problem Definition

This work mainly focuses on media sharing applications, where resources are usually tagged with terms – i.e., tags, that describe the content of the resources. Such resources may also have information specifying their geographical locations, expressed in longitude and latitude values, and are referred to as geotagged resources.

Let $\mathcal{D} = \{P_1, \ldots, P_N\}$ be a set containing $N = |\mathcal{D}|$ resources. Then, assume that each resource P_i can be annotated with a set of tag T_i, a temporal timestamp t_i and a geotag $\mathbf{g}_i = (latitude, longitude)$, such that $P_i = \{\mathbf{g}_i, \tau_i, T_i\}$, $i = 1, \ldots, N$. Without loss of generality, we assume our resources to be a set geotagged pictures downloaded from Flickr, that may or may not contain all of the above information at the same time. Further, let $\mathcal{E} = \{E_1, \ldots, E_M\}$, $M = |\mathcal{E}|$, be a set of picture clusters $E_i = \{P_{j_1}, \ldots, P_{j_{N_i}}\}$, $i = 1, \ldots, M$, each of which contains images related to the same event. To make our approach as general as possible, we assume that a query picture has only a set of textual tag terms – i.e., it does not contain any geotags or temporal timestamps. This means that following our setup above, a query picture related to an event $E_{i_q} \in \mathcal{E}$ can be expressed as $P_{j_q} = \{T_{j_q}\}$ – i.e., g_{j_q} and τ_{j_q} are not included. For simplicity, we will use $\mathcal{Q}$ to denote a query picture – i.e., $T_{j_q} = \mathcal{Q} = \{q_1, \ldots, q_n\}$, where $n = |\mathcal{Q}|$ and $q_i, i = 1, \ldots, n$ are query tag terms.

The problem addressed in this paper concerns how we can effectively retrieve event-related pictures with a query $\mathcal{Q}$, using only the textual tags. First, we investigate how current state-the-art information retrieval methods perform when applied on our dataset, and let the methods serve as the baseline for our experimental evaluation. Second, we study how a query expansion framework using a set of spatial features summarizing the spatial statistics of the distribution related to a tag, and a set of features defining geographical relatedness between two tags can help us improve the retrieval effectiveness. Third and finally, we compare our method with the baseline methods.

3.2. Exploring Interaction between Spatial Point Patterns

As mentioned in Section 1, our approach is based on geo-spatial features for picture tags. To achieve this, we have to build a *spatial profile* for each tag.

Assume now we have a large dataset $\hat{\mathcal{D}} \subseteq \mathcal{D}$ containing $L = |\hat{\mathcal{D}}|$ geotagged pictures – i.e., $\hat{\mathcal{D}} = \{\hat{P}_1, ..., \hat{P}_L\}$ and $\hat{P}_i = \{\mathbf{g}_i, T_i\}, i = 1, \ldots, L$. Further, let $\mathcal{V} = \{w_1, ..., w_W\}$ be the vocabulary with size $W = |\mathcal{V}|$ of the set of social tags used to annotate $\hat{\mathcal{D}}$. Then, to extract the spatial features from each tag $w_i \in \mathcal{W}$, we analyse the spatial characteristics for the tags using statistical *exploratory analysis* [27].

To be able to use and understand the ideas of exploratory analysis translated into our domain, we need to establish two important concepts our approach is founded on: *picture point processes* and *tag point pattern*. First, considering Flickr pictures as our geotagged web resources, we model the spatial distribution of pictures taken in a specific geographical area as *picture point processes*, which is formally defined as follows:

Definition 3.1 (Picture Point Process). *A Picture Point Process is a point process modelling the spatial distribution of pictures taken in a 2-dimensional study region $\mathcal{R}^2$. So, any realization of the random variable, $\mathcal{P}$, modelling the Picture Point Process is called Picture Point Pattern.*

Second, for each term $w_i \in \mathcal{V}$, we can assume that we have a set of points representing the spatial distribution of the tags in a studied region. With this assumption, we derive a so-called *Tag Point Pattern* from Definition 3.1 as:

Definition 3.2 (Tag Point Pattern). *A Tag Point Pattern $\mathcal{P}_{w_i}$ – or just $\mathcal{P}_i$ for simplicity – for a tag term w_i is a subset of a Picture Point Pattern $\mathcal{P}$, and is a set consisting of the geographical positions of pictures annotated with w_i.*

With these definitions, we can now use statistical exploratory analysis to derive the geo-spatial characteristics of image tags. More specifically, we use a tool called *multivariate Ripley K-function* [5] to get the geo-spatial features from the tags. It is used to study the interaction between two or more spatial point patterns. To help understand how this is done, below is a brief overview of the *multivariate Ripley's K-function*.

3.2.1. Multivariate Ripley's K-Function

The Ripley's K-function is mainly a tool for analyzing completely mapped spatial point patterns data in a two-dimensional space [5]. Hence, it can be used to determine the spatial distribution patterns of objects in spaces.

Let h denote a distance and λ be the intensity of a spatial point pattern, then Ripley's K-function, $K(h)$, is defined as [5]:

$$K(h) = \lambda^{-1}E[\# \text{ other points within distance } h \text{ of an arbitrary point}] \quad (1)$$

The multivariate Ripley's $K_{ij}(h)$ function is a generalization of $K(h)$, and is used to analyse the characteristics of an isotropic spatial point process. It

contains information about clustering and dispersion of point patterns at different distance scales h. The multivariate form aims at answering questions regarding the interaction between two or more point patterns – i.e., bivariate or multivariate point patterns. It is specified as follows [5]: Let λ_i and λ_j be the intensity of the spatial point patterns $\mathcal{P}_i$ and $\mathcal{P}_j$, and assume λ_i and λ_j being constant throughout $\mathcal{R}^2$. Then,

$$
\begin{aligned}
K_{ij}(h) \quad = \quad & \lambda_j^{-1} E[\# \text{ points of type } i \text{ within distance } h \\
& \text{from an arbitrary point } j].
\end{aligned} \tag{2}
$$

Translated to our application, a point here would be a geographical position of a picture. Restricting to the case of two point patterns, we have four K functions: two *self-K functions* $K_{11}(h)$, $K_{22}(h)$, and two *cross-K functions* $K_{12}(h)$, $K_{21}(h)$. The following is most used estimation of $K_{ij}(h)$, as proposed by Ripley [5]:

$$
\hat{K}_{ij}(h) = \frac{1}{\hat{\lambda}_i \hat{\lambda}_j A} \sum_k \sum_l I_h(d_{i_k j_l}), \tag{3}
$$

where $d_{i_k j_l}$ is the distance between a k-th point of type i and a l-th observed point of type j. $I_h(d_{i_k j_l})$ is an indicator, such that $I_h(d_{i_k j_l}) = 1$, if $d_{i_k j_l} \leq h$; and $I_h(d_{i_k j_l}) = 0$, otherwise. $\hat{\lambda}_i = n_i/A$ and $\hat{\lambda}_j = n_j/A$ are the intensity of the two spatial point patterns as the rate between the number of points and the considered area A.

The above four K_{ij} functions are used in the exploratory analysis to study the relationship between two spatial point patterns. For example, in the *independence approach* proposed by Lotwick and Silverman [28], the null model assume that two spatial point patterns are generated by two different and independent spatial processes. Under this independence assumption, with the bivariate form or the cross-K function, $K_{12}(h) = \pi h^2$. From this, the empirical/estimated cross-K function $\hat{K}_{ij}(h)$ calculated on the spatial point patterns, $\mathcal{P}_i$ and $\mathcal{P}_j$, can be compared with the null model to determine the distribution characteristics between the two point patterns as follows: *Attraction*, if $\hat{K}_{ij}(h) > \pi h^2$; *spatial independence*, if $\hat{K}_{ij}(h) = \pi h^2$; and *repulsion*, if $\hat{K}_{ij}(h) < \pi h^2$.

3.2.2. Cross-D Function

As can be derived from the above discussion, Ripley's cross-K functions are useful in characterising the distributions of spatial point patterns. However, the graph of the $\hat{K}_{ij}(h)$ function has normally a parabolic curve, which normally makes it less straightforward to interpret. As a result, a so-called L-function is often used instead. An L-function is defined as

$$
L_{ij}(h) = \sqrt{\frac{K_{ij}(h)}{\pi}}. \tag{4}
$$

Using the same assumption of independence of spatial point patterns as before, we get $L_{ij}(h) = h$. As with the K-function, the empirical values of $L_{ij}(h)$, $\hat{L}_{ij}(h)$, can be used to characterise tag point patterns $\mathcal{P}_i$ and $\mathcal{P}_j$ as

follows: $\hat{L}_{ij}(h) > h$ indicates *attraction* between the point patterns, $\hat{L}_{ij}(h) = h$ shows spatial *independence*, whereas $\hat{L}_{ij}(h) < h$ means *repulsion*. To further facilitate our interpretation, we normalize the cross-L function again to get a so-called *D-function* for two tag point patterns. Based on the empirical cross-L function, the D-function is given by

$$\hat{D}_{ij}(h) = \hat{L}_{ij}(h) - h. \tag{5}$$

Again, we can use $\hat{D}_{ij}$ to characterize the two tag point patterns $\mathcal{P}_i$ and $\mathcal{P}_j$ as follows: $\hat{D}_{ij}(h) > 0$ indicates attraction between $\mathcal{P}_i$ and $\mathcal{P}_j$; $\hat{D}_{ij}(h) = 0$ means we have independence between $\mathcal{P}_i$ and $\mathcal{P}_j$; whereas $\hat{D}_{ij}(h) < 0$ implies repulsion between $\mathcal{P}_i$ and $\mathcal{P}_j$. In the rest of the paper, assuming $\mathcal{P}_i \neq \mathcal{P}_j$, we refer this function to as *cross-D function*.

