Geo-Temporal Mining and Searching of Events from Web-based Image Collections

Massimiliano Ruocco

Doctoral Thesis
Submitted for the Partial Fulfilment of the Requirements for the Degree of

philosophiae doctor

Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology
Abstract

The proliferation of Web- and social media-based photo-sharing applications have not only opened many possibilities but also resulted in new needs and challenges. They have resulted in a large amount of personal photos being available for public access. One of the most interesting characteristics of these data is that they are surrounded by 1) textual annotations, also called tags, which are intended to describe and categorize, by collective user efforts, the uploaded resources 2) temporal information referring to when a picture has been taken and often by 3) a locational information describing where the picture has been taken. Despite the recent developments and technological advances in Web-based media-sharing applications, the continuously increasing amount of available information has made the access to the photos a demanding task. In general, we can address this challenge by allowing photo collections to be organized and browsed through the concept of events. We also believe most users are familiar with searching photo collections using events as starting points.

Aiming at supporting the detection and search of event-related photos, this thesis proposes a novel framework for extracting pictures related to real-life events from a collection of Web-images by leveraging on their temporal geographical and textual annotations and comparing the proposed approach with existing related state-of-the-art approaches. Second, a set of geographical features is proposed describing the characteristics of the geographical profile of query terms deriving concepts from exploratory analysis. Third, the thesis provides two different tag-based search framework to improve the effectiveness of searching images related to events. The first framework is based on temporal and geographical proximity of query terms to the temporal neighbors of a given timestamped query, while the second framework is based on a novel machine-learning based query expansion method combining the heterogeneous textual, temporal and geographical similarity between query terms and candidate expansion terms for the selection of the expansion terms given a free text textual query.

All the proposed methods have been evaluated by performing extensive experiments on real data gathered from media-sharing applications on the Web. Where possible, comparison with related techniques has been performed to reinforce the validity of the presented approaches. The proposed methods have shown promising results in both the extraction and clustering of event-related images and searching event-related pictures by using metrics from the state of the art.
Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Computer and Information Science, NTNU, Trondheim, with Associate Professor Heri Ramampiaro and with co-supervisors Associate Professor Roger Midtstraum and Associate Professor Randi Karlsen.

The PhD project is a formal part of CAIM (Context Aware Image Management) project, supported by the Research Council of Norway, grant number 176858 under the VERDIKT program.
Acknowledgements

First of all I would like to express my special appreciation and thanks to my supervisor, Associate Professor Heri Ramampiaro at the Department of Computer and Information Science (IDI) for giving me the chance to start this beautiful adventure and for supporting and advising me during these years. He always had time for discussing my research, reading and improving my texts. I also want to express my gratitude to my co-supervisors, Associate Professor Roger Midtstraum and Associate Professor Randi Karlsen.

I want to extend my gratitude to the IDI administrative staff, not only for their great help, but also for their kindness and for a morning smile.

I would also like to thank my colleagues at IDI, for having created an exciting, interesting and multicultural working environment. Thanks for everything: the trips, the lunches, the coffee breaks and the beers together. Without you, the days at NTNU would not have been the same. I miss all of them. A very special thanks goes to my office mate Naimdjion. His help in the first part of my PhD (and not only) has been special. Thanks for the chance of sharing and exchange knowledge, but also for the great fun we had together with our families as well.

All my gratitude is also to my parents, Roberto and Rosaria for supporting me in all my pursuits and for raising me with the love of the study and thinking with freedom. Thanks also to my brother Marco for continuous supporting me even without talking.

Finally, I would like to thank my family. Thanks to Sebastiano, Emma and the small and beautiful new baby that will come. They are my life and what gives me the strength every day. Thanks Ilaria, for your faithful and unconditional support, patience and encouragement. Without you this long journey, from those days in March 2009, could not have been possible. Grazie!
List of Tables

1.1 Summary of the used features over different tasks and different datasets. 8
1.2 Relations between contributions and publications. 21

4.1 Extended Stopword List 71
4.2 Summary of the information related to the San Francisco dataset 79
4.3 Top 20 frequent terms in 4.2(a) San Francisco Dataset and 4.2(b) Barcelona dataset 81
4.4 Precision of event clusters extraction over different granularity of space and time with parameter \(K = 1\) 83
4.5 Precision of event clusters extraction over different granularity of space and time with parameter \(K = 0.8\) 83
4.6 Precision of event clusters extraction over different granularity of space and time with parameter \(K = 0.8\) for Barcelona dataset 85
4.7 Summary of the average number of merged clusters and average number of tags for each event cluster detected for different values of \(K\). These are results from the top 20 event clusters. 88

5.1 Evaluation measures of the runs 96
5.2 Results of the first challenge among the groups 97
5.3 Results of the second challenge among the groups 97

6.1 The scores of the query expansion terms after baseline KL divergence (\(KL\)), and temporal KL divergence (\(KL^2\)). 110
6.2 MAP and RP by querying using the Title field 117
6.3 MAP and RP by querying using the Tag field 118
6.4 MAP and RP by querying using both the Tag and Title field 118
6.5 Short vs. long queries: Percentage improvements using the query expansion model compared to the standard retrieval model, in terms of MAP values. 119
6.6 Percentage of improvement of MAP and RP using different re-weighting model on BM25. The numbers 1, 2 and 3 in superscript in the table indicate the statistical significance improvements on the baseline, KL and KLT re-weighting models, respectively.

7.1 Top-10 tags extracted, ordering the list by $I_{MAX}$
7.2 Top-10 tags extracted, ordering the list by $I_{SUM}$
7.3 Precisions values using the different indexes
7.4 Top-5 tags extracted using rank differences in ordering the list by $I_{MAX}$
7.5 Top-5 tags extracted using rank differences in ordering the list by $I_{SUM}$

8.1 A Summary of the Term Features
8.2 A Summary of the Temporal Features
8.3 A Summary of the Spatial Features
8.4 Classification performance using different classifiers. Best scores within each column are type-set boldface.
8.5 MAP Comparison between baseline QE (KL) and QE with classifier learned with baseline+temporal features (KL_T) and baseline+temporal+spatial features (KL_ST). 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[189], KL_T
8.6 Comparison of the feature quality based on Information Gain
8.7 Comparison of the feature quality based on Gain Ration
8.8 Comparison of the feature quality based on Symmetrical Uncertainly.
List of Figures

1.1 Example of event on last.fm 1.1(a) and an example of Flickr image linked with that event by the machine tag 1.1(b). 5
1.2 Example studies vs. contributions vs. papers 19

2.1 Architecture of a General Information Retrieval System 25
2.2 Different spatial point pattern realizations 43

4.1 Spatial and temporal distribution of the pictures tagged with chelsea flower show and piccadilly circus 63
4.2 Illustration of the ideal assumption 64
4.3 Example of suffix tree built over the string $S=$Papua 67
4.4 Overview of the system 70
4.5 Example of annotation expansion. The original tag set $\mathcal{T}$ is shown on the left side. After the extension with temporal-string and location-string, the set becomes $\mathcal{T}'$. 72
4.6 Suffix tree construction 73
4.7 The area in San Francisco considered for gathering the Flickr photos of the San Francisco dataset 78
4.8 Effect of the cleaning on the tags and terms frequency for San Francisco dataset 81
4.9 P@50s at different value of $K$ over different granularity in space and time (e.g., $D100$ means day as temporal granularity and 100 meters as spatial granularity) 82
4.10 Precision of the algorithm on the Barcelona dataset over different time and space granularities 85
4.11 Min and Max of the average precisions at different time and space granularities 85
4.12 Part of the ranked list of the extracted event clusters with $K = 1$ 86
4.13 Part of the ranked list of the extracted event clusters with $K = 0.8$ 87

5.1 System Overview 93
6.1 Temporal distribution of the tags conference and atmedia and their co-occurrence .............................................. 109
6.2 Response time for our QE executions with the random test queries as function of the dataset size. .................................. 114
6.3 Distribution of number of tags per pictures .................................................. 115
6.4 Comparison between the number of pictures retrieved and the uniqueness of their TagSet .............................................. 116
6.5 MAP values with respect to using different retrieval models ................. 119
6.6 The MAP values as function of the value of γ for three different standard retrieval models as base for the query expansion models. In each figure from (a) - (c), each graph represents a combination of K and n values, expressed as \{K\}_{n}. .................................................. 122
6.7 The MAP values as function of the values of K and n – expressed as \{K\}_{n}, with the three different retrieval models as bases for the different query expansion models. .......................... 123
6.8 The MAP values as function of the of number of documents to re-rank, different retrieval models and different values of K and n ................ 124
6.9 Comparison of MAP and RP values of KL_ST against other query expansion models, as function of the values of K and n (expressed as \{K\}_{n}), using BM25 .................................................. 125

7.1 Spatial distribution of the pictures collected ............................................ 131
7.2 A spatial zoom of the pictures dataset pattern ........................................ 131
7.3 Spatial distribution of the point pattern related to the tag night ................ 131
7.4 K-function for the tag point pattern related to the tag night .................. 132
7.5 Statistics over the dataset ....................................................................... 136
7.6 Rank distance \(R_{diff}\), between the top-50 tags ordered by the two indexes \(I_{SUM}\) and \(I_{MAX}\), respectively. ............................... 138
7.7 Cumulative Average radius of significant clustering for the top-50 tags extracted. \(AVGrad(n)\) is expressed in kilometers. .......... 139
7.8 Cumulative Average radius of significant clustering for the top-n tags extracted. \(AVGrad(n)\) is expressed in kilometres. ......... 140
7.9 K-function of the tags in Table 7.4 and 7.5 .............................................. 141

8.1 Spatial distribution of the Tag Point Patterns related to the tag Old Naval College and the tag University of Greenwich at two different zoom (a) and (b) .............................................. 150
8.2 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). 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. .............................................. 150
8.3 Overview of the approach used .............................................. 154
8.4 Good expansion term selection process through classification .......... 160
8.5 Accuracy of the four different classifiers over the different values of $\lambda$ 164
8.6 Maximum gain of MAP from the baseline according to different top-k document used in the QE process .............................................. 167
## Contents

Abstract i  
Preface iii  
Acknowledgements v  
List of Tables viii  
List of Figures xi  
Contents xvii  

### I Introduction and Motivation 1

1 Introduction 3  
1.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3  
1.1.1 Events in Media-Sharing Applications . . . . . . . . . . . . . . 4  
1.1.2 Scope of the Thesis . . . . . . . . . . . . . . . . . . . . . . . . . . 10  
1.2 Research Context . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11  
1.3 Research Questions . . . . . . . . . . . . . . . . . . . . . . . . . . . 12  
1.4 Research Method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13  
1.5 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14  
1.5.1 Detection of Event-related Resources [C1] . . . . . . . . . . . . 15  
1.5.2 Spatiotemporal Analysis of Tag Point Pattern [C2] . . . . . . 16  
1.5.3 Tag-based Search of Event-Related Pictures [C3] . . . . . . . 17  
1.6 Included Publications . . . . . . . . . . . . . . . . . . . . . . . . . . 18  
1.7 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20  

2 Background 23  
2.1 Information Retrieval . . . . . . . . . . . . . . . . . . . . . . . . . . 23  
2.1.1 General IR Architecture . . . . . . . . . . . . . . . . . . . . . . . 24  

xiii
2.1.2 Term Weighting ............................................. 29
2.1.3 Retrieval Models ........................................... 30
2.1.4 Use of Relevance Feedback ............................... 35
2.1.5 Retrieval Evaluation ....................................... 37
2.1.6 Open Source Search Engine ............................... 39
2.1.7 Final Remarks .............................................. 40

2.2 Exploratory Analysis of Spatial Point Pattern .......... 40
  2.2.1 Point Process and Point Pattern ....................... 41
  2.2.2 Distributions for CSR Modeling ......................... 42
  2.2.3 CSR Tests .............................................. 45
  2.2.4 Extension to Spatiotemporal Data ....................... 48

3 Related Work .................................................. 51
  3.1 Event Detection within Photo-Sharing Applications .......... 52
  3.2 Event Detection from Textual Documents .................... 53
  3.3 Event Detection and Extraction
    from Microblogs ........................................... 54
  3.4 Event Detection using Visual Features ..................... 55
  3.5 Other Related Approaches ................................ 55