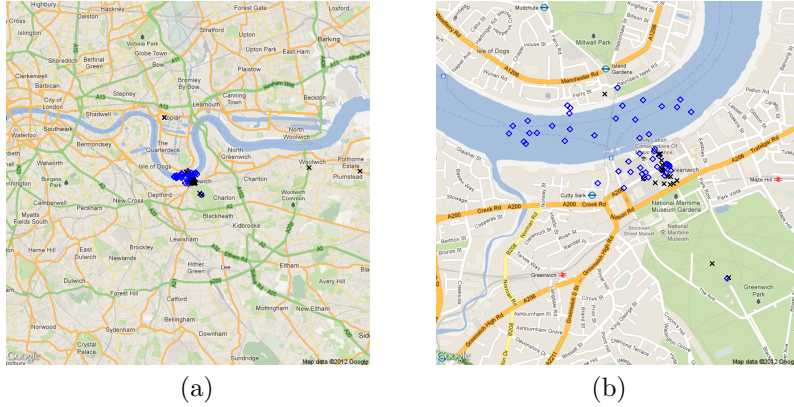**Example [Cross-D function]:** To explain our ideas, assume we have `"Old`

$$\text{(a)} \qquad\qquad\qquad\qquad \text{(b)}$$

Figure 1: Spatial distribution of the Tag Point Patterns related to the tag `Old Naval College` and the tag `University of Greenwich` at two different zooming (a) and (b)

`Royal Naval College"` and `"University of Greenwich"` as two specific tags, both referring to areas in London. Then, consider a cross-L function L_{12} between two tag point patterns, $\mathcal{P}_1$ and $\mathcal{P}_2$, as specified in Definition 3.2, related to these two tags, respectively. A general observation is that the University of Greenwich[6] is located within the area of the Old Naval College[7]. Thus, although the tags are syntactically different, they are connected and refer to the same geographical entity. Within our spatial statistics, this means that pictures tagged with `"Old Royal Naval College"` are spatially *attracted* to pictures tagged with University of Greenwich (See Figure 1a and 1b). To further illustrate this relatedness, consider the corresponding cross-D function $D_{12}(h)$ in Figure 2, varying the values of h between 0 and 2 km. Using the statistical test described above, we can check the validity of our observation about the spatial attraction

---

[6]See `http://en.wikipedia.org/wiki/University_of_Greenwich`
[7]See `http://en.wikipedia.org/wiki/Old_Royal_Naval_College`

among the studied point patterns. As can be seen in Figure 2, the graph of $D_{12}(h)$ (denoted as "observed" in the figure) is greater than the upper envelope (denoted as "higher" in the figure), at all values of h[8]. Hence, based on our distribution "rules" we have *"attraction"* between the two point patterns.
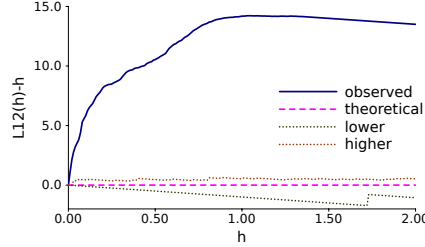

Figure 2: The empirical (*observed*) cross-D function $D_{12}(h)$ of the tag point patterns for `Old Naval College` and the tag `University of Greenwich` as a function of distance (in km). The confidence envelopes (95%) represented by its upper (*higher*) and *lower* borders, for the *theoretical* cross-D function under complete spatial randomness (CSR) are also shown.

In the following, we elaborate on how we extract our set of features based on the spatial characteristics of a tag point pattern, and the interaction between two spatial point patterns derived from the cross-D function.

4. Exploring the Spatial Distribution of Tags

Recall that the primary goal of this work is to find effective ways to exploit the spatial characteristics of tags to improve the retrieval performance. To achieve this, we investigate applying methods from spatial statistics to explore the spatial distribution of tags. In brief, we apply a collection of features derived from the bivariate Ripley's cross-L function presented in Eq. 4 and the Ripley's L-function for a single tag point pattern. To show how we do this, in Section 4.1 we present our method for extraction of general spatial features of tags, including both single and term-to-term spatial features. In Section 4.2, we focus on special tags, such as tags describing point-of-interests, and introduce a method to extract the spatial features for such tags.

4.1. Single and Term-to-Term Spatial Features

We divide the spatial characteristics for tag terms into two main classes: (1) *single-term spatial features*, which determine the aggregation tendency of a single tag spatial point pattern; and (2) *term-to-term spatial similarity* features, which are related to the geographical similarity between the spatial profiles of two considered tags w_i and w_j. In the following we explain how we extract both these features.

---

[8] We computed the envelope by simulating the random labelling with a null model and 99 simulations.

Assume we have a scale interval $S = [0 \ldots R]$ in kilometres, and that we divide the set of the induced intervals into K discrete and equidistant points h_k, $k = 1, \ldots K$. To extract both the single and term-to-term spatial features, we will use the D-function from Section 3.2.1, estimated over this interval.

To capture the clustering tendency of tag point patterns for single terms, in [29] we introduced two features called $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$ estimators. $\hat{I}_{SUM}$ is computed by extracting the positive area within the intersection between the D-function and the curve representing the null hypothesis; whereas $\hat{I}_{MAX}$ is the maximum distance between the D-function and the null hypothesis curve. If we assume that $\hat{D}_i$ represents the estimated value of our D-function for a tag point pattern p_i. Then, for a given tag term w_i,

$$\hat{I}_{SUM}(w_i) = \sum_{k=1}^{K} \left[\frac{\hat{D}_i(h_k)}{\sqrt{Var(\hat{D}_i(h_k))}} \right] \quad \text{and} \tag{6}$$

$$\hat{I}_{MAX}(w_i) = \max_{k=1,\ldots K} \left(\frac{\hat{D}_i(h_k)}{\sqrt{Var(\hat{D}_i(h_k))}} \right). \tag{7}$$

In other words, $\hat{I}_{SUM}(w_i)$ is computed by summing the difference between the D-function and the null hypothesis. A high value of $\hat{I}_{SUM}(w_i)$ means that there is a strong *aggregation* among pictures that are annotated with w_i and connected to the tag point pattern p_i. Further, $\hat{I}_{MAX}(w_i)$ is calculated by estimating the maximum normalized distance between the D-function and the null hypothesis. Hence, it determines the highest positive difference between the K-function of the tag point pattern that the D-function was derived from and the null hypothesis. A high value of $\hat{I}_{MAX}(w_i)$ means that the tag point pattern p_i contributes to a high degree of *clustering*.

For the bivariate case, we can do similar estimation of the attraction tendency of two tag point patterns as follows:

$$\hat{I}_{SUM}(w_i, w_j) = \sum_{k=1}^{K} \left[\frac{\hat{D}_{ij}(h_k)}{\sqrt{Var(\hat{D}_{ij}(h_k))}} \right] \quad \text{and} \tag{8}$$

$$\hat{I}_{MAX}(w_i, w_j) = \max_{k=1,\ldots K} \left(\frac{\hat{D}_{ij}(h_k)}{\sqrt{Var(\hat{D}_{ij}(h_k))}} \right), \tag{9}$$

where w_i and w_j are two specific tags with their tag point pattern p_i and p_j.

Our initial studies have shown the potentials and the usefulness of the above estimators [29]. To apply them in retrieval settings, however, we have to make them more generic, and introduce two new concepts: the *Relative Discrete Positive Area (RDPA)* and *Relative Discrete Maximum Distance (RDMD)*. The main idea is to extend $\hat{I}_{SUM}(w_i)$, $\hat{I}_{SUM}(w_i, w_j)$, $\hat{I}_{MAX}(w_i)$ and $\hat{I}_{MAX}(w_i, w_j)$ by including their behaviour at different scales, and not only at a fixed scale. So, let $\hat{g}_{Sum}$ denote the function representing the relative discrete positive area between the D-function and the null hypothesis in a given scale interval, and assume $\hat{g}_{Max}$ represents the maximum distance within the same considered in-

terval. Then, $\hat{g}_{Sum}$ and $\hat{g}_{Max}$ are computed as follows:

$$\hat{g}_{Sum}(w_i, [h_f, h_g]) = \sum_{k=f}^{g} \left[\frac{\hat{D}_i(h_k)}{\sqrt{Var(\hat{D}_i(h_k))}} \right] \quad \text{and} \tag{10}$$

$$\hat{g}_{Max}(w_i, [h_f, h_g]) = \max_{k=f,\ldots g} \left(\frac{\hat{D}_i(h_k)}{\sqrt{Var(\hat{D}_i(h_k))}} \right), \tag{11}$$

where f and g, with $f < g$, are two indexes related to two points h_f and h_g of the scale interval S. Note that if $f = 1$ and $g = K$, then $\hat{g}_{Sum}(w_i, [h_f, h_g]) = \hat{I}_{SUM}(w_i)$ and $\hat{g}_{Max}(w_i, [h_f, h_g]) = \hat{I}_{MAX}(w_i)$. In conclusion, the generalization captures more features, which divide and summarize the spatial characteristics over more sub-intervals within the original scale interval.

For the bivariate case, we apply a similar approach, and compute $\hat{g}_{Sum}(w_i, w_j, [h_f, h_g])$ and $\hat{g}_{Max}(w_i, w_j, [h_f, h_g])$ by replacing the D_i function with D_{ij}.

4.2. N-order Spatial Features

The features in Eq. 10 and Eq. 11 estimate the deviation of the D-function of the tag point pattern (or the two tag point patterns) from the null hypothesis – i.e., the spatial randomness for a single tag point pattern, and the spatial independence between two tag point patterns, respectively. In addition to this, in our study we observed that for some tags representing point-of-interests, the curve of the D-function related to a tag point pattern tends to be steeper within a short scale sub-interval. Therefore, to also capture such a characteristic, we propose a set of features, called *first order spatial features* that can extract the information on the shape of the curve of the D-function. In Geometry, the *derivative* $f'(x)$ of a source function $f(x)$ can generally be used to determine the slope coefficient of the tangent of the source curve at a point x. Using this as a starting point, our idea is to analyse the derivative function of the D-function for each sub-interval. Since the D-function is discrete over the scale values h_k, $k = 1, \ldots, K$, we apply the discrete equivalent of the derivative function, or more specifically the *forward finite difference* [30], as follows:

$$\hat{D}'_i(h) = \Delta_{l,m} \hat{D}_i(h) = \hat{D}_i(h_l) - \hat{D}_i(h_m), \quad \forall h_l < h_m, \tag{12}$$

where h_l and h_m are two specific scale points. Note that the value of $\hat{D}'_i(h)$ is positive at each scale point where the D-function increases, but negative at all scale points where D-function decreases. Moreover, the higher the positive value of the $\hat{D}'_i(h)$ is, the more the intensity of the function increases. Finally, for the bivariate form of the D-function, we can compute the derivative of $\hat{D}_{ij}(h)$ as $\hat{D}'_{ij}(h)$ by extending Eq. 12 to take into account both w_i and w_j.