II Mining and Detect Event in Social Media .................. 57

4 Event Mining of Social Media Pictures By Suffix Tree Structure 59
  4.1 Introduction ............................................... 60
  4.2 Problem Definition ........................................ 61
    4.2.1 Input Data .......................................... 62
    4.2.2 Hypothesis and Core Idea ............................. 62
    4.2.3 The Goal of the System ............................... 64
  4.3 Comparison with Previous Approaches ..................... 65
  4.4 The Suffix Tree Clustering Algorithm: An Overview ....... 66
    4.4.1 Suffix Tree Data Structure ........................... 67
    4.4.2 The Clustering Process ................................ 67
  4.5 Event Extraction and Clustering .......................... 69
    4.5.1 Cleaning ............................................ 70
    4.5.2 Annotation Expansion ................................ 71
    4.5.3 Suffix Tree Construction ............................. 72
    4.5.4 Candidate Event Clusters Extraction .................. 74
    4.5.5 Merging Candidate Event Clusters ..................... 75
  4.6 Evaluation ............................................... 77
    4.6.1 Dataset ............................................. 78
7 Spatial Features from Geographical Distribution of Social Tags 127
7.1 Introduction ........................................................................ 127
7.2 Problem Definition .................................................................. 129
7.3 Ranking and Extracting Spatial Hot-Spot Terms ......................... 130
   7.3.1 Case-Control Analysis .................................................. 131
   7.3.2 Proposed Estimators ...................................................... 133
7.4 Experiments ........................................................................... 134
   7.4.1 The Dataset .................................................................. 134
   7.4.2 Pre-Processing Step ...................................................... 136
   7.4.3 Results ........................................................................ 137
7.5 Related Work ......................................................................... 141
7.6 Conclusion ........................................................................... 142

8 Event-Related Image Search: Non-Timestamped Queries 143
8.1 Introduction ........................................................................... 144
8.2 Relation to Other Work .......................................................... 145
8.3 Preliminary ............................................................................ 146
   8.3.1 Data and Problem Definition ......................................... 147
   8.3.2 Exploring Interaction between Spatial Patterns ................ 148
8.4 Spatial Distribution of Tags ..................................................... 150
   8.4.1 Single and Term-to-Term spatial features ......................... 151
   8.4.2 n-order Spatial Features ............................................... 152
   8.4.3 Query Expansion Framework ......................................... 153
8.5 Experimental Setup .................................................................. 161
   8.5.1 Dataset ....................................................................... 161
   8.5.2 Evaluation Methodology ................................................ 161
8.6 Results .................................................................................. 163
   8.6.1 Classification Accuracy .................................................. 163
   8.6.2 Retrieval Effectiveness Comparison ................................... 165
   8.6.3 Analysis of the Features ................................................. 167
8.7 Conclusion ........................................................................... 169

IV Conclusion and Future Work 171

9 Conclusion .............................................................................. 173
   9.1 Contributions of the Thesis .................................................. 173

xvi
9.1.1 Detection of Event-Related Resources .......................... 174
9.1.2 Spatiotemporal Analysis of Tag Point Pattern .................. 174
9.1.3 Tag-based search of event-related pictures ....................... 175
9.2 Answers to research questions ..................................... 175
9.3 Future Work ....................................................... 176

References 178
Part I

Introduction and Motivation
Chapter 1

Introduction

This chapter introduces the main topics of the thesis. It mainly focuses on its motivation and explanation of contributions. Specifically this chapter is organized as follows. Section 1.1 explains the main motivation behind the thesis work. In Section 1.2, the research context is presented. In Section 1.3 the research questions are formulated and presented. The main contributions of the thesis are presented in Section 1.5 and the included publications in Section 1.6. Finally a summary of the content for each chapter is presented in Section 1.7.

1.1 Motivation

Advances in digital technologies and the widespread Web 2.0 paradigm have made media-sharing communities such as Flickr\textsuperscript{1} or Panoramio\textsuperscript{2}, a common place where pictures are freely uploaded and tagged. Many people own mobile phones with a camera and it is a common practice to take pictures and upload these pictures into a media-sharing application. Often these pictures are accompanied by two kinds of metadata: a set of annotations, added by the user that are generally named with the term tag, and camera-specific metadata – i.e., the EXIF data\textsuperscript{3}. Temporal data and locational data are often included in the EXIF data. We call this set of information related to the picture, contextual data. Media-sharing applications and tagging systems [106] normally allow users to upload pictures with their personal tags. These are free text, short and unstructured and pictures are stored

\textsuperscript{1}http://www.flickr.com
\textsuperscript{2}http://www.panoramio.com
\textsuperscript{3}EXIF is a standard of the Camera and Imaging Products Association (CIPA). See also http://www.cipa.jp/english
in a social context. From this point of view, a picture belongs to a specific user, generally defined by a unique ID. Further each user may be linked to other users with whom he/she shares common interests and some resources.

For example, Flickr hosts more than five billion pictures, and users upload around 7,000 images every minute. A lot of its multimedia content represents events. An event is an experience attended by people that can be documented through different media. Users, can share these multimedia documents on different social media platforms in form of pictures – e.g., Instagram, Panoramio and Flickr, video – e.g., Youtube\textsuperscript{4} or short text – e.g., Facebook\textsuperscript{5}, Twitter\textsuperscript{6}. In particular photo collections of image-sharing tools are full of images captured for example during a music concert, a wedding ceremony, an important football match or simply in a social gathering like a dinner between friends. This is natural since people tend to think in terms of events, also because they represent a natural abstraction of the real world. Dividing and grouping media through events is thus helpful for improving the browsing, organizing, searching and consuming the experience of these kinds of data. This is also reflected by the increasing diffusion and use of Web services such as Eventful\textsuperscript{7}, Yahoo! Upcoming (retired in April 2013), and last.fm\textsuperscript{8}, helping users to share and manage all the information related to events, plus the possibility of performing a search of such archive of shared events.

\section*{1.1.1 Events in Media-Sharing Applications}

This section defines the concept of event in social media applications with respect to the following points of views: the related tasks, the multimedia type used to describe the event, and the features used to define it in the different tasks.

\textbf{Tasks}

The concept of event in multimedia has attracted many researchers from different fields and most of the works can be divided in three main tasks, event modeling, event classification and event searching and detection as follows:

- \textbf{Event Modeling}: an event is a structured abstraction of a real-life experience. Modeling an event involves a high variety of aspects such as the temporal duration and the geographical extension of the event, whether it is periodic or aperiodic, and the participation of living and non-living objects in the observed event. In addition, the modeling process must consider

\textsuperscript{4}http://www.youtube.com
\textsuperscript{5}http://www.facebook.com
\textsuperscript{6}http://www.twitter.com
\textsuperscript{7}http://www.eventful.com
\textsuperscript{8}http://www.last.fm
1.1. Motivation

Figure 1.1: Example of event on last.fm 1.1(a) and an example of Flickr image linked with that event by the machine tag 1.1(b).

which sensor data models it, and how to model the interaction between events. Three different aspects of interaction have been identified in the literature, as summarized in [150]: 1) mereological referring to the part-of relation between events [130] that means that an event can be composed of different sub-events, 2) causality, referring to the fact that events can be classified in cause events or effect events [94, 151], 3) correlation, referring to the relationship between two or more events having common cause [157]. The objective of the research in this area task is to define a generalized model for common understanding of events in multimedia. Most of the works re-
lated to the event representation task has been done in the area of semantic Web and is well represented in primary workshops such as DERIVE⁹, co-located with the ISWC¹⁰ conference series, EBMIP 2013¹¹, co-located with the ACM Multimedia 2013 conference, and the 1st Workshop on Social Multimedia and Storytelling¹², co-located with the International Conference on Multimedia Retrieval 2014¹³. In the literature, different event models have been proposed in form of ontologies such as Linked Open Descriptor of Events (LODE) [156] and Event-Model-F [152] supporting geographical and temporal representation and the event relationships described before.

- **Event Classification** refers to the task of assigning a specific class to a detected event from a source dataset. The event classification task in particular refers to categorizing events in local and global [47], periodic and aperiodic [30] and assigning a detected event to a specific category of a given ontology [47]. In [47], the authors analyze the possibility of detecting event from the Flickr dataset and classify them according to categories of YAGO ontology [162]. In the same work, the authors investigate the possibility of classifying the event pictures as local and global with respect to the kind of the event detected. In [174], the authors propose a framework for detecting and grouping a soccer game event from a Twitter dataset and classifying it in different type of soccer events and finally assigning them to a specific team. Finally in [30] the authors propose a method for extracting tags related to events from annotated pictures and classifying them as periodic/aperiodic event tags.

- **Event Searching and Detection**: the task of detecting events from a large collection of different user-contributed multimedia contents, refers to the process of extracting event-related resources and grouping them according to the events they represent. A great effort in such area has been done in the context of MediaEval¹⁴ Workshop, in the task of Social Event Detection (SED) task [120]. Most of the approaches are based on the clustering method over heterogeneous metadata associated to the pictures from the associated dataset [186, 125, 12, 14, 118] employing different techniques to fuse and combine the extracted heterogeneous features. Some approaches also rely on external sources and knowledge databases – i.e., DBPedia, to extract additional and structured information related to events [92, 93, 16].

---
⁹http://derive2013.wordpress.com/
¹⁰http://iswc2013.semanticweb.org/
¹¹http://ebmip.disi.unitn.it/
¹²http://sites.google.com/site/socialmultimediastorytelling/
¹³http://www.icmr2014.org/
¹⁴http://www.multimediaeval.org/
Media Types and Features
The tasks illustrated before have been explored over different media for describing observed real-life events. This section describes different approaches proposed in the literature for tackling tasks over different media, first focusing on image-based collections, second textual-based collections and finally on video collections.

- **Image-based sharing applications.** The MediaEval benchmarking initiative collects most of the prominent works on social event detection. In 2013, for example, the dataset comprises around 400K pictures accompanied with their metadata – i.e., timestamps, location, tags, title and description and others. All the pictures are surrounded by timestamps but not all of them, as in the real world, are provided with locational information. Works such as [186, 125, 20, 12, 38, 176], out of that workshop also make use of this dataset, or previous version of it. Another dataset employed in the task of event search and detection is the Event Media dataset [170], composed by more than 1.7 million pictures gathered from Flickr and associated by around 140.000 of events from last.fm, Eventful and Upcoming repositories. This dataset has been employed in [168, 91, 93] for the task of event searching and detection. Other two datasets, named Upcoming dataset and last.fm dataset have been presented in [14] for the event clustering task. The first is composed by 270.000 pictures associated with around 9.00 events from Upcoming repository, whereas the second consists of around 600.000 pictures associated to around 25.000 events from last.fm. These datasets we used in the experiments in this thesis. Thus they are discussed in more detail later in the thesis. Finally there are other works [116, 165, 47, 119] using their own dataset. In [116], the authors experiment their social event detection method on a dataset of 1.9 million pictures associated to around 11.000 events from last.fm, gathered from Flickr with a process similar to the one in [14]. For the same task, the authors of [165] use a dataset of around 1.000 pictures collected from the smart-phones of different users. For the event classification framework proposed in [47], the authors gathered a collection of 2.6 million pictures from Flickr, associated with around 138.000 Wikipedia pages related to events. Finally, for the event detection framework proposed in [119], the authors perform their experiments on their own dataset of around 200.000 images gathered from Flickr.