Besides determining the slope of the D-function, we are also interested in knowing about the *concavity* of this function at some point x. This gives us more information about the structure or the shape of the function, thus providing us more spatial features. We call such features *second order spatial features*, which we get by doing further derivation of the function $\hat{D}'_i(h)$. As before, we estimate the resulting $\hat{D}''_i(h)$ function by finite differences. This means that we

can extract the spatial features from $\hat{D}'_i(h)$, $\hat{D}'_{ij}(h)$ and $\hat{D}''_i(h)$, $\hat{D}''_{ij}(h)$ using the positive area and the maximum distance estimators in Eq. 10 and Eq. 11.

In the next section, we show how the spatial features presented above are useful, especially when used in a query expansion framework for event-based image retrieval.

5. Query Expansion Framework

Query expansion techniques have been one of the most studied approaches within the information retrieval field since the work by Maron and Kuhns [31]. However, new application areas have made query expansion still needed in order to improve the retrieval effectiveness [32]. Nevertheless, reinventing query expansion techniques is not the focus of this work, per se. Rather, we use it as a framework to evaluate the effectiveness of our proposed method on event-related image retrieval. In this section, we specifically elaborate on how we use our proposed spatial features within a query expansion framework. In addition, we explain how spatial features can be combined with temporal features for better retrieval performance.

5.1. Overview of the Kullback-Leibler Expansion Model

A general query expansion model is a post-processing step in a retrieval system that expand and re-weight an original query with terms from top-k retrieved documents that are assumed to be pseudo-relevant. Such top-k retrieved documents are also called feedback documents.

The Kullback-Leibler divergence-based approach (or just KL-divergence) is a query expansion approach that has been proven effective focusing on retrieval performance [16]. The main idea with KL-divergence is to analyse the term distributions, and maximize the divergence between the distribution of the terms from the top-k retrieved documents and the distribution of terms over the entire collection. The terms chosen for the query expansion are those contributing to the highest divergence – i.e., the terms having the highest so-called KL-scores [14]. To compute the KL-score for a specific term t in the feedback documents, the following equation is used [14]:

$$KL = P_{Rel}(t) \log \left[\frac{P_{Rel}(t)}{P_{Coll}(t)} \right], \tag{13}$$

where $P_{Rel}(t)$ and $P_{Coll}(t)$ are the probability that t appears in the top-k documents and the collection, respectively. $P_{Rel}(t)$ can be estimated by the normalized term frequency of t in the top-k documents, while $P_{Coll}(t)$ can be computed as the normalized frequency of t in the entire collection. This also means that using Eq. 13, terms with low probability in the entire collection and high probability on the retrieved top-k documents have the highest KL-score.

After the expansion terms have been selected, we can proceed to re-weighting the query terms. A classical approach for this is the Rocchio's algorithm [15]

using the Rocchio's Beta equation [33], given by:

$$\hat{w}(t_q) = \frac{tf_{q_{t_q}}}{\max tf_q} + \beta \frac{w(t_q)}{\max w}, \tag{14}$$

where $\hat{w}(t_q)$ denotes the new weight of a term t_q of the query, $w(t_q)$ is the weight from the expansion model – i.e., $KL_{Div}(t_q)$, $\max w$ is the maximum weight from the expanded weight model, $\max tf_q$ is the maximum term frequency in the query, and $tf_{q_{t_q}}$ denotes the frequency of the term in the query.

Since KL divergence is currently one of the state-of-the-art query expansion approaches, it has been the natural baseline approach for our experiments.

5.2. Learning-based Query Expansion Framework

As can be inferred from our discussion in previous sections, the approach proposed in this paper concerns using geo-spatio temporal features in query expansion frameworks. In particular, we develop a learning-based approach to choose good expansion terms and maximize the retrieval performance.
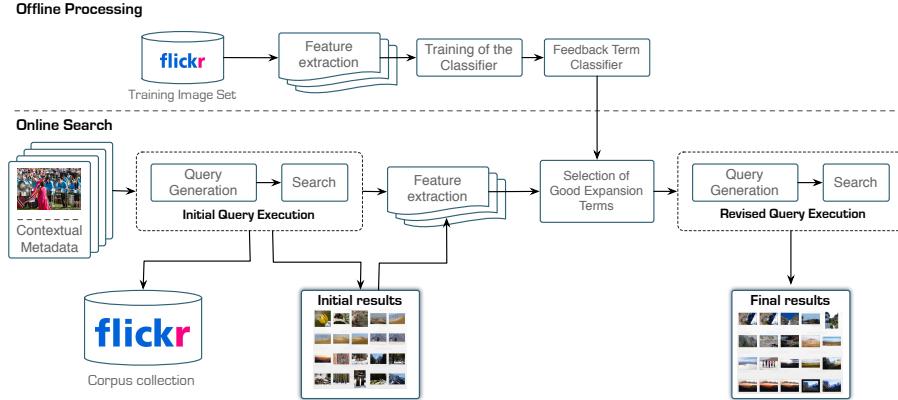

Figure 3: Overview of our supervised learning-based query expansion framework

Figure 3 shows the principle behind our approach. As shown in this figure, the process is divided into two main parts consisting of an offline processing and an online search module. In the offline part, we mainly focus on building a classification model for selecting good candidate terms. In the online part, on the other hand, the main focus is on using the model in a search context to select the actual – previously unseen – candidate expansion terms. Algorithm 1 summarizes the steps in the query expansion (QE) process.

An important question is: how do we select the candidate expansion terms? To answer this question, recall $\mathcal{Q} = \{q_1, ..., q_n\}$ is query consisting of n terms, and $\mathcal{E} = \{e_1, ..., e_m\}$ denote the set of candidate terms for the query expansion process. A *good* candidate expansion term e_i is a term that improves the retrieval performance of the original query $\mathcal{Q}$. Building on a similar principle as the approach in [13], we find e_i by computing the improvements in the average

---
Algorithm 1 Query expansion procedure

 1: **Run query** $\mathcal{Q}$ applying ranking model r
 2: **Get** the set **D** of top-N relevant docs
 3: **Extract** unique tags from **D** and get the candidate expansion term set $\mathcal{E}$
 4: **for** $e_j \in \mathcal{E}$ **do**
 5:     $\mathcal{X} \leftarrow ExtractTermFeats(e_j, \mathcal{Q})$ ▷ Sec. 5.2.1
 6:     $\mathcal{Y} \leftarrow ExtractTemporalFeats(e_j, \mathcal{Q})$ ▷ Sec. 5.2.1
 7:     $\mathcal{Z} \leftarrow ExtractGeoFeats(e_j, \mathcal{Q})$ ▷ Sec. 5.2.1, Sec. 5.2.2
 8:     **Calculate** confidence value $Conf$ ▷ Sec. 5.2.3, Fig. 4
 9:     **Combine KL** score and confidence value $Conf$ in a single score $\rightarrow$
    $KL_{Final}(\mathcal{Q}, e_j)$ ▷ Sec. 5.2.3
10: **end for**
11: **Rank** $e_j \in \mathcal{E}$ terms according to $KL_{Final}(e_j) \rightarrow \mathcal{E}_{Rank}$
12: **Re-build** $\mathcal{Q}$ with the top-k terms from $\mathcal{E}_{Rank} \rightarrow \hat{\mathcal{Q}}$
13: **Run Query** $\hat{\mathcal{Q}}$ by using ranking model r
---

precisions (AP). The idea is as follows. First, for any e_i we compute the average precisions gained from running the original query $\mathcal{Q}$. We call this $AP(\mathcal{Q})$. Then, we calculate $AP(\mathcal{Q} + e_i)$, which is average precision for the query we get from expanding $\mathcal{Q}$ with a specific candidate term expansion e_i. Finally, we find out the improvement in term of average precision from the original query to the expanded on by computing

$$AP_{diff}(\mathcal{Q}, e_i) = \frac{AP(\mathcal{Q} + e_i) - AP(\mathcal{Q})}{AP(\mathcal{Q})}. \tag{15}$$

In other words, a candidate expansion term e_i is a good term if $AP_{diff}(\mathcal{Q}, e_i)$ is positive. Otherwise, it is considered as a bad term. In practice, a threshold θ is used to control the difference value, such that $AP_{diff}(\mathcal{Q}, e_i) > \theta$ means we have a good expansion term, whereas $AP_{diff}(\mathcal{Q}, e_i) < \theta$ means e_i is a bad term. Cao et al. [13] suggest $\theta = 0.005$ as the default threshold. However, because the application area of [13] is mainly different from ours, we decided to do an empirical study with different classification algorithms to find the optimal value of θ (see Section 7.1).

To perform the actual term selection, we define the selection task as a binary classification problem. The main idea is to learn a classifier to discriminate the good expansion terms from the bad ones. Thus, we use Eq. 15 as a basis for the learning process, and to define the positive examples for the classifier. As we will discuss in Section 7, the main advantage with this approach is its effectiveness. However, to achieve good results, selection of features is a crucial task. Below, we discuss how we select the feature set that can be used to represent each expansion term e_i. Thereafter, we explain how we use a classifier to compute a confidence value as function of e_i, as part of the retrieval process. Finally, we present a way to combine this value with the baseline KL-score of the same candidate term to re-rank the set of candidate expansion terms $\mathcal{E}$.

5.2.1. Selecting the Feature Set

Selecting the right set of features has a direct impact on the accuracy of a classification algorithm. This is also one of the reasons we emphasize the importance of studying the effects of selection of features in the end (retrieval) results. To learn a classifier, we define a vector of features for each candidate expansion term e from the top-k retrieved items, given a query $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$. Table 1 lists the features we study in this work. We group them into three sets of features: *term*, *temporal* and *spatial features*. Since our focus is on event-based retrieval, this calls for features beyond those describing document contents only.