As mentioned previously, most of the approaches have been solved by employing clustering algorithms and supervised classifiers using different kinds of features extracted from data. They can be divided in four classes. The first are the **textual** features, extracted from tags, description and title metadata surrounding the picture. They can be term frequency (TF), inverse document frequency (IDF) or TF-IDF. The similarity between these
### Chapter 1. Introduction

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>Papers</th>
<th>Task</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MediaEval2011</td>
<td>74K</td>
<td>Zaharieva et al. [186]</td>
<td>Event Detection</td>
<td>• • • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Petkos et al. [125]</td>
<td>Event Detection</td>
<td>• • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brenner et al. [20]</td>
<td>Event Detection</td>
<td>• • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wang et al. [176]</td>
<td>Event Detection</td>
<td>• •</td>
</tr>
<tr>
<td>MediaEval2012</td>
<td>160K</td>
<td>Zaharieva et al. [186]</td>
<td>Event Detection</td>
<td>• • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dao et al. [38]</td>
<td>Event Detection</td>
<td>• • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bao et al. [12]</td>
<td>Event Detection</td>
<td>• • •</td>
</tr>
<tr>
<td>EventMedia</td>
<td>1.7M</td>
<td>Trad et al. [168]</td>
<td>Event Search</td>
<td>• •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Liu et al. [91]</td>
<td>Event Classification</td>
<td>• • •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Liu et al. [93]</td>
<td>Event Detection</td>
<td>• •</td>
</tr>
<tr>
<td>Upcoming Dataset</td>
<td>270K</td>
<td>Becker et al. [14]</td>
<td>Event Clustering</td>
<td>• •</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paniagua et al. [116]</td>
<td>Event Detection</td>
<td>•</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Firan et al. [47]</td>
<td>Event Classification</td>
<td>•</td>
</tr>
</tbody>
</table>

Table 1.1: Summary of the used features over different tasks and different datasets.

Features can be calculated by cosine similarity or the Jaccard index. The second are the **visual** features such as local features such as scale-invariant feature transform (SIFT) [98], speeded up robust features (SURF) [13] and pyramid histogram of visual Words (PHOW) [18], texture descriptor such as edge histogram descriptor (EHD) [104], color features such as color layout descriptor (CLD) [76] or global features such as color histogram and Gabor features. Here, every picture is represented in the bag of word model (BOW). The third are the **temporal** features that are generally represented by the timestamp included in the EXIF information related to the pictures or in the metadata surrounding them. It is represented as the number of past seconds from the Unix epoch\(^{15}\). The distance is simply measured in terms of the absolute difference in seconds, hours or days. The fourth and last features are the **geographical** features represented as a pair of real values indicating latitude and longitude. Geodesic distance can be used as a dissimilarity measure between two geographical points.

Table 1.1, summarizes the list of work described above and the different features, tasks and dataset used. It is possible to observe that the size of the dataset used span from a minimum of 74,000 to a maximum of 1.7 million images, using different combinations of the four classes of features.

\(^{15}\text{http://en.wikipedia.org/wiki/Unix_time}\)
1.1. Motivation

- **Microblogging services:** In the last few years, on social network websites such as Twitter, users publish an enormous amount of real-time and dynamic data. These kinds of Web services are also named microblogging services, referring to short text messages that a user can post on his/her profile. Every day, on Twitter more than 500 million posts are published, by users, describing their activities, opinions and what is happening around them. The event detection task on this kind of data has been mainly tackled as a real-time problem, and the main challenge has been to process massive streams of data to detect events online [54, 178, 175, 146, 1]. In most of them, three kinds of features are extracted: 1) various kinds of textual features, 2) temporal information, extracted from the tweet metadata, and 3) geographical location of the tweet, in form of the geographical latitude and longitude real values associated to the tweet coming from a device equipped with GPS sensor. In the case of devices without a GPS sensor, the position has been predicted from the geonames included in the tweet. In particular, in [54] the textual features are extracted by using TF and TF-IDF in a vector space model representation of the tweets. The method starts by employing the geographical and temporal features to detect geographical region with unusual spatiotemporal pattern of the tweets, followed by an adaptive version of the K-means clustering over the textual content of the tweets. The extracted clusters are then classified in event- or not-event clusters by exploring the cardinality of the extracted groups. The experiments are performed on a dataset of 3 million tweets, gathered from Twitter, in a 2 month period. In [175], the presented framework for online detection of events, first keeps track of the actual geo-temporal clusters, and then classifies the groups in event- or not-event-clusters by employing a classifier learned on several textual features extracted from the content of each tweet of the cluster. Finally, in the work [146], the authors investigate how to detect an earthquake event from tweets. First a Support Vector Machine (SVM) classifier is employed for each tweet, to detect if it is related to an ongoing earthquake or not. The classifier is learned again on a set of content features extracted from the textual posts of each tweet. Temporal and geographical (approximated from the user location if not explicitly included in the tweet) data extracted from the tweet are used to define a spatiotemporal model able to detect the position of the earthquake.

- **Video-sharing applications:** most effort in the task of event search and detection on videos has been done in the TRECVID multimedia event detection (MED) task\(^1\). This initiative provides benchmarking data in the form of the collection of videos gathered from video-sharing application,

\(^1\)\text{http://www.nist.gov/itl/iad/mig/med13.cfm}
and an event kit. The event kit is the query representing an event, and the objective is to find videos related to the query event. An event kit includes the event name, textual information such as a definition and a description of the event and example of videos and maybe audio representing the event. Most of the prominent works on event search and detection on videos rely on the dataset provided to this challenge.

The approaches for the above tasks can be divided in two groups. In the first group [11, 11, 112, 90, 70], video and events are represented as bag-of-words (BOW) models based on 1) visual features, such as global, local and motion features, 2) textual features as the transcription of the audio of the video or capturing the text in the video sequence by performing automatic speech recognition (ASR) and optical characters recognition (OCR) tools and 3) audio features. The mapping process between the events and videos is mainly performed by learning kernel-based classifiers. In the second group of works [172, 4, 108, 59], the videos are represented by higher level semantic concepts detected from the content of each videos, such as those related to object, scene, people, genre and actions.

### 1.1.2 Scope of the Thesis

The main focus of this thesis is in the domain of the image-sharing applications and in particular considering the task of event detection and search on the Flickr dataset. The work presented in this thesis is based on the following challenges and issues, mainly derived from the literature review:

- In event detection and searching, the approaches in the literature never start with a free text query but generally start from a defined idea and kind of event a user is looking for (as in the SED task in MediaEval). Other approaches are based on visual matching from a query image, limiting the scalability of the method.

- The abundance of geotagged resources give the possibility of easily extracting a geographical profile of terms. By extending the concept of event in the domain of the image-sharing application there is a need to define a method for employing this raw geographical data. The reason of doing this is the creation of a general retrieval model in order to improve the detection and search of event-related resources.

- Web services such as last.fm are used to link events with media (Flickr pictures). They generally focus on public events and not personal events.
1.2 Research Context

- Not all the pictures related to an event in an image-sharing application such as Flickr has a link with a last.fm event – e.g., 5 billion pictures in Flickr against around 2.2 million pictures with a machine tags of last.fm. On the other hand not all the pictures related to an event listed in last.fm are linked to the event itself.

- For public events, even if pictures are grouped in Albums or Groups, not all the pictures related to that event, in the whole Flickr dataset are included into these Albums or Groups.

- Therefore, as it happens for last.fm, the users, through specific machine tags, contribute by linking their personal pictures on Flickr, to a specific last.fm event (see Figure 1.1). At the time this thesis is written, the pictures in Flickr with a last.fm machine tags were around 2.2 million, a very small part of the entire number of images in Flickr.

### 1.2 Research Context

These studies have been carried out as a part of a four year PhD program at the Department of Computer and Information Science, NTNU. The PhD project is a formal part of the CAIM (Context-Aware Image Management) project\(^\text{17}\). CAIM is a research project funded by Research Council of Norway and it is focused on research and the development of tools for Context-Aware image management, where image description, query formulation, retrieval from heterogeneous distributed environments, and ranking are designed for using context information. Important application domains are those requiring image capture and multimodal retrieval in mobile environments. CAIM is performed in cooperation with University of Tromsø, University of Bergen, Munich University of Technology, Univ. of Hawaii at Manoa and Telenor R&D in the areas of data management, image processing, information retrieval, multimedia and mobile systems. The objectives are to research methods and techniques for:

- Dynamic context capture and management
- Multimodal information retrieval algorithms based on visual queries (using current or system selected images)
- Context-data and positioning information
- Context-based ranking and presentation of multimodal information on mobile units

\(^{17}\)http://caim.uib.no
• End-user applications for testing CAIM concepts and algorithms
• Specifications and prototypes systems for next generation mobile units

In the four years, 12.5% of the time was spent on teaching duties in courses TDT4117 - Information Retrieval and TDT4186 - Operating Systems both in Fall 2011 and Fall 2012. As part of the PhD program five courses have been attended and successfully passed: IT8802 - Advanced Information Retrieval, DT8116 - Web Mining in Fall 2009, TDT4215 - Web Intelligence and DT8108 Topics in Information Technology in Spring 2010 and finally DT8114 - PhD Seminar in Computer and Information Science in Spring 2013.

1.3 Research Questions

The Web has increasingly populated resources in different media forms that are geotagged, timestamped and surrounded by text related on the resource itself. A huge amount of these resources also represent real-life events. Starting from the possibility of exploring and leveraging these geographical raw data in synergy with the temporal information associated to a collection of resources and the correlated text, with the objective of improve the searching, browsing and detection of events, the main research questions conducting this research is the following:

• [RQ] - Can a raw geographical and temporal information, surrounding annotated resources, improve browsing, detection and text-based searching of events?

The principal research question can be divided in the following sub-questions:

• [RQ1] - Can raw geographical and temporal information surrounding resources improve the clustering and detection of resources geotagged and timestamped in a dataset?
  The research question aims to find the possibility of employing the raw geographical and temporal data surrounding images and the associated textual tags, for a novel clustering algorithm, or extensions of existing ones for the purpose of detection and grouping images in events from a collection of geotagged and timestamped resources.

• [RQ2] - Can a raw geographical and temporal term profile improve existing retrieval and query expansion models for the task of tag-based search of event-related resources?

18In this thesis "geographical" and "spatial" refer to the same concept
Tag-based search is generally performed for searching textual resources over the tag representing them. It makes use of IR models by extracting textual features. With this research question, the intention is to employ geographical and temporal profiles of tag, derived from an underlying collection of geotagged and timestamped resources, to derive retrieval models and query expansion models, from the state of the art, aiming to improve effectiveness of searching event-related resources.

- **[RQ3]** - *How can raw geographical and temporal data be modeled and cooperate to improve and extend the retrieval performance of a tag-based search of event-related resources?*

  From the previous research question, a main challenge come from the observation that temporal and geographical profiles are heterogeneous characteristics of the terms. Thus this work investigates how to make these characteristics cooperate for the same purpose as specified in RQ2.

- **[RQ4]** - *Can a geographical and temporal tag profile improve tag co-occurrence and tag similarity?*

  Deriving the spatial and temporal profile of terms, gives us the possibility of employing techniques for deriving similarity over these dimensions and analyzing how they can cooperate to calculate term-to-term similarity.

- **[RQ5]** - *How can a geographical and temporal term profile be used to infer semantics to the tags?*

  Exploring the possibility of modeling spatial and temporal patterns related to the terms, this work investigates the possibility of defining indicators that are able to measure semantic characteristics of terms.

### 1.4 Research Method

The research method is based on quantitative research – i.e., systematic and empirical evaluation collecting relevant data and using a statistical framework to test the effectiveness of a given hypothesis. In this context, the research method followed can be summarized in the following steps:

- **Knowledge and State of the Art**: reading related literature, participation in Summer Schools, attending classes and establishing connections and discussing with other researchers are all parts of this step in order to understand the state of the art and acquire knowledge about the trends.

- **Finding the Problem**: the literature review must focus on a set of narrow
sub-fields. It is important to extract the principal problems and challenges in the current approaches.

- **Formulation of Hypothesis**: once the set of problems and challenges are discovered, the need is to find way to improve the existing approaches. In order to do that a set of hypothesis must be formulated.

- **Data Collection**: data collection is a strategic field in order to prepare the base for the evaluation step.

- **Framework Design**: in order to address the hypotheses there is a need to prepare a set of experiments. In order to perform these kinds of experiments it is necessary to design a framework.

- **Experiment and Test**: a set of specific experiments need to be programmed according to the specific purpose.

- **Evaluation**: in order to evaluate the hypotheses after the framework has been designed, there is a need to select and develop the metrics to evaluate the performance of the designed approach and metrics to compare that with a baseline approach.

- **Conclusions**: observations and conclusion need to be extracted from the experiments.

- **Writing Report**: all the steps regarding the research need to be annotated and structured in a report to follow the progress of the research.

The above steps are iterative. This means that for each solved problem, a set of new problems may appear. Then, the steps need to be reiterated.