Term Features ($\mathcal{X}$): The term features consist of features that are used to characterize a document content. They are chosen based on the hypothesis that terms that contribute to improve the retrieval effectiveness are those being most frequent and distinctive [13]. Existing studies suggest using features related to the distribution of the candidate term e in the feedback documents and the whole collection, and those capturing the co-occurrence of e with the terms in the original query $\mathcal{Q}$ [13, 18]. It is, however, worth noting that these features has mainly been applied in full-text document retrieval, where term redundancy is normal, and thus term frequency would be an important feature. Since a tag generally appears only once for each picture, term frequency as a feature has generally no impact on the classification accuracy. For this reason, in our experiments, our set of term features does not include term frequency but other traditional statistical features such as document frequency (DF) [34]. As part of evaluating our approach, we will use the term features in implementing the baseline approach for our experiments. This will also allow us to assess how well the features suggested in this work improve the retrieval performance.

Temporal Features ($\mathcal{Y}$): Once again, since our focus is on event retrieval, we are interested in capturing how each term in image tags contributes to characterising the images over time periods. Therefore, we need a set of statistical features that represent the temporal distribution of the term in the whole collection. Here, we propose single term features and term-to-term features related to the temporal correlation of the candidate expansion term and the query terms. More specifically, to capture the characteristics of the temporal distribution of a single term, we adopt the concept of *kurtosis* defined as μ_4/μ_2^2, where μ is the mean and μ_j is the j-th central moment. Kurtosis were originally proposed by Jones and Diaz [21] to capture the dynamics of a time series. It can be used to quantify the probability distribution concentrated in peaks of a time series – i.e., the "peakedness". In this work, we propose to measure the peakedness for both a single candidate expansion term e ($KURT1$), and the combination of a candidate expansion term e with a term q_i from the original query ($KURT12$).

In addition to this, we are interested in knowing about the randomness of terms over time. A way to detect such a randomness is to use *autocorrelation* [35]. In general, autocorrelation is computed by finding the statistical correlation between two values of the same variable at a given time t_l and another time t_{l+m}. Such values can, for example, be the number of occurrences of

Feature	Description
Term Features	
$DF0(e)$	Raw document frequency.
$DF1(e)$	Inverse document frequency: $\log(N/DF0)$.
$DF2(e)$	Inverse document frequency smooth: $\log(1 + N/DF0)$.
$DF3(e)$	Probabilistic inverse document frequency: $\log((N - DF0)/DF0)$.
$CoOccSingle(e)$	Co-occurrence with single query terms: $\log(\frac{\sum_{i=1}^{n} C(q_i,e)}{n})$, $n = |\mathcal{Q}|$.
$CoOccPair(e)$	Co-occurrence with pairs of query terms: $\log(\frac{\sum_{(q_i,q_j)\in\mathcal{Q}} C(q_i,q_j,e)}{n})$, $n = |\mathcal{Q}|$.
Temporal Features	
$KURT(e)$, $KURT(\mathcal{Q}+e)$	The kurtosis value of the time series for the pictures annotated with an expansion term e, and for the pictures annotated with both an expansion term e and a query $\mathcal{Q}$, respectively.
$AC(e)$, $AC(\mathcal{Q}+e)$	The autocorrelation value of the time series for the pictures annotated with an expansion term e, and for the pictures annotated with both an expansion term e and a query $\mathcal{Q}$, respectively.
$CC(\mathcal{Q},e)$	The maximum cross-correlation between the time series for the pictures annotated with an expansion term e and the time series for the pictures annotated with a query $\mathcal{Q}$.
Spatial Features	
$\overrightarrow{D}_{Max}(e)$, $\overrightarrow{D}_{Max}(e + \mathcal{Q})$	The vector of the values from the $\hat{g}_{Max}$ function, related to the D-function of the spatial point patterns for pictures annotated with a candidate expansion term e, and with both e and $\mathcal{Q}$, respectively.
$\overrightarrow{D}_{Max}(e,\mathcal{Q})$	The vector of the values from the $\hat{g}_{Max}$ function, related to the cross-D-function between tag point patterns associated to e and the tag point patterns of $\mathcal{Q}$.
$\overrightarrow{D}_{Sum}(e)$, $\overrightarrow{D}_{Sum}(e + \mathcal{Q})$	The vector of the values from the $\hat{g}_{Sum}$ function, related to the D-function of the spatial point patterns for the pictures annotated with a candidate expansion term e, and with both the terms e and $\mathcal{Q}$, respectively.
$\overrightarrow{D}_{Sum}(e,\mathcal{Q})$	The vector of the values from the $\hat{g}_{Sum}$ function, related to the cross-D-function between tag point patterns associated to e and the tag point patterns of $\mathcal{Q}$.

Table 1: A Summary of the Set of Features

a term e at specific times. The hypothesis is that bursty events in a time series normally contribute to a high autocorrelation value [21]. To capture this, we compute the first order *autocorrelation* of a time series for both a single candidate expansion term e ($AC1$), and the combination of a candidate expansion term e with a term q_i from the original query ($AC12$). Finally, to measure the temporal similarity between the time series of two different terms q_i and e, we can apply the *cross-correlation* measure (CC) [36, 24]. Cross-correlation is computed by assessing the correlation of the frequency of q_i and e to measure the relationship between q_i and e. To compute the temporal features, we varied

the time windows or bins from one day to seven days, with which seven days gave the best results. To summarize, we investigate how combining previously proposed temporal features would affect the retrieval performance. These have proven successful in other more general information retrieval approaches, but the way we analyse the effects of their combination within event-related image retrieval haven't been done before.

Spatial Features ($\mathcal{Z}$): As explained in Section 1, the concept of event is strongly related to the spatial dimension – i.e., *geographical location*. We hypothesize that a good expansion term is *spatially correlated* with at least one of the query terms. This is the main reason we study the impact of clustering tendency, with respect to the spatial distribution for the pictures annotated with the candidate expansion terms. As part of this, we compute the spatial features as presented in Section 4.1. For each pair of terms e and q_i, we first extract the set of geographical world tiles $\mathcal{T}_{q_i,e}$ containing spatial points related to documents annotated with q_i, spatial points associated to documents annotated with e, and those related to documents annotated with both q_i and e. Next, we extract a set of six spatial feature vectors from each tile $\mathcal{T}_{q_i,e}$. The first three feature vectors are the vectors computed using $\hat{g}_{Max}$ – i.e., the *relative discrete maximum distance* function for the specified tag point pattern (see Eq. 11), consisting of $\overrightarrow{D}_{Max}(e)$, $\overrightarrow{D}_{Max}(e + \mathcal{Q})$, $\overrightarrow{D}_{Max}(e, \mathcal{Q})$. The second set of feature vectors are based on $\hat{g}_{Sum}$ – i.e., the *relative discrete positive area* function (see Eq. 10), consisting of $\overrightarrow{D}_{Sum}(e)$, $\overrightarrow{D}_{Sum}(e+\mathcal{Q})$, $\overrightarrow{D}_{Sum}(e, \mathcal{Q})$. For all the extracted features, we compute the values of the functions by varying the distance values from 0 to 1 km, with a step of 0.1 km. Finally, for both the resulting first and second derivative of $\hat{g}_{Max}$ and $\hat{g}_{Sum}$, we perform similar operations as described in Section 4.2. Note that as can be inferred from this, the input query $\mathcal{Q}$ used to extract the features may have varying dimensions. However, this does not cause problem but may only affect the number of spatial points used to build the feature vectors, which is, according to Eq. 2 – 5, implicitly decided by the value of the distance scale h.

In this work, we study the impacts of with these features combining the temporal features. In Section 7, we analyse their usefulness and importance with respect to improving the retrieval performance.

5.2.2. Combining the Spatial Features using the World Dataset

Our dataset has been built from a collection of Flickr pictures covering the whole world map. For this reason, the spatial distribution of the pictures is not uniform. To cope with this, we divide the entire world map into a number of tiles. More specifically, we divide the world map into grids with size of one latitude degree and one longitude degree. We span the latitude in the range of $[-180, ..., +180]$ degrees, while the longitude in $[-70, ..., +70]$, instead of $[-90, ..., +90]$ degrees to avoid the Arctic and Antarctic areas, since these areas have normally poor photographic activity. The width of each tile for each (or one) degree of latitude is constant, and has a size of 111 km, while following the latitude values, the tile heights vary from around 0 at the poles to around

17

111 km at the equator. This would give us in total 50,400 tiles. However, to restrict the computation cost, we only consider tiles containing a significant number of pictures – i.e., more than 1,000 pictures. Let such tiles be significant tiles, denoted by $\mathcal{T}$.

With this in mind, we extract the spatial feature vectors for a pair of terms w_i and w_j – e.g., a query term q_i and an expansion term e, as follows. First, let $\mathcal{T}_i = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2} \ldots, \mathcal{T}_{i_N}\}$ be the set of N tiles containing pictures tagged with w_i, and $\mathcal{T}_j = \{\mathcal{T}_{j_1}, \mathcal{T}_{j_2} \ldots, \mathcal{T}_{j_M}\}$ be the set of M tiles containing pictures tagged with w_j. Then, to find a significant tile, $\mathcal{T}_{ij} \in \mathcal{T}$, containing pictures tagged with both w_i and w_j, we merge the two sets $\mathcal{T}_i$ and $\mathcal{T}_j$. Finally, to get the feature vectors, for each tile, $\mathcal{T}_{ij}$, we compute the bivariate D-function and the corresponding estimators for the tag point patterns for both w_i and w_j (see Section 4).

To have a data structure allowing efficient feature extraction operations, we index each tile $\mathcal{T}_l$ as a document composed by the set $\mathcal{W}_{\mathcal{T}_l} = \{w_{l_1}, \ldots, w_{l_{|\mathcal{T}_l|}}\}$ of all tags annotating the pictures from each tile $\mathcal{T}_l$. To do this, we create an inverted index, $\mathcal{I}$, for each tag $w_{l_i} \in \mathcal{W}_{\mathcal{T}_l}$, which we can formulate formally as follows:

$$\mathcal{I} : \{w_i \longrightarrow \{< \mathcal{T}_{i_1}, tf_{\mathcal{T}_{i_1}}(w_i) >, < \mathcal{T}_{i_2}, tf_{\mathcal{T}_{i_2}}(w_i) >, ...\}\}_i \qquad (16)$$

This means that each tag is linked to an inverted list containing the id of the tile and the term frequency, $tf_{\mathcal{T}_{i_1}}(w_i)$, of a tag, w_i.

Selection of the Tiles for Spatial Features Extraction. To select the tiles for spatial feature extraction, we are mainly interested in the tiles containing pictures that are annotated with both at least one term in $\mathcal{Q}$ and a candidate expansion term e. However, to make the spatial features suitable for our classifier, we select only one tile that is most representative to a specific input query $\mathcal{Q}$. We call this the best tile. To do this, we first run $\mathcal{Q}$ on our dataset. Then, we select the first K geotagged pictures from the resulting ranked list. Finally, we select the tile containing the highest TF-IDF-based ranking score. For simplicity, by treating tiles as documents, we index and search them using $Solr$[9] search platform. Thus, the resulting list of tiles is ranked using Lucene scores[10].

5.2.3. Query Re-weighting Process

We now explain how we perform the re-weighting process using the sets of features presented in the previous sections.

Figure 4 shows a part of the term selection and re-weighting process. As depicted in the figure, the *Temporal Classifier* is trained with positive and negative examples using only term and temporal features, while the *Spatio-Temporal Classifier* is trained with instances using the complete set of features. Thus, given a query term q_i and the candidate expansion term e, we first extract the complete set of features. Thereafter, the input instances are classified

---

[9] http://lucene.apache.org/solr/
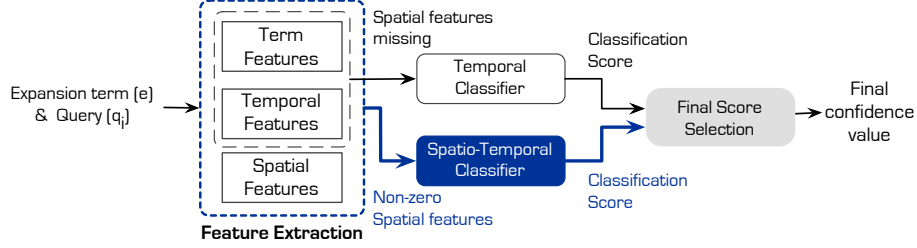[10] http://lucene.apache.org/core/3_6_2/scoring.html

18

Figure 4: Good expansion term selection process through classification

with both of the classifiers. Finally, a *Final Score Selection* module designates the final confidence value. By default, our system produces scores based on all three feature sets. However, we might have a situation in which we do not have geo-tagged pictures that are annotated with both e and any $q_i \in \mathcal{Q}$. Thus, producing the spatial feature vectors from the functions $g_{Max}(\mathcal{Q}+e)$ and $g_{Sum}(\mathcal{Q}+e)$ would be hard. If this happens, then the final score from the Final Score Selection module is based on the term and temporal features only.

We call the final confidence score for good candidate expansion terms from our expansion term selection process $Conf(+|e)$. To produce the final Kullback-Leibler (KL) score for the query expansion process, we combine $Conf(+|e)$ with the term-based KL-score as follows:

$$KL_{Final}(e) = \alpha KL(e) + (1 - \alpha)Conf(+|e). \tag{17}$$

Note that to allow this combination, both $Conf(+|e)$ and $KL(e)$ values are normalized. Here, α is a constant used to decide which component should have the highest contribution. That is, $\alpha = 1$ means that we only apply the regular KL-divergence, whereas $\alpha = 0$ means we rely entirely on the classification modules to choose good expansion terms. Since we are interested in the impacts of our expansion terms to the retrieval performance, we let both components to have equal contributions to the final score – i.e., we use $\alpha = 0.5$.

The confidence value $Conf(+|e)$ is computed based on the idea that both the temporal and the spatio-temporal classifiers give their contributions exploring terms over different dimensions, and that they complement, rather than extend each other. With this in mind, $Conf(+|e)$ can be computed as follows:

$$Conf = \begin{cases} 0, & \text{if } Conf_T < 0.5 \text{ and } Conf_{ST} < 0.5 \\ Conf_T, & \text{if } Conf_T > 0.5 \text{ and } Conf_{ST} < 0.5 \\ Conf_{ST}, & \text{if } Conf_T < 0.5 \text{ and } Conf_{ST} > 0.5 \\ \frac{Conf_T + Conf_{ST}}{2}, & \text{if } Conf_T > 0.5 \text{ and } Conf_{ST} > 0.5 \end{cases} \tag{18}$$

Here, $Conf_T(+|e)$ and $Conf_{ST}(+|e)$ are the confidence values from the Temporal Classifier and Spatio-Temporal Classifier, respectively. The choice of the value 0.5 as threshold is made based on the fact that $0 \leq Conf(+|e) \leq 1$ and that we aim at having final confidence values higher than half the highest possible value. Our experimental results have shown that this is a sensible choice.

19

6. Experimental Setup

In this section we present our dataset and the methodology for our experimental evaluation.

6.1. Dataset

To perform our experiments with tag-based search of event retrieval pictures and to check the feasibility of our approach, we use a large dataset of pictures gathered from Flickr[11] covering a time period from 01.01.2006 to 31.12.2010 and without spatial restrictions. This results in a final dataset consisting of 88,257,485 pictures, of which 18,861,585 pictures are without any tags and around 23.5% are with 1 to 3 tags. For relevance judgement we apply the well-established *Upcoming dataset* [37] as our ground truth. It has also been used previously in other related approaches [38]. Specifically, the Upcoming dataset consists of 270,425 pictures from Flickr, taken between 01.01.2006 and 31.12.2008, each of which belongs to a specific event from the Upcoming event database[12]. The unique number of events are 9,515. Each event is composed by a variable number of images, varying from 1 to 2,398 pictures. This large number and the heterogeneity of the included events are the main advantage of the Upcoming dataset, and the main reason we decided to use it. For generality, we merged the Upcoming dataset with the set of other Flickr pictures.

To perform our experiments, we indexed all image tags using Terrier[13]. As part of the dataset preparation, we perform a preprocessing step consisting of tokenization based on whitespace and punctuation marks; stemming, by using the Porter stemmer algorithm [39]; and English stopword removal.

6.2. Evaluation Methodology

In this section, we briefly explain how we evaluate our approach. First, we present our input query set. Second, we discuss the methods we used as baseline for our experiments. Third, we elaborate on the evaluation metrics we applied.

6.2.1. Input Query Set

We randomly selected set of 150 pictures, one for each event cluster in the Upcoming dataset, and use the tags annotating the pictures as queries. We divide this set of queries into two subsets, one subset consisting of 100 queries that we use to train and evaluate the performance of the classifiers, and another subset consisting of the remaining 50 queries that are used as the test set to evaluate the retrieval effectiveness of the proposed retrieval framework. For completeness, in Table 2, we show some example of input queries used in our experiments.

---

[11]We used Flickr API, `http://www.flickr.com/services/api/`
[12]See `http://www.cs.columbia.edu/~hila/wsdm-data.html`
[13]See `http://www.terrier.org/`

Query	Event Description
  hammermuseum, weswood, ioecho	*Concert of "The Duke Spirit" band at UCLA Hammer Museum, 17th of July, 2008*
  spiritualized, coachella	*Coachella Valley Music and Arts Festival, 26th of April, 2008*
  gibsonamphitheatre, universalcitywalk	*Download 2008 music festival at Gibson Amphitheatre, Los Angeles, 20th of July, 2008*

Table 2: Example of queries extracted from the Upcoming dataset.

6.2.2. Baseline Methods

To assess the effectiveness of the retrieval framework, we compare our models with several baseline methods. First, we perform the searching process by using classical retrieval models, including the *Vector Space Model (VSM)*, *Okapi BM25 (BM25)* [40], and the *Language Model (LM)* for information retrieval – with Jelineck and Dirichlet smoothing. Since BM25 gave the best results in term of effectiveness, we only show the results related to this model. We use the default parameter values $k_1 = 1.2$, $k_3 = 8$ and $b = 0.75$ as baseline for our evaluation. As a query expansion model, we use the basic KL-divergence model (*KL*) and the machine learning approach with the baseline features as proposed Lin et al. [18] as baseline (*KLML*). For simplicity and readability, we only show the results of *KL* since we observed that the MAP values of *KLML* are comparable with the MAP values of *KL*. We compare the baseline approaches with our proposed methods, first by comparing them with a query expansion framework applying a classifier trained with the combination of *terms* and *temporal features* (*KL_T*); and then a framework with a classifier learned with the combination of *terms*, *temporal*, and *spatial features* (*KL_ST*). Note that in addition to the above models, we also experimented with the Mixture Model [11] and the Relevance Model [10], also incorporating the feedback documents in the ranking score computation. However, the results from these experiments were, though comparable, worse than those from the *BM25+KL* query expansion models. Thus, for simplicity we did not include the results from these experiments.

6.2.3. Comparison with Related Work

To have a fair comparison with similar approaches, we implemented the geo-temporal tag relatedness by Zhang et al. [25], which we, from now on, refer to as *ZKYC[25]* for simplicity. As with our approach, with *ZKYC*[25] the similarity between two tags is computed by comparing their temporal and geographical distributions with so-called *geo-spatial*, *temporal*, and *geo-temporal* similarity measures. First, they quantize the world map (space) into m tiles of 1 degree, and the time into n temporal bins of two weeks. Then, they extract the tag features based on the three measures using vectors of numbers of users applying a tag in each bin. This means that for a specific tag the geo-spatial feature vector contains m elements of numbers of users applying that tag in each bin; the temporal feature vector contains n elements of numbers of users applying the tag in each bin; and the geo-temporal feature vector or matrix contains $m \times n$ elements of the counts of unique users tagging a picture within the geo-temporal bin. All vectors are normalized with l^2-norm. Zhang et al. [25] get the similarities between two tags by computing the euclidean distance between the two corresponding feature vectors.