### 1.5 Contributions

This section summarizes the main contribution of the thesis. Specifically it illustrates the three main contributions listed as C1-C3, in accordance with the research questions presented in Section 1.3 and further divided in other sub-contributions. The contribution are directly derived and related to the list of publications listed in Section 1.6.
1.5. Contributions

1.5.1 Detection of Event-related Resources [C1]

**Improving Clustering and Extraction of Event-Related Pictures by considering geographical and temporal dimension [C1.1]**

We have proposed a mining algorithm to extract and cluster groups of images representing events (i.e., wedding, football matches, parties). The definition of event[2] has been extended in the context of image-sharing tools such as Flickr or Panoramio where pictures are normally surrounded by a set of metadata (textual annotation, temporal and geographical information). The proposed approach is based on the well-known *Suffix Tree Clustering Algorithm*, previously used only for a dataset of textual documents. We extended this work by implementing a new algorithm that was enriched with a more sophisticated block for merging group of images representing the same events. The intuition behind this last refinement block is inspired by the well-know DBSCAN [45] clustering algorithm. We have also provided extended analysis on the dataset. To summarize, the principal contributions are the following: use of geographical and temporal information in order to extract picture semantics, employing a scalable algorithm feasible for large-scale dataset and extension of a clustering algorithm used previously only in datasets of textual document and extension of the previous definition of an event, in the context of media-sharing application.

**Improving Effectiveness of Detection and Grouping of Event-related pictures by using external knowledge sources [C1.2]**

We aim to improve the performance of a retrieval system focusing in particular on the extraction of tagged resources related to events. The domain is composed of a dataset of resources tagged with different kinds of information such as time, location and textual tag. In a real dataset and real queries there are the following challenges: 1) considering that an event is *something happening at a specific time at a specific place* [182] and *tagged with a specific tag* (in the domain of tagged resources), the user might not specify in the query all the semantic levels (i.e., is simply looking for *all* the football matches in a specific city or all the *music events* in a certain location in a large time slice), 2) there can be noise in the tags because the tagging process is subjective and tags may be not representative of the pictures, 3) not all the pictures might be geotagged. For this purpose, we have proposed an event detection framework, demonstrating the effectiveness of using external knowledge sources in the retrieval process – i.e., by using entity names from third party knowledge databases as Wikipedia[^19], incorporated in a query expansion process for query refinement.

1.5.2 Spatiotemporal Analysis of Tag Point Pattern [C2]

The datasets of Flickr pictures that are considered, consist of pictures from photo-sharing tools in which images are associated with metadata. In particular, they include geographical information represented by a pair of real numbers $g = (\text{lat}, \text{lon})$, the latitude and longitude, the temporal information represented by the timestamp, and the set of textual annotation, the tags. Hence, these photos are characterized to be placed in a spatiotemporal domain and then considered has been taken in a certain time and in a certain place. Formally, if we assume each term $t_i$ in the Vocabulary $V$ is constructed from the dataset, we can consider to have a set of $M$ points $s_{1,t_i}, \ldots, s_{M,t_i}$ representing the term distribution in the spatiotemporal domain. The presence of a term $t$ in a certain point in the spatial(-temporal) domain, is represented by the picture tagged with $t$. We call this distribution the tag-point pattern and each part of this distribution may be modeled by a random variable. Note that, the hypothesis is that event-related tags are grouped in spatiotemporal and temporal space, while locational-tag is grouped in the spatial area. This is not a trivial problem since the underlying picture distribution is not homogeneous in both the spaces – i.e., heterogeneous point pattern.

Definition of novel features from Geographical Distribution of tags for extraction of locational tags [C2.1]

In order to capture the tag semantics and define a statistical model for geographical profile we explore the effectiveness of existing spatiotemporal and spatial clustering algorithm to evaluate the cluster tendency of each point pattern. We also investigated the use of statistical methods to analyze the spatial and spatiotemporal regularity of tag-point patterns. As part of this we also performed analysis applying global methods such as Complete Spatial Randomness (CSR) [42]. Using this global method it is necessary to address the following problems: 1) analysis of very large set of point patterns and 2) definition of a measure to evaluate regularity of the point pattern and a measure to compare two or more tag-point patterns. To the best of our knowledge there is still a lack in the application and comparative analysis of Spatiotemporal statistics[164] (used generally in epidemiology) for analyzing geographical distribution of pictures-related tags and geographical profile of tags. For this purpose we proposed a set of robust geographical features derived from the well-founded Exploratory Analysis of spatial point pattern, demonstrated to be effective for ranking and extraction of tags related to a geographical point of interest.

Definition of novel tag relatedness similarity measure improving retrieval effectiveness in QE framework [C2.2]

The derived set of geographical features proposed for comparison of geographical tag profiles, has been used also for analyzing the tag relatedness between two tags.
and incorporating that measure in a query expansion framework. This measure has been shown to improve the effectiveness of the searching process of an event-related pictures retrieval system.

### 1.5.3 Tag-based Search of Event-Related Pictures [C3]

As stated before, social media content repositories, such as Youtube, Flickr, Panoramio and Twitter, where users may share and upload resources such as images – e.g., Flickr, video – e.g., Youtube, or simple textual post – e.g., Facebook or Twitter in the past decades have enjoyed. Users have also the possibility to annotate the uploaded resources with textual metadata, generally called tags and followed eventually by a timestamp and geographical information. Hence it may be useful to index these resources using the tags in order to permit the user to retrieve them by tag. We refer this as **tag-based search**. From this point of view, our contributions are divided in the following three points: 1) Analyzing the effectiveness of different textual metadata for retrieval purpose, 2) Improving tag-based search of event-related images of timestamped queries, 3) Improving tag-based search of event-related images of non-timestamped queries. The following describes these contributions in more depth.

**Analyzing effectiveness of textual metadata for retrieval purpose [C3.1]**  
We analyzed and explored the effectiveness of the different metadata surrounding the image-sharing pictures as queries. To assess its effectiveness and analyze the role of the fields in the metadata, we used different combinations of the textual metadata as queries and document representation. Specifically, we evaluated how Title, Tag and their combination affect the retrieval effectiveness. To do this, we first use title only as a document, then tag only, and finally description only. Thereafter, we test different combinations of these fields as follows: title and tag; and title, tag and description.

**Improving tag-based search of event-related images of timestamped queries [C3.2]**  
We proposed a query expansion model by extending the KL function for ranking. In previous works, the KL function has been demonstrated to be one of the most effective for ranking and selection candidate expansion term in a query expansion framework. We extended it by incorporating temporal proximity between the timestamp of the query and timestamp of retrieved documents, and geographical distribution of query terms. The effectiveness of the proposed framework has been shown to be effective also for the most challenging task of short queries – i.e., from one to three terms.
Improving tag-based search of event-related images of not timestamped queries [C3.3]

We proposed a query expansion method solving the same task presented in C3.2. Here the proposed solution assume a not timestamped query as input. For this purpose we propose a machine learning approach to combine and incorporate heterogeneous information, temporal geographical and textual, of query terms for the selection of the best expansion term.

1.6 Included Publications

This section presents the list of papers published during the PhD work, documenting our contributions above. They follow a temporal and logical sequence. The first two papers (papers 1 and 3) refer to the Web mining problem of extracting and grouping event-related pictures by using the annotated text, geographical and temporal information surrounding them. After that we explored the possibility of extracting and retrieving event-related pictures from collection of pictures not all containing both geographical and temporal information (paper 2). Then, in paper 4 we summarized the work done and we introducing new challenges and issues. The next set of papers are more related to IR problems. We presented frameworks aimed to retrieve event-related pictures from term free query. In particular in paper 5, the problem of searching event-related tagged pictures starting from a timestamped queries. The proposed solution leverage on temporal and geographical proximity from the query and the resources in the dataset. Then, paper 7 extends the paper 5 by including more detailed experiments and further methods. Finally, in paper 8, we proposed a novel framework, for the same task of paper 5, but considering not timestamped query. The proposed solution, is based on a machine learning based framework for query expansion. This incorporates the temporal and geographical features extracted from a distribution of terms and described in the paper 6. In Figure 1.2 we summarize, over a temporal line, first the logical sequence of the included papers and relations between them, and second the sequence of activities completed during the PhD program. This thesis is based on the following list of papers:

Summary: The content of this paper is included in Chapter 4 and is aimed at answering the RQ1.
1.6. Included Publications

Figure 1.2: Temporal and logical overview of research progresses, included publications and contributions of the research work

*Summary:* The content of this paper is included in Chapter 4 and is aimed at answering the RQ3.

*Summary:* The content of this paper is included in Chapter 4 and is aimed at answering the RQ1.

Summary: The content of this paper is partially included in Chapter 4 and mainly summarizes the motivation of the whole research.

Summary: The content of this paper is included in Chapter 7 and is aimed at answering the RQ5.

Summary: The content of this paper is included in Chapter 6 and is aimed at answering the RQ4.

Summary: The content of this paper is included in Chapter 6 and is aimed at answering the RQ4.

Summary: The content of this paper is included in Chapter 8 and is aimed at answering the RQ2, RQ3 and RQ4.

As a summary, Table 1.2 shows the relations between the publications and the set of contributions listed in Section 1.5.

1.7 Thesis Outline

The thesis is composed by four different parts addressing specific topics. Each part is further divided in chapters discussing different aspects of the main topic. These
1.7. Thesis Outline

Table 1.2: Relations between contributions and publications.

<table>
<thead>
<tr>
<th>Contributions</th>
<th>Papers</th>
<th>C1.1</th>
<th>C1.2</th>
<th>C2.1</th>
<th>C2.2</th>
<th>C3.1</th>
<th>C3.2</th>
<th>C3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| parts are organized as follows: |

**Part I** This part comprises the current chapter, describing the main motivation behind the research work and presenting the fundamental concepts useful for the comprehension of that.

- **Chapter 1** presents the motivation and the research process behind the research work presented in this thesis. Therefore it also presents the research questions conducting the whole research work.

- **Chapter 2** introduces the fundamentals of information retrieval and spatial statistics useful and required for understanding the research results presented in the thesis.

**Part II** This part presents the work related to extraction and clustering of event-related resources from media-sharing applications.

- **Chapter 4** describes a framework for extracting and grouping event-related pictures from a dataset of geotagged, timestamped images downloaded from a media-sharing application.

- **Chapter 5** presents the results of the framework presented at Mediaeval 2010, in the Social Event Detection task.

**Part III** This part describes and presents solutions related to the problem of searching events related pictures from databases of resources annotated with textual, temporal and, when present, geographical information.
• Chapter 6 presents a framework for improving the retrieval of event-related pictures, by leveraging on temporal and geographical information of the documents, from a timestamped query.

• Chapter 7 describes a novel set of geographical features describing the characteristic of the geographical tag profile.

• Chapter 8 describes a novel framework, extending existing state of the art in query expansion, for the same task as of Chapter 7 but assuming a not timestamped query.

Part IV  This part contains final conclusion and suggestions for future work.

• Chapter 9 contains the final discussion and overall conclusion related to the contributions of this thesis.
Chapter 2

Background

As discussed in the previous chapter, the main contribution of the thesis is related to the exploration of geographical and temporal data, mainly gathered from photo-sharing application. First for mining and improving the tag-based retrieval of real-life events, and second, more in general for improving existing retrieval models incorporating feedback documents still for the same scope. This chapter introduces some basic concepts that can facilitate the understanding of the content of this thesis. First it presents (Section 2.1) the main ideas behind a retrieval system, and more in detail the components and techniques involved in the process of searching from a dataset and in particular focuses on retrieval models incorporating the feedback documents. Next Section 2.2 focuses on the concepts of geographical and temporal distribution of terms, describing some of the existing statistics techniques for exploring geographical and temporal distribution of points.

2.1 Information Retrieval

Information Retrieval (IR) is a field of Computer Science, mainly dealing with the representation, search and manipulation of collection of electronic documents and other information resources [22]. The goal of an IR system is to retrieve and rank documents from the collection, given a user query.

We can assume a query \( q \) to represent the intent of a user, searching for some kind of information. In an IR system, the retrieved documents are generally ranked according to their relevance to \( q \). This leads to the concept of scoring function that can be formally represented as:

\[
f : (Q, D) \rightarrow \mathcal{R}
\]  

(2.1)
where $Q$ is the set of all possible user queries and $D$ is the dataset of documents on which the IR system relies. The function $f$ assigns a score defining how much the retrieved documents $d_j \in D$ is relevant to the user query $q \in Q$. The concept of query is mainly related to the concept of information need. In other words, a query $q$ often reflects the user’s information need. From a user point of view, this concept is crucial in IR, since it represents the user intent. Although the information need can be more or less complex it should be always possible to translate it into a query or sequence of queries, where a query is a set of selected terms best representing the semantic of the information need.

An IR system should generally support users in searching, browsing and filtering operations from the data collection. Moreover it should provide post-operation on retrieved documents such as grouping and clustering operation. These are operations generally performed to improve the user’s visualization experience. Existing IR systems, also named search engines, are often classified into desktop search engine and Web search engine. Desktop search engine refers to a kind of search tool for finding and browsing content across personal files in the context of users’ personal computers. Most of important operating systems have desktop search features. Examples are the following: 1) *Windows Search* is the desktop search engine incorporated into Windows OS, 2) *Spotlight* is the the one incorporated in *Apple Mac OS X Tiger* and 3) *Unity Dash* is the desktop search engine part of *Ubuntu Desktop*. On the other hand, Web search engine refers to a kind of search tool for finding and browsing across Web contents. The most popular examples are Google\(^1\), Bing\(^2\), Yahoo\(^3\) and Baidu\(^4\).

### 2.1.1 General IR Architecture

A general IR System is composed of different parts producing a variety of output each and working on different scopes. An illustration of a general IR system is shown in Figure 2.1. The two main requirements in building an IR system are efficiency and effectiveness. Efficiency is related to the response time of a user query and the capabilities of processing a user query with minimal request of resources such as CPU time, I/O time, network bandwidth, memory space and disk space [9]. The Indexing Process and the choice of the right indexing structure plays an important role in the satisfaction of this requirement. Effectiveness concerns the retrieval and ranking component of an IR System and is more related to the quality of the retrieved documents with respect to the user need. Hence it is strictly related to the concept

---

\(^1\)http://www.google.com

\(^2\)http://www.bing.com

\(^3\)http://www.yahoo.com

\(^4\)http://www.baidu.com
2.1. Information Retrieval

Figure 2.1: Architecture of a General Information Retrieval System

of relevance of a document (or set of ranked documents) to a given user query. The measurement of effectiveness generally involves user assessors who decide whether a document is relevant or not to a given query. In Section 2.1.5 we present some evaluation measures for assessing effectiveness of a given retrieval model. From Figure 2.1, we can observe that a general IR system architecture consists of the following three main parts:

- Crawling [WCR]
- Indexing [IND]
- Retrieval and Ranking [RR]

Each of the three main components in an IR system are covered by different research area in IR. Most of the research documented in this thesis is related to the retrieval and ranking part. In the following, we briefly describe WCR and IND components of a general IR System architecture, including their issues and open challenges.

**Web Crawling (WCR)**

This component is mainly part of a Web search engine architecture. The main role of a **Web crawler** for a Web search engine is to gather Web pages and its hyperlinks from the World Wide Web for further indexing purpose. The basic version of a Web crawler performs a recursive process starting from a set of Web pages, called **seeds**, and recursively traversing and downloading the Web pages following their
out links. Due to the large size, and the dynamic nature of the World Wide Web, crawling is not a trivial procedure, and many issues and challenges must be tackled to make crawling efficient. In particular a good crawler must balance the properties of coverage, quality and freshness. The first refers to the property of covering a maximum possible number of topics. The second refers to the quality of the Web pages gathered – i.e., most visited pages on the Web. The third refers to how much of the downloaded Web pages are up to date. These properties contradict each other since, for example, a crawler that we want to have a high coverage, would hardly have frequent up-to-date Web pages. Because of practical issues such as limited resources in time and network bandwidth and the properties of the Web, in order to find a good balance between these three properties, the behavior of a Web crawler is generally considered as the result of the combination of the following policies [28]:

- **Selection Policy**: stating the pages that must be downloaded
- **Revisit Policy**: stating when to check for Web pages that has been updated
- **Politeness Policy**: stating how to avoid excessive use of bandwidth of websites
- **Parallelization Policy**: stating how to coordinate the distributed Web crawlers

The success of most of Web search engines depends on the effectiveness of the Web crawlers. Examples of real cases are the following: Microsoft Bing Search Engine rely on BingBot\(^5\) Web crawler, Fast Search & Transfer in the distributed crawler FASTCrawler [136], Yahoo! Slurp\(^6\) for Yahoo! search engine and finally GoogleBot\(^7\) for Google search engine.

**Indexing (IND)**

The indexing process refers to all the procedures related to the creation and maintenance of an index structure. An **index** is a data structure created from the gathered corpus, with the main goal to speed up the search process. Two main data structures have been employed for indexing data: inverted index and suffix tree. They are applied depending on the purposes and requirements of the IR system.

An inverted index is a data structure storing unique terms from a corpus of documents. The set of such terms are called **vocabulary**. An entry, called **posting** is defined for each document containing a word from the vocabulary. Then, we use a posting list to group all the occurrences of a document in the corpus containing

---

\(^5\)http://en.wikipedia.org/wiki/Bingbot

\(^6\)http://en.wikipedia.org/wiki/Yahoo!_Slurp

\(^7\)http://www.google.com/webmasters/bot.html
specific term. This means that the posting list is associated to every term in the vocabulary. In the basic form, each entry is composed by the ID of the documents containing the term associated to the entry and the number of occurrence the term appear in the document. Different extension of the basic inverted index have been proposed in the past in order to improve efficiency of some index operations. In the **full inverted index**, the posting list also store other information such as the position of each occurrence of the term in the document for supporting phrase search and document scoring based on term proximity. Web search engines generally provide user interface for phrase search. For example in the *Google* search engine, queries with double quotes locate documents containing exactly the phrase inside the quotes. Furthermore, with the Boolean retrieval model, to locate documents containing the query terms no additional information are needed. For the vector space model, however, sorting the posting lists based on the *tfidf* weight can facilitate the ranking of the retrieved documents. Further extension of the inverted index has been proposed in recent works. For example in [109], the authors propose to enrich the posting list with incoming hyperlinks for improving retrieval efficiency, while [167, 17] propose extension of the basic inverted index to support semantic search.

*Suffix Tree* is a tree structure mostly used for string representation. It is built with all the suffixes of the string. Due its characteristics, the suffix tree data structure is suitable for operation involving string application such as longest common substring and pattern matching. The main advantage of this structure is its linear time computational cost for construction, insertion and search. This data structure is described in more detail in Section 4.4.1.

**Ranking and Retrieval (RR)**

This part of the process aims to retrieve the documents that are relevant to the user query and then assign a score to each of them according to their relevance to the query. Different blocks can be identified in this phase. The main blocks are the following. First a **matching process**, retrieving the documents related to the user query. Second a **retrieval** model assigning a **ranking score** to each document retrieved according to their relevance to the user intent. There is also a **query expansion** block having the purpose of boosting the performance and effectiveness of the retrieval process.

The **matching process**, is the process of comparison between the user query and the indexed documents. The main goal of this stage is to retrieve documents relevant for the given query. In modern IR systems, the matching process relies on an index and it is generally called **index matching**. Here each term of the query is processed one at a time and a score is eventually calculated. Each document containing at least a term of the query is efficiently extracted, by using the underlying indexing data structure. The efficiency of the matching is based on the fact that only documents
containing at least a query term are processed. The rules for calculating the ranking score of a document according to a given query are defined by the retrieval model.

Both retrieval models and query expansion are relevant parts of this thesis. For this reason it will be described in more detail in two different sections (Section 2.1.3 and Section 2.1.4).

Text Pre-Processing
Both documents to be indexed and queries to be processed in a retrieval system, can be pre-processed by a set of steps. The main goal is to represent the document efficiently in space, in term of storing size, and time, in term of query processing time. The main steps are briefly described as follows:

- **Tokenization** is the process of dividing a text into smaller units called tokens. In the process of constructing an index, each document must be broken in tokens and at the same time removing characters, such as punctuation. A token represents a concept, or more in general a semantic unit which is ready to be processed. There are many issues with tokenization. The main problem concerns on how to select the correct tokens from a sequence of characters, and which characters have been used to split the sentences. In English for instance, the problem is on how to tokenize contractions (e.g. “we’ll”) and hyphenation – i.e., the use of hyphens, such as “-” to join terms in a compound word (e.g. “self-respecting”). The problem of tokenized hyphenated words can be solved by either using heuristic rules, or treating it as a classifier problem. Other languages present other issues. For example in German and Norwegian the tokenizer needs a compound splitter to split compound words (for example “Rindfleischetikettierungsüberwachungsaufgabenübertragungsgesetz”), while in Chinese or Japanese there are no spaces between words. To ensure that a sequence of characters in a document to index will always match with the same sequence in a user query, a good practice to partially solve some of these issues is to use the same tokenizer for processing documents in the indexing process and queries before the retrieval stage.

- **Stopword removal.** This process aims to remove common words from the documents. Terms occurring frequently in the documents collection are not discriminative and thus do not help in matching a query to documents. A **stopword list** composed by the most frequent terms in the collection, and other content-poor words such as pronouns, articles, conjunctions and prepositions are normally used as the basics for the stopward removal. Nevertheless removing stopwords may introduce some weakness in IR systems in terms of lower recall. A typical example is the query “to be or not to be”. A stopword removing process on this query would remove almost all the terms,
reducing the initial sentence into “be” and thus making the retrieval process totally ineffective. Partly due to this issue, modern Web search engines do not perform stopword removal during the indexing process. Moreover, they must often support phrase search queries – i.e., searching for documents containing an exact sentence or phrase.

- **Lemmatization and Stemming.** The goal of this process is to reduce all the morphological versions (inflected or derived) of a word to their base form, called stem. These two processes work in different ways. Lemmatization aims to reduce inflected or derived form of words in their base form but the process is performed by using a vocabulary and performing a morphological analysis of the word. For example the words “are”, “been”, “am” are all lemmatized as the common lemma “be”. The stemming process, on the other hand, is easier and faster and works simply by cutting off the ending part of the word following determined rules and without any knowledge of the context there. The Porter Stemmer [127] algorithm has been shown as one of the most effectives stemming algorithms.

To summarize text pre-processing is fundamental to determine the final set of terms, constituting vocabulary. Its size depends on the applied term processing steps presented above.

### 2.1.2 Term Weighting

The terms of a document can be more or less informative or representative for a certain document. For this reason a numerical value, called term weight can be assigned to each term, according to its informativeness within a document of the collection.

A first basic way is to assign the term frequency $t_{f,t,d}$ representing the number of times a term $t$ occurs within the document $d$. The basic idea is to give more importance to terms frequently appearing in a document. The drawback of this statistic is that it initially considers all the terms in the document of equal importance.

To overcome this drawback, Sparck Jones in [161] proposed the inverse document frequency $idf_t$ of a term $t$ defined as:

$$ idf_t = \log \frac{N}{df_t} $$

(2.2)

where $N$ is the number of documents in the collection, and $df_t$, is the document frequency measure representing the number of documents of the collection containing at least one occurrence of the term $t$. This statistic is based on the idea that
terms occurring too frequently in the collection are not discriminative and thus not useful for retrieval purposes. As can be derived from this the value of $idf_t$ is high for rare terms (with low $df_t$ values) and low for more frequent terms (i.e. those with higher $df_t$ values).

In [147], the authors proposed to combine $tf_{t,d}$ and $idf_t$ statistics in a single weight measure, named $tfidf$ as the product of the term frequency and inverse document frequency as follows:

$$tfidf_{t,d} = tf_{t,d} \times idf_t$$

(2.3)

The main characteristics are the following:

- higher values are given to terms occurring frequently within a small number of documents
- less weight are given to terms occurring rarely in a document
- lower weight are given to terms occurring in many documents

The weights of terms in a document are often stored with the raw term frequency as augmented information into the index structure – i.e., in the inverted index structure it is stored in the posting list related to the term with the raw term frequency and the position of the term in the document. More detailed discussion on the different variants of $tf_{t,d}$, $idf_t$ and $tfidf_{t,d}$ are presented in [9].