As can be inferred from this, the main difference between our approach and ZKYC[25] is the geographical and geo-temporal features used and how they are extracted. Specifically, with ZKYC[25], the geographical feature vector related to a tag is static, and representing the distribution of the tag over a single size of bin; whereas basing our approach on the Ripley K-function enable characterizing the geographical distribution of tags over non-fixed geographical scales. As discussed previously, the Ripley K-function also allow us to extract the clustering properties of tags. In our experiments, we pay special attention to how this difference affects the retrieval performance. Specifically, we consider a range between 0 km and 3 km of scales when computing the K-function. According to Zhang et al. [25], the geo-temporal features yielded the best results. For this reason, we only compare our approach using with the one applying the geo-temporal features. To incorporate this relatedness in a retrieval framework and compare it with our approach, we define a ranking equation equivalent to Eq. 17 as $\alpha KL + (1 - \alpha)rel_{geo-temp}$, where α is a constant deciding the contribution of the components, KL is the KL-score and $rel_{geo-temp}$ denotes the geo-temporal tag relatedness score. We tune and select the best value of the parameter α over a set of 50 queries.

6.2.4. Evaluation Metrics

To evaluate the retrieval performance of all the models, we use Mean Average Precision (MAP), a widely used evaluation metric within information retrieval [34]. We compute our MAP values based on 1,000 retrieved documents (images). To make sure that any improvements are statistically significant, we perform paired two-sample one-tailed t-tests at $p < 0.05$ or 95 % confidence interval. Any stated improvements in this paper are all statistically significant, unless otherwise specified.

22

7. Results

In this section we perform two different analyses. First, we study the impact of using our temporal and spatial features on training classification algorithms. Second, we investigate the effectiveness of using the temporal and spatial features in a classifier with an optimal feature selection procedure.

7.1. Classification Accuracy

As part of the process of designing a good classifier for selection of good expansion terms, we investigated which classifier is suitable for our application. We evaluated several existing classifiers with respect to their classification accuracy, and selected the classifier yielding the best accuracy. Specifically, we tested our method using *Naïve Bayes* classifier, *Support Vector Machine* (*SVM*), *C4.5* decision tree (also named *J48*) and *Random Forest*. We used Weka [41] machine learning toolkit with default parameter settings to test the classifiers. The training set was composed by a set of 1,000 terms, equally divided into good and bad terms. These were obtained by randomly selecting the feedback terms from the results of running the queries using the training set.

To perform a thorough evaluation, we calculated the *accuracy*, *precision* and *recall* values for each classifier, with a leave-one-out cross validation. We performed the test for five different training sets that we obtained by selecting a positive and a negative class using different values of threshold θ (see also Section 5). The θ values we selected were 0.001, 0.005, 0.01, 0.05, 0.1, and 0.5.

We summarize the averaged results in Table 3. Here, "+" is the positive class containing the good candidate terms, whereas "−" denotes the negative class holding terms considered bad expansion terms. From these results, we can observe that the general performances of the classifiers are good using the proposed set of features. The overall best result was gained by using Random Forest classifier, with an accuracy of around 95%. The precision for the classification of the good terms was 93% and the recall was as high as 97.56%.

		Precision		Recall	
	Accuracy (%)	+	−	+	−
Naïve Bayes	59.12	0.6102	0.6092	0.6180	0.5644
SVM	69.22	0.6802	0.7072	0.7288	0.6556
C4.5/J48	91.52	0.8870	0.9490	0.9536	0.8768
Random Forest	**94.98**	**0.9288**	**0.9736**	**0.9756**	**0.9240**

Table 3: Comparison of the classification performances. The best scores in each column are type-set boldface.

We now analyse the behaviour of the accuracy value of the four proposed classifier over the different θ values. The results is summarized in Figure 5.

Here, we can observe that the higher the threshold value is, the more the accuracy of the classifier increases. Moreover, both J48 and Random Forest (RF) outperformed the Naïve Bayes (NB) and SVM, with high margin.
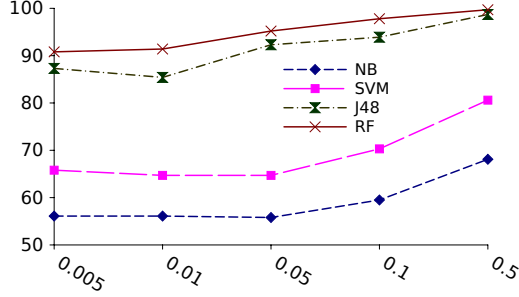

Figure 5: Accuracy of the four different classifiers over the different values of θ

There are several factors that may affect the performance of classification algorithms, which can also be used to explain this. These include randomness and sparsity of the actual dataset, the probability of noises and outliers, the size of the dataset, and the number of independent features – i.e., dimensionality. In addition, many algorithms need calibration to perform well [42]. Focusing on our experiments, the results showed that tree-based classification approaches work best, of which Random Forest is the best classifier. This is because we experimented with a large dataset that has a high degree of randomness and a high number of independent features. This conclusion is also supported by results from other studies [42, 43]. Moreover, the fact that we applied the classifiers with default parameters, with no tuning, played an important role. Since we focus on the ability to treat the classifiers as a "black-box", squeezing out every bit of performance by tweaking the classifiers' parameters is beyond the scope of this work. In conclusion, we choose Random Forest as the base classifier for our framework.

7.2. Retrieval Effectiveness Comparison

As part of our evaluation, we performed a comparative study on the retrieval performance. We compared our approaches with the baseline methods by executing a standard retrieval model – i.e., the *BM25*, and applying the query expansion models described in the previous section – i.e., *Kullback-Leibler (KL)* divergence, in combination with both the temporal features (KL_T) and with the spatio-temporal features (KL_ST). More specifically, we used the Rocchio's framework weighting model, with both the KL divergence model to choose the expansion terms. For each query expansion run, we used the default value of β from [33] – i.e., $\beta = 0.4$, and chose the first n terms of the top-k documents for the *Rocchio's Beta* weighting model. The numbers of pseudo relevant documents, k, were set to 20, 40, 60, 80, 100, and 120, and the numbers of selected terms, n, were 15, 25, 35, 45, and 55. Finally, we performed the query expansion

#Doc	#Term	$BM25$	KL	KL+ZKYC[25]	KL_T	KL_ST
20	15	0.4448	0.4601	0.4614	0.4752^{12}	0.4816^{12}
	25	0.4448	0.4605	0.4626	0.4755^{12}	0.4838^{123}
	35	0.4448	0.4618	0.4638	0.4761^{12}	$\mathbf{0.4838}^{123}$
	45	0.4448	0.4618	0.4634	0.4764^{12}	0.4833^{123}
	55	0.4448	0.4618	0.4624	0.4761^{12}	0.4838^{123}
40	15	0.4448	0.4708	0.4744	0.4786^{12}	0.4870^{123}
	25	0.4448	0.4714	0.4738	0.4799^{12}	0.4880^{123}
	35	0.4448	0.4705	0.4734	0.4813^{12}	0.4885^{123}
	45	0.4448	0.4726	0.4757	0.4843^{12}	$\mathbf{0.4918}^{123}$
	55	0.4448	0.4717	0.4745	0.4827^{12}	0.4913^{123}
60	15	0.4448	0.4665	0.4674	0.4816^{12}	0.4848^{12}
	25	0.4448	0.4685	0.4696	0.4818^{12}	0.4909^{123}
	35	0.4448	0.4704	0.4733	0.4856^{12}	0.4951^{123}
	45	0.4448	0.4712	0.4721	0.4877^{12}	0.4957^{123}
	55	0.4448	0.4703	0.4731	0.4867^{12}	$\mathbf{0.4957}^{123}$
80	15	0.4448	0.4697	0.4706	0.4803^{12}	0.4894^{123}
	25	0.4448	0.4699	0.4711	0.4847^{12}	0.4935^{123}
	35	0.4448	0.4718	0.4733	0.4862^{12}	0.4951^{123}
	45	0.4448	0.4712	0.4731	0.4884^{12}	0.4979^{123}
	55	0.4448	0.4719	0.4741	0.4890^{12}	$\mathbf{0.5001}^{123}$
100	15	0.4448	0.4613	0.4621	0.4701^{12}	0.4727^{12}
	25	0.4448	0.4611	0.4619	0.4755^{12}	0.4802^{12}
	35	0.4448	0.4634	0.4642	0.4781^{12}	0.4849^{123}
	45	0.4448	0.4613	0.4621	0.4803^{12}	0.4879^{123}
	55	0.4448	0.4621	0.4631	0.4820^{12}	$\mathbf{0.4891}^{123}$
120	15	0.4448	0.4592	0.4601	0.4681^{12}	0.4709^{12}
	25	0.4448	0.4589	0.4610	0.4769^{12}	0.4814^{12}
	35	0.4448	0.4606	0.4625	0.4774^{12}	0.4870^{123}
	45	0.4448	0.4589	0.4606	0.4829^{12}	0.4899^{123}
	55	0.4448	0.4595	0.4606	0.4845^{12}	$\mathbf{0.4914}^{123}$

Table 4: MAP Comparison between baseline QE (KL) and QE with classifier learned with baseline+temporal features (KL_T) and baseline+temporal+spatial features (KL_ST). The best scores within each row and each group are type-set boldface. The numbers 1,2,3 in the superscript in the table indicates statistical significance improvements with respect to KL, $KL+ZKYC$[25], KL_T, respectively.

baseline model based on the geo-temporal tag similarities as proposed by Zhang et al. [25] – i.e., the $ZKYC$[25] discussed in Section 6.2.3.

Table 4 lists the results from our experiments. As shown, the baseline query expansion method is better than the baseline $BM25$ in all of our tests, with the best MAP improvement of 6.2%. We can also see that ranking the feedback tags using KL and $ZKYC$[25] to select query expansion terms does not significantly improve the effectiveness of the ranking score of KL. This is mainly because

ZKYC[25] captures the feature of a tag distribution on a fixed geo-temporal scale, due to the size of the geo-temporal bin.