### 2.1.3 Retrieval Models

The principle behind most retrieval models is to specify the document and query representation and to define and formalize the concept of relevance between a document and a query by a retrieval function. Retrieval models are generally categorized into **Boolean model**, **vector space model**, **probabilistic models** and **language models**. In most IR models, documents and queries are represented as vector of size $|V|$ – i.e., the number of index terms. The components of each vector are the weights associated to each index terms as follows:

$$d_j = [w_{j1}, \ldots, w_{j|V|}]$$

(2.4)

$$q = [w_{q1}, \ldots, w_{q|V|}]$$

(2.5)

Next, we briefly describe and discuss the retrieval models.

**Boolean Model**

In the Boolean model, the main idea is to find documents matching a query consisting of terms and Boolean operators. Within a document the weight of a term is specified
based whether it is present in the document or not. That is, the document vector has element \( w_{ij} \) such as:

\[
    w_{ij} = \begin{cases} 
        1 & \text{if term } t_i \text{ appear in } d_j \\
        0 & \text{if not}
    \end{cases}
\] (2.6)

In the Boolean queries terms are generally logically combined using Boolean operators such as AND, OR and NOT. Given a Boolean query, the IR system retrieves every document making the query logically true. The drawback of this model is that there is no notion of ranking or partial matching – i.e., a document is simply relevant or not relevant for a query.

**Vector Space Model**

In contrast to the Boolean model, the vector space model (VSM), was developed with the aim at permitting partial matching and ranking of documents given a user query. This means that documents that partially contain the terms of the initial query are also retrieved and ranked. As in the Boolean model, the basic idea behind VSM, originally proposed by Salton et al. [148], is to represent, both document \( d_j \) and query \( q \), as vectors of equal dimension. Here the weight, \( w_{ij} \), associated to the index terms in the document and query vector representation, are greater than zero and generally are the \( tfidf \) weights. To measure the degree of similarity between the vectors representing \( d_j \) and \( q \) a distance between the two vectors is computed. A general way to calculate this distance is to consider the cosine similarity – i.e., the cosine value of the angle between the two vectors representation in the linear space of the index terms as follows:

\[
    \text{sim}(q, d_j) = \frac{d_j \cdot q}{|d_j||q|},
\] (2.7)

where \(| \cdot |\) is the vector norm and \( \cdot \) represents the inner product between the two vectors \( d_j \) and \( q \). The similarity value varies from 0 to 1 depending on the degree of similarity. VSM has been seen as the standard similarity model in early IR systems. Today, it is still considerable as baseline for modern research works. The major advantages with VSM are its simplicity, as well as the possibility to rank documents.

**Probabilistic Models**

In the probabilistic models, the relevance of a document to a given user query is based on the probability ranking principle [138]. It models the probability that a user judges a document \( d_j \) relevant with respect to a query \( q \) and assumes the existence of an underlying ideal set of relevant documents to \( q \) maximizing this probability. The probabilistic model measures the similarity between a document and a user query by the probability of \( d_j \) being relevant to \( q \) as the ratio between the
probability of the document $d_j$ to be relevant to $q$ and the probability of $d_j$ not to be relevant to $q$. Assuming $R$ as the binary random variable for the user judgment with value $R = 1$ for relevant documents and $R = 0$ for non-relevant documents, the idea can be formalized as follows:

$$sim(q, d_j) = \frac{P(R = 1 | d_j, q)}{P(R = 0 | d_j, q)}$$

(2.8)

Substituting $R = 1$ with $R$ and $R = 0$ with $\bar{R}$ and following Bayes’ rules, Equation 2.8 can be written as follows:

$$sim(q, d_j) \sim \frac{P(d_j | R, q)}{P(d_j | \bar{R}, q)}$$

(2.9)

where $d_j$ and $q$ are binary vectors representing absence or presence of an index term into the document or query. Assuming the binary independence assumption, i.e. statistical independence between the index terms, the similarity can be written as follows:

$$sim(q, d_j) \sim \prod_{t_i | w_{ij} = 1} p_{iR} \prod_{t_i | w_{ij} = 0} (1 - p_{iR}) \prod_{t_i | w_{ij} = 1} q_{iR} \prod_{t_i | w_{ij} = 0} (1 - q_{iR})$$

(2.10)

where $p_{iR}$ is the probability of an index term $t_i$ to be present in a randomly selected relevant document and $q_{iR}$ is the probability of the same index term to be present in a randomly selected not relevant document. After taking logarithms, Equation 2.10 is reduced to the following equation:

$$sim(q, d_j) \sim \sum \left( \log \frac{p_{iR}}{1 - p_{iR}} + \log \frac{1 - q_{iR}}{q_{iR}} \right)$$

(2.11)

this is the equation to score and rank documents in the probabilistic model. A way to estimate the probability $p_{iR}$ and $q_{iR}$ has been proposed in [139] as follows:

$$p_{iR} = \frac{r_i}{|Rel|}$$

(2.12)

$$q_{iR} = \frac{n_i - r_i}{N - |Rel|}$$

(2.13)

where $|Rel|$ is the number of relevant documents for the given query and $(N - |Rel|)$ the number of not relevant documents, therefore $r_i$ is the number of documents containing the term $t_i$ and $(n_i - r_i)$ is the number of not relevant documents containing the term $t_i$. In the absence of relevance information to calculate $r_i$ and $R$, an initial guess is to assume $r_i = 0$ and $R = 0$. With this assumption, Equation 2.11 becomes:

$$sim(q, d_j) \sim \sum \log \frac{N - n_i + 0.5}{n_i + 0.5}$$

(2.14)
This last equation can be considered as a form of *idf* weight. It contains in the numerator the value $|N|$ of number of documents and in the denominator the value $|n_{i}|$. In order to produce better estimation of the probability $p_{ir}$ and $q_{ir}$, the process might be further iterated by incorporating the top-k ranked documents and extracting from them the relevant and non-relevant documents. The disadvantage of the above scoring model is the need for a set of iteration to have good results. Moreover, the process for estimating the similarity between document and query does not consider the frequency of the index terms within the documents, but only presence or absence.

To overcome these weaknesses, the BM25 [140, 141], of the family of the Best Match models, has been proposed. This model incorporates in fact term frequency and document length normalization into the ranking model as follows:

$$sim_{BM25}(q, d_{j}) \sim \sum \left( \frac{(k_{1} + 1) \times tf_{ij}}{k_{1} \times ((1 - b) + \frac{b \times length(d_{j})}{avg \_ doclength} + f_{ij})} \right) \times \left( \log \frac{N - n_{i} + 0.5}{n_{i} + 0.5} \right),$$

(2.15)

where $length(d_{j})$ represents the number of tokens of the document $d_{j}$, $avg \_ doclength$ is the average size in the documents of the collection. The parameters $b$ and $k_{1}$ in the equation are tunable. The first, can be in the range $[0,1]$ and the second can assume real values greater than zero. As can be observed the equation takes into account the term frequency $tf_{ij}$ that is scaled by the ratio between $length(d_{j})$ and $avg \_ doclength$, and adjusted with the parameters $k_{1}$ and $b$. Recommended values for these two parameters are $k_{1} = 0.75$ and $b = 1.2$ [72].

**Language Modeling**

**Language Models (LM)** came originally from natural language processing such as machine translation for estimating the probability distribution of the words in a certain language. The idea behind LM for information retrieval is to consider each query $q$ as a sample of a generative model and each document of the collection as a sample of a specific model distribution. In IR problems the simple and efficient unigram language model considering the probability of each term of the language independent to the others is generally used. This means that the order of words is irrelevant. This leads to the bag of words approach where the probability of generating a set of words from a model distribution does not change even if the order of the terms change. This means that the document $d_{j}$ can be seen as a sample generated from the document model $\theta_{d_{j}}$. This means that given a query $q$, any document can be ranked according to the probability of the model $\theta_{d_{j}}$ of generating the query $q$. We call this probability a **query likelihood**. The goal of **query likelihood approach** is to rank the document $d_{j}$ according $P(d_{j}|q)$, the
probability of a document to be relevant to the query $q$. Using Bayes rule:

$$P(d_j|q) = \frac{P(q|d_j) \times P(d_j)}{P(q)}$$  \hspace{1cm} (2.16)

where $P(q)$ is the query probability that is equal for all the documents and then removed; and $P(d_j)$ denotes the probability of each document that can be considered drawn by uniform distribution and can also be removed. In conclusion $P(d_j|q) \sim P(q|d_j)$. This means that assuming a multinomial unigram model, the probability that the query $q$ is generated by the document model $\theta_{d_j}$ is calculated as follows:

$$P(q|\theta_{d_j}) = \prod_{q_i \in q} P(q_i|\theta_{d_j})$$ \hspace{1cm} (2.17)

and by using maximum likelihood estimation (MLE) for estimating the probability of a single term $q_i$ of the query $q$:

$$\hat{P}_{MLE}(q_i|\theta_{d_j}) = \frac{tf_{ij}}{\text{length}(d_j)}.$$ \hspace{1cm} (2.18)

The drawback in maximum likelihood estimator is the fact that for 1) terms of the query that are very sparse in the documents and 2) terms, part of the information need, not in the document, the likelihood is underestimated. In the literature, smoothing techniques has been proposed to overcome these issues [31]. They are also able to avoid zero probabilities for query terms not occurring in the document. The general approach is based on the idea that not occurring terms (in a document) can occur in the query but their likelihood should be close (and not more) to their probability of being randomly selected from the whole collection. Formally, if $tf_{ij} = 0$, then $\hat{P}(q_i|\theta_{d_j}) \leq \frac{cf_{qi}}{T}$ where $cf_{qi}$ is the raw count of the query term $q_i$ in the whole collection and $T$ the number of tokens of the collection. Two main smoothing procedures were proposed in the literature. The first, named Jelinek-Mercer smoothing (or linear interpolation), combines the probability of the query term within the whole collection with the probability of the same term of being generated from the document model [69]:

$$\hat{P}(q_i|\theta_{d_j}) = \lambda \times \hat{P}_{MLE}(q_i|\theta_{d_j}) + (1 - \lambda) \times \hat{P}_{MLE}(q_i|\theta_C).$$ \hspace{1cm} (2.19)

where $\theta_C$ is the model estimated from the whole collection $C$ and $\lambda$ is a tunable parameter. The smaller is $\lambda$, the more smoothing is given to the probability. The second, named Bayesian Updating process has been proposed in [103] as follows:

$$\hat{P}(q_i|\theta_{d_j}) = \frac{f_{ij} + \alpha \times \hat{P}(q_i|\theta_C)}{\text{length}(d_j) + \alpha}.$$ \hspace{1cm} (2.20)
This equation is tuned by the parameter $\alpha$. The larger is $\alpha$ the more smoothing is given to the probability.

In addition to the above approaches, statistical-based approaches have been shown effective with respect to document ranking. As discussed by [82], one of the most promising approaches in this respect is based on the language model for information retrieval. In such approach, for a given query $q$, a retrieved document $d_j$ can be ranked by measuring the divergence between the document language model $\vartheta_{d_j}$ and the query language model $\vartheta_q$. This means that the score for $d_j$ based on query $q$ is computed as follows [82]:

$$
\text{score}(q, d_j) = -KL(\vartheta_q \parallel \vartheta_{d_j})
= - \sum_{t_i \in V} p(t_i | \vartheta_q) \log \left( \frac{p(t_i | \vartheta_q)}{p(t_i | \vartheta_{d_j})} \right)
$$

(2.21)

This score is based on the Kullback-Leibler divergence between the document language model $\vartheta_d$ and the query language model $\vartheta_q$. One of the advantages of this model is also the simplicity in incorporating terms from relevant feedback documents as we show in the next section.

### 2.1.4 Use of Relevance Feedback

The idea behind relevance feedback is to reinforce the original query intent by making it more similar to the relevant documents [25] and can be generalized by the following procedure:

- A query $q$ is submitted to a retrieval system $S$
- $S$ returns a result set $D$
- Users selects relevant documents browsing the set of result set $D$
- Terms from selected relevant documents are scored (scoring step)
- Initial query $q$ is modified to a query $q'$ after assigning different term weights (re-weighting step) and adding new relevant terms (query expansion step)
- Query $q'$ is re-submitted to the system $S$

This method is generally referred to as explicit feedback since it requires the interaction with the user in the selection of relevant and non-relevant documents. To avoid the interaction with the user, the set of relevant documents can be selected
from the top-k retrieved documents. This implicit feedback method is also called pseudo relevance feedback or blind feedback. Relevance feedback is generally implemented by the Rocchio method [142] and its variants. The original definition has been formulated as follows:

\[ q' = \lambda q + \frac{\beta}{|D_R|} \sum_{d_j \in D_R} d_j - \frac{\gamma}{|D_\bar{R}|} \sum_{d_j \in D_\bar{R}} d_j \]  

(2.22)

where \( D_R \) is the set of known relevant documents to the initial query \( q \), of the whole dataset and \( D_\bar{R} \) the set of known non-relevant documents. Considering documents \( d_j \) and query \( q' \) as vectors, this formula move the original query to the centroid of the not relevant documents to the centroid of the relevant documents. The three parameters \( \alpha, \beta \) and \( \gamma \) can tuned in order to give more importance to the initial query or the set of known relevant and not relevant documents. Both the cases presented (explicit and implicit feedback) can be seen as methods based on local analysis since the information on query relevance are taken from the returned documents. Another alternative is the global analysis which refers to the relevance feedback methods exploring external resources to derive feedback information.

Finally, to include feedback documents in the approach summarized in the Equation 2.21, it is necessary to re-compute the query language model (hereafter referred to as “model” for simplicity). Different approaches has been presented to do this. According to [101], the most effective approaches are the mixture model [188] and the relevance model [84]. Let the set \( F = \{d_1, ..., d_k\} \) be the set of feedback documents i.e.– the top-k retrieved document given a user query and a retrieval model, the mixture model proposed in [188] aims at including feedback documents \( F \) into the KL divergence in Equation 2.21, where the problem concerns the estimation of \( \vartheta_d \) and \( \vartheta_q \). The idea is to estimate the query model \( \vartheta_q \) using any known information related to the user need – i.e., the relevant set and the query. These are combined in a smoothing fashion to estimate the new query model:

\[ \hat{\vartheta}_q' = \alpha \hat{\vartheta}_F + (1 - \alpha) \hat{\vartheta}_q, \]  

(2.23)

where \( \vartheta_F \) is the model estimated from the feedback documents. Further, two approaches are presented to estimate \( \vartheta_F \). In the first approach, the feedback documents are generated by a mixture model composed by the query topic model \( p(t|\vartheta) \) and the collection language model \( p(t|C) \) to also consider the not-relevant part in the feedback documents.

\[ \log p(F|\vartheta_F) = \sum_{t_i \in V} c(t_i, F) \times \log((1 - \lambda)P(t_i|\theta_F) + \lambda P(t_i|C)) \]  

(2.24)

where \( c(t_i, F) \) is the number of terms \( t_i \) in \( F \). In this approach, the Expectation Maximization (EM) algorithm [41] is used to estimate \( \vartheta_F \). In the relevance model
2.1. Information Retrieval

approach, on the other hand, the idea is to minimize the divergence between the query topic model and the model from the feedback documents using the following equation:

\[ \vartheta_F = \arg\min_{\vartheta_F} = \text{Div}(\vartheta_F; F, C), \quad (2.25) \]

where

\[ \text{Div}(\vartheta_F; F, C) = \frac{1}{|F|} \sum_{j=1}^{N} \text{Div}(\vartheta_F\|\vartheta_{d_j}) - \lambda\text{Div}(\vartheta_F\|p(t|C)). \quad (2.26) \]

Here \( \text{Div}(\vartheta_F\|p(t|C)) \) is a regularized element introduced to consider documents sharing common terms. The idea is to give high probability to terms that are not common according the collection language model but common according the model from feedback documents. To summarize, in both approaches, the retrieval processes consist of three steps: 1) estimation of \( \hat{\vartheta}_F \) using Equation 2.24 or Equation 2.25, 2) calculation of \( \hat{\vartheta}_{q'} \) using Equation 2.23, and 3) computing the divergence between \( \hat{\vartheta}_{q'} \) and \( \hat{\vartheta}_F \) using Equation 2.21.

Recent works [102, 100, 110] have presented extensions of these relevance feedback approaches. The idea behind [102] is that terms in the document close to a query term are more likely consistent with the topic of the query. The proposed model assigns the weight according to the proximity of the document term to the query terms. From a similar assumption in [110], the authors extend the Rocchio’s model incorporating proximity measures. Finally the work [100] explores the problem of finding the best parameter values in the mixture model, for tuning the balance between the original query and the feedback information.

2.1.5 Retrieval Evaluation

As mentioned in Section 2.1.1, one of the main requirements for an IR system is the effectiveness with respect to the quality of the documents retrieved, given user queries. Several evaluation measures have been proposed in the literature for evaluating the effectiveness of a retrieval method. Every evaluation measure assumes that each document can be classified in a binary way as relevant or not relevant according to a user need – i.e. the user query. This section briefly presents measures for not ranked documents – i.e., precision, recall, F-measure, and ranked documents – i.e., Mean Average Precision (MAP), R-Precision, precision at k (P@k).

First, precision represents the fraction of relevant documents in a set of retrieved ones as follows:

\[ \text{precision} = \frac{|\text{Rel} \cap R|}{|R|}, \quad (2.27) \]
where \( R \) is the set of retrieved documents and \(|Rel|\) is the set of relevant documents to the user query.

Another measure evaluating how exhaustively the results from the retrieval method satisfy the user query, is represented by \textit{recall}. It is the fraction of relevant documents in the retrieved documents defined as follows:

\[
recall = \frac{|Rel \cap R|}{|Rel|} \tag{2.28}
\]

The two measures express different properties of the method evaluated. High recall means that the retrieval method returns the majority of the relevant documents while high precision states a higher number of relevant documents in comparison to the irrelevant ones. The \textbf{F-measure}, combines precision and recall in a single measure, in a weighted harmonic mean as follows:

\[
F = \frac{2 \times \text{(precision)} \times \text{(recall)}}{\text{precision} + \text{recall}}. \tag{2.29}
\]

The more different precision and recall are, the smaller the \( F \) value is. Therefore we get a high value of \( F \) when both precision and recall values are high.

In the case of a ranked set of documents, it has been proposed the \textbf{precision at k} (\( \text{P@k} \)), represents the value of the precision at \( k \) ranking position. \( P@k \) is useful for evaluating Web search engines where it is usual for the user to browse the top set of documents. One of the advantages in computing this measure is that there is no need for the knowledge of the whole set of relevant documents – i.e., only need an assessment of the top-\( k \) documents retrieved.

One of the most effective and used measures for measuring effectiveness over a set of diverse queries is the \textbf{mean average precision} (MAP). It is a single value summary of the ranking quality of the results. Assuming \( Q \) a set of queries, \( MAP \) is defined as:

\[
MAP(Q) = \frac{1}{|Q|} \sum_{q_j \in Q} AP(q_j). \tag{2.30}
\]

Here \( AP(q_j) \) is the \textbf{average precision} for the query \( q_j \) as follow:

\[
AP(q_j) = \frac{1}{|Rel_{q_j}|} \sum_{k=1}^{|R|} \left( rel(k) \times P@k \right), \tag{2.31}
\]

where \(|Rel_{q_j}|\) is the set of documents relevant for the query \( q_j \), \( rel(k) \) is a function equal to 1 if the document \( k \) in the set of retrieved document \( R \) is relevant, 0 otherwise.
Another single value summary for ranking quality, related to $MAP$ is the \textbf{R-Precision}, calculating the precision at $|R|$, where $|R|$ is the number of relevant document for a given user query. This measure is normalized for the number of relevant documents of the query and it is calculated for a single query of the experiment.

### 2.1.6 Open Source Search Engine

Open source search engines are generally set of software library providing basic blocks and general functions for the development all the part of a search engine. They differ on the programming language used for the implementation, the ranking function they implement and the indexing process they use. Different works on comparing their performance over different tasks have been presented in the past [171, 7, 122]. In this section, we briefly describe the characteristics used for developing the ranking models proposed in the thesis and for the experiments with the baseline models.

**Terrier IR Platform**

Terrier (TERabyte RetrIEveR) [114, 115], is an open source project started by the Information Retrieval Research Group at the Department of Computing Science, at the University of Glasgow. It is a flexible, transparent, scalable, efficient, and effective open source search engine platform implemented in java. It also supports quick development and evaluation retrieval applications on large-scale collections of documents. Terrier implements Divergence From Randomness weighting models and most of the state-of-the-art probabilistic and statistical retrieval models such as TF-IDF, BM25 and Language Models. Its modularity property makes Terrier easy for developing new ranking models or for extending the existing ones.

**Apache Lucene and Solr**

Due to its high performance, Lucene is one of the most successful and used full-featured search engine library. It is supported by the Apache Software Foundation and is released under the Apache Software License. The indexing process is scalable and index can be incrementally expanded. It provides also powerful and accurate searching functionality based on the concept of field – i.e., every document is considered as a set of different fields such as title, body, and so on. Fielded searching and multiple-index searching is also admitted. Due to its success, several other search engines have been built on top of Lucene such as Solr. Solr is a popular, highly scalable, fast open source search platform and its major features include powerful full-text search, faceted search, hit highlighting, dynamic clustering, rich document

\[8\text{http://ir.dcs.gla.ac.uk/}

\[9\text{http://lucene.apache.org/solr/}

(e.g., Word, PDF) handling and geospatial search. It also provides distributed search
and index replication, and it powers the search and navigation features of many of
the world’s largest Internet websites.

**Lemur Project**
The Lemur project\(^{10}\) was a project started as a cooperation between *University
of Massachusetts Amherst* and the *Carnegie Mellon University*. The main goal
with the project was to develop all set of tools supporting research on Information
Retrieval. The most famous component is the **Indri** search engine developed in
*C++* language, and provide library for text retrieval over large collection for single
machine or distributed search. As with Lucene, Indri can handle multiple fields per
document that is an important properties for Web Searching. It implements most
of the query language models, supports automatic query expansion and is highly
scalable.

### 2.1.7 Final Remarks

This section has presented several concepts from information retrieval (IR). As dis-

cussed later many of these concepts have been used to develop the idea and the
evaluation of this work. First, the suffix tree structure is presented in the Chapter 4
as the baseline data structure for the mining process. Second the ranking models are
needed as the baseline models in Chapters 6 and 8 as comparison with the model
proposed in the the work. Third the implementation of the retrieval models has
been done based on the Terrier 3.5 framework while the indexing process has been
performed by using both **Terrier 3.5** and **Solr 1.4** frameworks. Finally the retrieval
evaluation metrics such as MAP and RP have been used to assess the effectiveness
of the methods we propose.

### 2.2 Exploratory Analysis of Spatial Point Pattern

The **point process statistic** aims to explore the structure of a set of points dis-

dributed in a two or three dimensional space. In the past, exploratory techniques
related to the point process statistic have been used in different fields such as epide-

miology and ecology. This section briefly presents some basic concept of the point
process statistic. In particular, the preliminary theory on **Spatial Point Process, Spa-
tiotemporal Point Process, Complete Spatial Randomness Test (CSR)**, and **Poisson
distribution** which is the benchmark point process used to perform the CSR test.

\(^{10}\)http://www.lemurproject.org/
2.2.1 Point Process and Point Pattern

The **spatial point processes** are stochastic models of irregular point patterns. A **point pattern** is a non-ordered collection of points in some area or set and is the realization of a point process. A **point process** may be specified by $N$, the random variable representing the **counting measure**. For a subset $B \subset \mathbb{R}^d$, $N(B)$ is the random number of points in the sub-area $B$. Generally, the point processes are assumed to have the property of **simplicity** that means that all points are different, i.e., there does not exist any $(i,j) : x_i = x_j$.

Point processes are generally described by the **number distribution** given by:

$$P(N(B) = n) \quad n = 0, 1, ...$$

(2.32)

The above equation describes the probability that there are exactly $n$ points in the set $B$. It can be described as a univariate distribution. There can be infinitely many sets $B$ and the probabilities can differ among these sets. Hence, there are infinitely many number distributions that describe a point process.

A point process $N$ can have the properties of **stationarity** and **isotropy** according whether the distribution of the process is invariant to the translation or invariant to the rotation. A process with both the properties is called **motion-invariant**. Another property is the **ergodicity** meaning that the average over a large sample yields the same results as the average resulting from many small samples [68].