Overall, both our proposed query expansion methods outperform both of the baseline methods, for all the combinations of number of documents and number of terms. For *KL_T*, the maximum MAP improvement is 9.7%, while for *KL_ST*, the improvement is 12.4%. Moreover, studying the MAP values, both *KL_T* and *KL_ST* outperform *ZKYC*[25]. As discussed in Section 6.2.3, an important difference between our method and *ZKYC*[25] is the property of geo-temporal attractiveness of terms with respect to scales. Recall that with *ZKYC*[25], the geo-temporal features are extracted using a fixed scale. In contrast, our methods allow extracting the features at different scales, and take spatial attractiveness into account (see Section 3). Because the concentration of pictures normally vary both in time and space, considering spatial attractiveness and scales is important. The above results further confirm this importance. In conclusion, the ability to capture the geo-temporal attractivenesses of terms at different geo-temporal scales leads to improved retrieval performance.

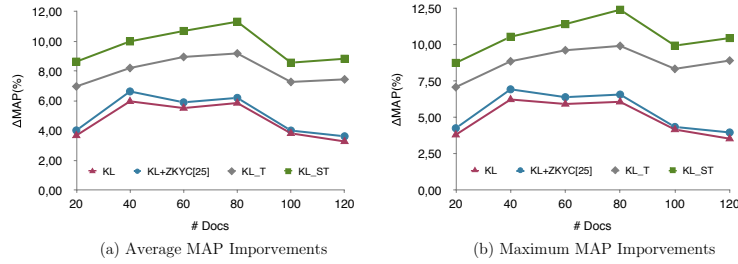(a) Average MAP Improvements   (b) Maximum MAP Improvements

Figure 6: Comparison of MAP improvements as function of feedback documents

In Figure 6, we summarize the improvements of MAP compared to *BM25*, while varying the numbers of feedback documents (k). Specifically, in Figure 6a, for each method, we first take the average MAP values for different numbers of terms. Then, we plot the values as function of the number of feedback documents. In Figure 6b, on the other hand, we plot the best MAP values for each method by only taking into account the numbers of feedback documents, independent of the number of terms. As we can observe in both graphs, all the four query expansion methods have similar trends; that is, the search process using each method gains benefits from the query expansion until reaching a specific number of documents – i.e., a breakpoint, and thereafter this benefit decreases. However, the breakpoint for both of our two approaches (*KL_T* and *KL_ST*) is much higher than with both the baseline *KL* and *ZKYC*[25] – i.e., 80 versus 40. The reason for this is that with baseline *KL*, the set of candidate expansion terms are explored by considering only document features, which seems to be too restrictive. Moreover, *ZKYC*[25] does not consider spatial attractiveness and variation in scales.

7.3. Analysis of the Features

In this section, we analyse the effectiveness the temporal and spatial features we have used to learn the classifiers for selection of the expansion terms. The question we want answered is: Do the features we have proposed in this paper contribute to improve the classification accuracy, and which features work best? To ensure comprehensiveness, we perform our analyses using three different widely-used correlation-based feature evaluation methods. More specifically, we use *Information Gain (IG)* [44], *Gain Ratio (GR)* [45] and *Symmetrical Uncertainly (SU)* [46]. Information Gain is given by $IG(C, F) = \mathcal{H}(C) - \mathcal{H}(C|F)$, where $\mathcal{H}(C)$ is the entropy of a class C and $\mathcal{H}(C|F)$ is the entropy of the class, given a feature F. Gain Ratio is the direct extension of Information Gain, which is $GR(C, F) = IG(C, F)/\mathcal{H}(C)$. Symmetrical Uncertainly (SU) evaluates the goodness of a subset of features F by comparing its symmetrical uncertainty with another subset of features [46]. Let $F_{Sub_1} \subset F$ and $F_{Sub_2} \subset F$ such two subsets. Then, $SU(F_{Sub_1}, F_{Sub_2}) = IG(F_{Sub_1}, F_{Sub_2})/(\mathcal{H}(F_{Sub_1}) + \mathcal{H}(F_{Sub_2}))$. As before, we use Weka to implement of the feature selection methods.

Table 5, 6 and 7 report the IG, GR and SU scores, respectively, for the features we used in our classification of good and bad expansion terms. They show which features are the best using the *baseline and temporal features* compared with applying *baseline, temporal and spatial features*.

Baseline+Temporal		Baseline+Spatial+Temporal	
Feature	IG Score	Feature	IG Score
$coOccSingle_{Whole}$	0.104	$AC2$	0.204
CC	0.066	$RDPA2_{Second}[1]$	0.106
$KURT12$	0.065	$RDMD12[2]$	0.097
$AC12$	0.046	$RDPA12[3]$	0.088
$DF3_{Feedback}$	0.035	$RDMD12[1]$	0.086
$coOccSingle_{Feedback}$	0.034	$RDMD12[3]$	0.086
$coOccPair_{Feedback}$	0.031	$RDPA12_{First}[3]$	0.080
$DF0_{Feedback}$	0.029	$RDPA12[1]$	0.074
$DF1_{Feedback}$	0.025	$DF3_{Feedback}$	0.064
$DF2_{Feedback}$	0.025	$RDMD12_{Second}[1]$	0.063
$DF2_{Whole}$	0.021	$RDPA12_{First}[1]$	0.062
$DF0_{Whole}$	0.021	$RDPA12[2]$	0.062
$DF1_{Whole}$	0.021	$RDMD12[4]$	0.056
$DF3_{Whole}$	0.021	$KURT2$	0.048
$coOccPair_{Whole}$	0.021	$coOccSingle_{Whole}$	0.048
$KURT1$	0.000		
$AC1$	0.000		

Table 5: Comparison of the feature quality based on Information Gain. RDMDs are the features related to the relative discrete maximum distance feature vectors. RDPAs are the features related to the relative discrete positive area vectors. "L" means that we use a cross-L (or cross-D) function. "*First*" and "*Second*" stand for first and second order feature, respectively. "[number]" denotes the number, k, of intervals h_k used to compute the L (or D) function (see Section 4).

Baseline+Temporal		Baseline+Spatial+Temporal	
Feature	RG Score	Feature	RG Score
$KURT12$	0.104	$RDMD12_{Second}[4]$	0.157
$AC12$	0.086	$RDPA12_{First}[4]$	0.116
$coOccSingle_{Feedback}$	0.079	$RDPA12_{Second}[3]$	0.111
$coOccSingle_{Whole}$	0.056	CC	0.109
CC	0.054	$RDMD12_{First}[4]$	0.109
$DF0_{Feedback}$	0.040	$RDPA12[3]$	0.107
$DF3_{Feedback}$	0.040	$RDPA12_{Second}[4]$	0.107
$DF1_{Feedback}$	0.036	$RDMD12[3]$	0.105
$DF2_{Feedback}$	0.036	$RDMDL12_{Second}[2]$	0.098
$coOccPair_{Feedback}$	0.031	$RDMD12_{Second}[2]$	0.093
$coOccPair_{Whole}$	0.025	$RDMD12[2]$	0.093
$DF3_{Whole}$	0.025	$RDMDL12_{Second}[2]$	0.088
$DF2_{Whole}$	0.025	$RDMD12[4]$	0.085
$DF0_{Whole}$	0.025	$RDMDL12[4]$	0.078
$DF1_{Whole}$	0.025	$RDPAL12[2]$	0.078
$KURT1$	0.000		
$AC1$	0.000		

Table 6: Comparison of the feature quality based on Gain Ration. RDMDs are the features related to the relative discrete maximum distance feature vectors. RDPAs are the features related to the relative discrete positive area vectors. "L" means that we use a cross-L (or cross-D) function. "*First*" and "*Second*" stand for first and second order feature, respectively. "[number]" denotes the number, k, of intervals h_k used to compute the L (or D) function (see Section 4).

Focusing on the baseline and temporal features, these results show that with all the three feature selection methods – i.e., IG, GR and SU, none of the features related to the temporal autocorrelation ($AC1$) and kurtosis ($KURT1$) have any impact on the classification. This means that the information about peaks in the temporal distribution of candidate expansion terms does not seem to have any effects on determining good candidate expansion terms. However, the temporal correlation between the distribution of documents annotated with a candidate expansion term and of those annotated with a term from the initial query – i.e., $AC12$, $KURT12$ and $AC12$, seem important, as their scores are within the top-5 highest scores. Similar observation can be made on the cross-correlation – i.e., CC, between the time series of a candidate expansion term and a query term.

Focusing on our set of features – i.e, the *baseline, temporal and spatial features*, on the other hand, our observation is that with all the three feature selection methods, the most important features are those related to the vectors $\overrightarrow{D}_{Max}(\mathcal{Q}, e)$ (called $RDMDL12$ in Table 5, 6 and 7), $\overrightarrow{D}_{Max}(\mathcal{Q} + e)$ (or $RDMD12$), $\overrightarrow{D}_{Sum}(\mathcal{Q}, e)$ (called $RDPAL12$ in Table 5, 6 and 7) and $\overrightarrow{D}_{Sum}(\mathcal{Q} + e)$ (or $RDPAL12$). This means that the features related to the spatial distributions of the documents annotated with both the candidate expansion terms and the query terms, and the spatial correlation between the two tag point patterns have a strong impact on the classification results. As a conclusion, our analysis

Baseline+Temporal		Baseline+Spatial+Temporal	
Feature	SU Score	Feature	SU Score
$KURT12$	0.080	$AC2$	0.107
$coOccSingle_{Whole}$	0.073	$RDPA12[3]$	0.096
$AC12$	0.060	$RDMD12[2]$	0.095
CC	0.059	$RDMD12[3]$	0.094
$coOccSingle_{Feedback}$	0.048	$RDMD12[4]$	0.067
$DF3_{Feedback}$	0.037	$RDPA12_{First}[3]$	0.064
$DF0_{Feedback}$	0.034	$RDPA12[2]$	0.063
$coOccPair_{Feedback}$	0.031	$RDPA12_{First}[1]$	0.062
$DF1_{Feedback}$	0.029	$RDMD12_{Second}[1]$	0.057
$DF2_{Feedback}$	0.029	$RDPA2_{Second}[1]$	0.057
$DF2_{Whole}$	0.023	$RDPA12[4]$	0.054
$DF0_{Whole}$	0.023	$RDPAL12_{First}[2]$	0.054
$DF1_{Whole}$	0.023	$DF3_{Feedback}$	0.053
$DF3_{Whole}$	0.023	$RDMD12[1]$	0.052
$coOccPair_{Whole}$	0.023	$RDMD12_{First}[3]$	0.051
$KURT1$	0.000		
$AC1$	0.000		

Table 7: Comparison of the feature quality based on Symmetrical Uncertainly. RDMDs are the features related to the relative discrete maximum distance feature vectors. RDPAs are the features related to the relative discrete positive area vectors. "L" means that we use a cross-L (or cross-D) function. "$First$" and "$Second$" stand for first and second order feature, respectively. "$[k]$" denotes the number, k, of intervals h_k used to compute the L (or D) function (see Section 4).

confirms the importance of using the spatial correlations between a candidate expansion term and a query term as features for classification of good and bad candidate expansion terms.

8. Conclusion

In this work, we have developed a new approach to effectively retrieve event-based images from typical media sharing applications, such as Flickr. To achieve this, we have developed a new method using a new set of spatial features extracted from image tags to capture the characteristics of the spatial distributions of such tags. This has included applying rigorous statistical exploratory analysis of spatial point patterns to extract the geo-spatial features. As we have shown in this paper, with these features, we have been able to both summarize the spatial characteristics of the spatial distribution of a single term, and identify the similarity between the spatial profiles of two terms. Further, aiming at improving the retrieval performance, we have investigated the gain of combining our geo-spatial features with a set of temporal features from the current state-of-the-art approaches within information retrieval. In addition, we have studied the usefulness of our method by applying our features in a machine-learning-based query expansion framework. More specifically, we have used our spatial and temporal features to select of the best candidate terms for the query expansion process.

The originality of this work lies in the way we extract these features and how we use them to choose the best expansion terms. Our experiments and extensive analyses, including comparison against the baseline methods and existing work, have demonstrated the effectiveness of our approach. These have particularly shown the importance of our proposed spatial features and the feasibility of our approach.

Nevertheless, there are interesting aspects of this work that we have left for further investigation. First, to further explore the usefulness of our spatial features in more general information retrieval settings, we currently study applying our approach on other resources than pictures. Second, in this paper, we have focused on selecting candidate expansion terms as a binary classification problem. As part of making our approach even more generic, we are investigating performing unsupervised selection of expansion terms based on their associated temporal and geo-spatial characteristics. Third, we are exploring the combination of this approach with Open Linked Data usage, such as DBPedia, to further improve the choice of best expansion term candidates.

Acknowledgement

References

[1] S. Papadopoulos, R. Troncy, V. Mezaris, B. Huet, I. Kompatsiaris (Eds.), Social Event Detection at MediaEval 2011: Challenges, Dataset and Evaluation, 2011.

[2] M. Ruocco, H. Ramampiaro, A scalable algorithm for extraction and clustering of event-related pictures, Multimedia Tools and Applications (2012) 1–34.

[3] A. Rae, V. Murdock, A. Popescu, H. Bouchard, Mining the web for points of interest, in: Proc. of ACM SIGIR 2012, ACM, 2012, pp. 711–720.

[4] Z. Yin, L. Cao, J. Han, J. Luo, T. S. Huang, Diversified trajectory pattern ranking in geo-tagged social media, in: SDM, 2011, pp. 980–991.

[5] B. D. Ripley, The second-order analysis of stationary point processes, Journal of Applied Probability 13 (1976) 255–266.

[6] J. Allan, R. Papka, V. Lavrenko, On-line new event detection and tracking, in: Proc. of ACM SIGIR 1998, ACM, 1998, pp. 37–45.

[7] T. Brants, F. Chen, A. Farahat, A system for new event detection, in: Proc. of ACM SIGIR 2003, ACM, 2003, pp. 330–337.

[8] N. Gkalelis, V. Mezaris, I. Kompatsiaris, A joint content-event model for event-centric multimedia indexing, in: Proc. of the IEEE Fourth International Conference on Semantic Computing (ICSC 2010), IEEE CS, 2010, pp. 79–84.

[9] M. R. Trad, A. Joly, N. Boujemaa, Large scale visual-based event matching, in: Proc. of ACM ICMR 2011, ACM, 2011, pp. 53:1–53:7.

[10] V. Lavrenko, W. B. Croft, Relevance based language models, in: Proc. of ACM SIGIR 2001, ACM, 2001, pp. 120–127.

[11] C. Zhai, J. Lafferty, Model-based feedback in the language modeling approach to information retrieval, in: Proc. of ACM CIKM 2001, ACM, 2001, pp. 403–410.

[12] T. Tao, C. Zhai, Regularized estimation of mixture models for robust pseudo-relevance feedback, in: Proc. of ACM SIGIR 2006, ACM, 2006, pp. 162–169.

[13] G. Cao, J.-Y. Nie, J. Gao, S. Robertson, Selecting good expansion terms for pseudo-relevance feedback, in: Proc. of ACM SIGIR 2008, ACM, 2008, pp. 243–250.

[14] C. Carpineto, R. de Mori, G. Romano, B. Bigi, An information-theoretic approach to automatic query expansion, ACM TOIS 19 (2001) 1–27.

[15] J. Rocchio, Relevance Feedback in Information Retrieval, 1971, pp. 313–323.

[16] J. Lafferty, C. Zhai, Document language models, query models, and risk minimization for information retrieval, in: Proc. of the ACM SIGIR 2001, ACM, 2001, pp. 111–119.

[17] Y. Lv, C. Zhai, A comparative study of methods for estimating query language models with pseudo feedback, in: Proc. of ACM CIKM 2009, ACM, 2009, pp. 1895–1898.

[18] Y. Lin, H. Lin, S. Jin, Z. Ye, Social annotation in query expansion: a machine learning approach, in: Proc. of ACM SIGIR 2011, ACM, New York, NY, USA, 2011, pp. 405–414.

[19] D. Zhou, J. Bian, S. Zheng, H. Zha, C. L. Giles, Exploring social annotations for information retrieval, in: Proc. of WWW 2008, ACM, 2008, pp. 715–724.

[20] W. Dakka, L. Gravano, P. Ipeirotis, Answering general time-sensitive queries, IEEE TKDE 24 (2012) 220 –235.

[21] R. Jones, F. Diaz, Temporal profiles of queries, ACM TOIS 25 (2007).

[22] M. Keikha, S. Gerani, F. Crestani, Time-based relevance models, in: Proc. of ACM SIGIR 2011, ACM, 2011, pp. 1087–1088.

[23] S. Whiting, I. A. Klampanos, J. M. Jose, Temporal pseudo-relevance feedback in microblog retrieval, in: Proc. of ECIR 2012, Springer, 2012, pp. 522–526.

[24] K. Radinsky, E. Agichtein, E. Gabrilovich, S. Markovitch, A word at a time: computing word relatedness using temporal semantic analysis, in: Proc. of WWW 2011, ACM, 2011, pp. 337–346.

[25] H. Zhang, M. Korayem, E. You, D. J. Crandall, Beyond co-occurrence: discovering and visualizing tag relationships from geo-spatial and temporal similarities, in: Proc. of ACM WSDM 2012, ACM, 2012, pp. 33–42.

[26] A. Sun, S. S. Bhowmick, K. T. Nam Nguyen, G. Bai, Tag-based social image retrieval: An empirical evaluation, JASIST 62 (2011) 2364–2381.

[27] P. J. Diggle, Statistical Analysis of Spatial Point Patterns, Hodder Arnold Publishers, 2003.

[28] H. W. Lotwick, B. W. Silverman, Methods for analysing spatial processes of several types of points, Journal of the Royal Statistical Society. Series B 44 (1982) 406–413.

[29] M. Ruocco, H. Ramampiaro, Exploratory analysis on heterogeneous tag-point patterns for ranking and extracting hot-spot related tags, in: Proc. of the SIGSPATIAL LBSN 2012, ACM, 2012, pp. 16–23.

[30] G. D. Smith, Numerical solution of partial differential equations: finite difference methods, Oxford University Press, 1985.

[31] M. E. Maron, J. L. Kuhns, On relevance, probabilistic indexing and information retrieval, Journal of the ACM 7 (1960) 216–244.

[32] C. Carpineto, G. Romano, A survey of automatic query expansion in information retrieval, ACM Computing Surveys 44 (2012) 1.

[33] J. Pérez-Agüera, L. Araujo, Comparing and combining methods for automatic query expansion, Advances in Natural Language Processing and Applications Research in Computing Science 33 (2008) 177–188.

[34] R. A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search, Addison-Wesley, New York, 2011.

[35] G. E. Box, G. M. Jenkins, G. C. Reinsel, Time series analysis: forecasting and control, John Wiley & Sons, 2013.

[36] S. Chien, N. Immorlica, Semantic similarity between search engine queries using temporal correlation, in: Proc. of WWW 2005, ACM, 2005, pp. 2–11.

[37] H. Becker, M. Naaman, L. Gravano, Learning similarity metrics for event identification in social media, in: Proc. of WSDM 2010, 2010, pp. 291–300.

[38] Y. Wang, H. Sundaram, L. Xie, Social event detection with interaction graph modeling, in: Proceedings of the 20th ACM MM 2012, ACM, 2012, pp. 865–868.

[39] M. F. Porter, An algorithm for suffix stripping, Program: electronic library and information systems 14 (1980) 130–137.

[40] S. E. Robertson, S. Walker, Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval, in: Proc. of ACM SIGIR 1994, 1994, pp. 232–241.

[41] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: an update, SIGKDD Explorations Newsletter 11 (2009) 10–18.

[42] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: Proceedings of ICML 2006, ACM, 2006, pp. 161–168.

[43] K. Balog, N. Takhirov, H. Ramampiaro, K. Norvaag, Multi-step classification approaches to cumulative citation recommendation, in: Open research Areas in Information Retrieval (OAIR 2013), 2013, pp. 121–128.

[44] Y. Yang, J. O. Pedersen, A comparative study on feature selection in text categorization, in: Proc. of ICML 1997, Morgan Kaufmann Publishers Inc., 1997, pp. 412–420.

[45] G. Forman, An extensive empirical study of feature selection metrics for text classification, The JMLR 3 (2003) 1289–1305.

[46] L. Yu, H. Liu, Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution, in: Proc. of ICML 2003, AAAI Press, 2003, pp. 856–863.