**Central limit theorem** is also proved for the number of points in a large window which can be regarded as number of points in many sub-windows.

**Mean for Point Processes**

The concept of mean in the context of point processes, where the value $N(B)$ is a random variable and $B$ is a bound subset, is indicated as **intensity measure** $\lambda(x)$. It is generally indicated as a **first-order measure**. This value is the expectation – i.e., mean number of points, in a unit area/volume. The intensity is formally described as:

$$\lambda(x) = \lim_{dx \rightarrow \infty} \frac{E(N(dx))}{a(dx)}$$

(2.33)

**Variance for Point Processes**

The variance $\text{var}(N(B))$ in the context of point processes of the random variable $N(B)$ may be considered as:

$$\text{var}(N(B)) = E((N(B) - E(N(B)))^2$$

$$= E(N(B) - \Lambda(B))^2$$

$$= E(N(B))^2 - \Lambda(B)^2$$

(2.34)
As this shows, the variance depends on the set $B$, and a point process is associated with infinite variances according to the subset $B$. Using this, it is possible to define high order moments as:

$$E(N(B))^k \quad k = 2, 3, ... \quad (2.35)$$

Summary of Point Processes Characteristics
There are several ways to summarize the characteristics of point processes. These are useful in exploratory and data analysis for describing the peculiarity of the underlying distribution. In the classic univariate statistic where $x$, $s^2$ and the empirical distribution function $F_n(x)$ summarizes the characteristics of a random variable. Similarly, the characteristics of the spatial point processes are summarized by numerical values or by functions.

Numerical values, summarizing the characteristics of the spatial point processes, are also named first order characteristics and correspond to the mean, in the standard statistics. The most important numerical characteristic for stationary point process is the intensity $\lambda$ (for non-homogeneous spatial point processes this number is a function of the location $\lambda(x)$) which indicates the mean number of points per unit area or volume.

Functions summarizing the characteristics of the spatial point processes, are in the class of the second order summary statistics. They are represented as a function of a distance from a certain point of the distribution, describing the spatial properties of a point to the rest of the points of the distribution. The nearest-neighbor distance $D(r)$ and Ripley’s $K$-function $K(r)$ are the most important point-dependent functional summary characteristics. These functions will be described in depth later in the chapter.

2.2.2 Distributions for CSR Modeling

A given spatial point pattern may exhibit or not exhibit various kinds of interactions between its points. For instance, the point may occur in clusters, may exhibit regularity or may be completely spatially random (Figure 2.2).

The property of the point pattern of exhibiting a completely random pattern is called complete spatial randomness (CSR). A theoretical model with this property forms an important basis for comparison, as a null model and as reference for the construction of summary characteristic. Assuming stationarity and the lack of interaction, completely random processes can be characterized as Poisson Point Processes. As a result, it is possible to perform a CSR test that is able to verify whether a given pattern exhibits spatial clustering or regularity, or whether it may
be considered CSR. The following subsection briefly explains the binomial point process and Poisson point process that are able to model the complete spatial randomness. The next section describes how to perform the CSR, and complete spatiotemporal randomness (CSTR) test, for spatiotemporal point pattern test.

**Binomial Point Process**

The above-mentioned Poisson process can be seen as a generalization of the binomial process. Thus before discussing CSR, it is natural to start by giving an overview of the binomial point process. The binomial model represents a null model for finite point process. It consists of \( n \) points randomly (uniformly and independently distributed in \( W \)) scattered in a bounded set \( W \subset \mathbb{R}^d \). The area in the case of a two-dimensional space (or volume, in case of three-dimensional space), is denoted by \( a(W) \). A single random point \( x \) is formally \textbf{uniformly distributed} in \( W \) if

\[
P(x \in A) = \frac{v(A)}{v(W)}
\]  

for all subsets \( A \) of \( W \). This means that the probability that \( x \) takes its position in the subset \( A \) of \( W \) is equal to the ratio of the area of the sets \( A \) and \( W \). If the points \( x_1, \ldots, x_n \) form a binomial point process in \( W \) then the random pattern formed by points is \( N^n_W \) that can be even considered a random set. The random number of points in \( A \), for each subset \( A \) of \( W \), is denoted by \( N^n_W(A) \) and follows a binomial distribution. The distribution probability of this binomial point process is given by:

\[
P(N^n_W(A) = k) = \binom{n}{k} p^k (1 - p)^{n-k} \quad k = 0, \ldots, n
\]

This model is close to a CSR model but it is not sufficient because \( W \) is fixed (and then its number of points \( n \)). In addition \( N^n_W(A_i) \) varies with the changing subset \( A_i \) (with \( i = 1 \ldots n \)), which are not independent even though they are disjoint.
In fact, for example, $N^n_W(A) = m$ directly implies the number of another subset $(N^n_W(W - A) = n - m)$. It is necessary to find a suitable probability distribution of the random number of points, such that the number of points in one subset cannot be predicted from the number of points in another subset.

**Poisson Point Process**

To derive a suitable model for the CSR hypothesis it is necessary to remove the conditioning effect described before for the binomial process. It can be done by applying a Poisson approximation to the binomial distribution. According to the Poisson limit theorem, with $n$ tending to infinite and assuming the same density of points ($\lambda = n/a(W)$) in each subset of the area, the binomial probability in Equation 2.37 converge to a Poisson probability as follows

$$P(N(A) = n) = \frac{\lambda^n a(A)^n}{n!} e^{-\lambda a(A)}$$

(2.38)

if the region $W$ is enlarged toward infinity (to cover the whole $\mathbb{R}^d$) while $n$ is allowed toward infinity. Under this assumption, expanding the limiting density:

$$\lambda = \lim_{m \to \infty} \frac{\lambda(n_{W_m})}{n_{W_m}} = \lim_{m \to \infty} \frac{n_{W_m}}{v(W_m)}$$

(2.39)

is a finite and positive value. Then $N_W(A)$ become asymptotically Poisson distributed with mean $\lambda a(A)$. Finally if there is a set of random variables $N(A)$ describing cell-counts of a finite area over whole plane designated as a spatial point process on this plane, then any process governed by the Poisson probabilities in Equation 2.38 is designated as spatial Poisson process on this plane.

Hence assuming that the number of points $N$ is random and follow a Poisson distribution, then we set the CSR model. According to the intensity function, it is possible to divide point processes into homogeneous and inhomogeneous as follows.

**Homogeneous Point Process** A homogeneous Poisson point process $N$ defined on $W$ has the properties of stationarity and isotropy. For any bounded region $A$ it satisfies the following two fundamental properties:

- $N(A)$, for each bounded set $A$ is Poisson distributed with mean $\lambda v(A)$ for some constant intensity $\lambda$
- $N(A_i)$ in $k$ disjoint sets form $k$ independent random variables

The intensity $\lambda$ is constant and describes the mean numbers of points to be found in a unit area/volume for all bounded sets $A$.

The Poisson distribution of a homogeneous Poisson Point Process $N(A)$ with intensity $\lambda$ and parameter (mean) $\lambda v(A)$ is defined by the Equation 2.38
Inhomogeneous Point Process  The homogeneous Poisson point process is a generalization of the Inhomogeneous Point Process. In the homogeneous case the constant intensity \( \lambda \) is replaced by an intensity function \( \lambda(x) \) whose values vary with location \( x \). The second property of the inhomogeneous Poisson point process remain the same while the first is generalized:

- \( N(A) \), for each bounded set \( A \) is Poisson distributed with mean \( \int_B \lambda(x)a(A) \)
- \( N(A_i) \) in \( k \) disjoint sets form \( k \) independent random variables

2.2.3 CSR Tests

In many natural phenomena it is possible to observe spatial point patterns. To analyze the existence of spatial dependence between the points in the point pattern, it is useful to perform a CSR test. The null hypothesis of CSR says that conditional on the \( N(A) \) number of events in \( A \subset R_d \), the events are independent and uniformly distributed on the observed region \( A \). In this case, the CSR test is not rejected and the point process may be modeled as a Poisson Point Process. In other words CSR test is a central hypothesis separating events that do not interact each other either “repulsively” (regularity of events) or “attractively” (clustering of events).

To test the CSR of a spatial point pattern \( N(A) \), \( A \in R \), the standard is to compare it with a complete spatial random process like the Poisson Point Process that will be explained later. The hypothesis of CSR for a spatial point pattern is the following [99]:

The number of events \( m \) in any planar region \( A \) follows a homogeneous Poisson distribution with mean \( \lambda A(A) \) where the expected constant intensity is \( \lambda \)

In the previous section we already seen the form of the distribution in the Equation 2.38 with mean \( \lambda A(A) \). The \( m \) events in \( A \) are an independent random sample on \( A \). This means that any event has the same probability to occur at any position in \( W \).

Existing methods for testing the CSR hypothesis may be divided in the quadrat method, nearest-neighbor method and method of K-functions as follows

Quadrat Method  Let \( X = m_1, ..., m_n \) be the counts of events, when the sampling windows \( W \) are divided in \( n \) quadrants – i.e., square regions of equal area. The randomness can be tested by using an index of dispersion test. CSR hypothesis assert that the cell count distribution for each quadrant must follow a Poisson distribution as in Equation 2.38 of mean \( \lambda A(A) \) and
counts in the disjoints quadrant are independent. A test statistic of the hypothesis is to use the Pearson’s $\chi^2$ goodness of fit test [10]:

$$\chi^2 = \frac{(n - 1)s^2}{\bar{m}}$$

(2.40)

where $\bar{m}$ is the mean of the expected counts and $s^2$ is the observed variance. The index of dispersion is $I = s^2 / \bar{m}$ and is the estimation of this ratio. For a Poisson distribution the index $I$ is 1 since the mean is equal to the variance. Then $I > 1$ suggest possible dispersion, $I < 1$ possible clustering.

The drawback of this procedure is that it is strictly related to the size of the selected square. Different types of index of dispersion try to overcome this problem ([36] and [57]).

Nearest Neighbor Method This method is based on the observations, from each event $m_i$ of the distance with the nearest neighbor in $W$. Let us consider the Poisson distribution function, given in Equation 2.38 as the model of complete randomness with intensity $\lambda$. Let $d$ be the distance between the events and $\pi d^2$ the area of the circle centered on the target event. Let $D$ be a random variable denoting the nearest neighbor distance from the target event $e_i$ to the rest of the pattern. If $C_d(e_i)$ is the circular region centered in $e_i$ with radius $d$ then:

$$P(D > d) = P(N(C_d(e_i) = 0)) = e^{-\lambda \pi C_d(e_i)} = e^{-\lambda \pi d^2}$$

(2.41)

It follows that the probability distribution function $F(r)$ of the nearest neighbor event distances under CSR assumption is:

$$F(r) = P(D \leq d) = 1 - P(D > d) = 1 - e^{-\lambda \pi d^2}$$

(2.42)

Different statistical methods exist to test the CSR hypothesis based on nearest neighbor. The most common are the ones proposed by Clark-Evans [34], Hopkins [66] and Blyth-Ripley [23].

The drawback of this method is that since it only uses the closest event, it only catches the smallest scale of the pattern, and thus the only smallest-scale clustering tendency.
2.2. Exploratory Analysis of Spatial Point Pattern

**K-function method**

When it is convenient to exhibit different structures on different scales, it is necessary to apply point pattern analysis methods that take into account such a scale effect. The K-function method is one such method.

The K-function method or **reduced second moment order** was originally proposed in [134] for homogeneous and isotropic spatial point processes over the whole space and consider randomly sampled cells of different sizes. The definition is:

\[
\lambda K(h) = E(\#(\text{evts within distance } h \text{ of an arbitrary evt})),
\]

(2.43)

where \( \lambda \) is assumed constant throughout \( R \).

In K-function analysis, the scales become a fundamental variable. Equation 2.43 contains information about cluster and dispersion and it is the function of the distance \( h \). So, \( \lambda K(h) \) represents the expected number of extra events within distance \( h \) of an arbitrary event in a circular area of radius \( h \). In case of a homogeneous and isotropic Poisson process, this value is \( \lambda K(h) = \lambda \pi h^2 \).

A suitable estimation of \( K(h) \) is:

\[
\hat{K}(h) = \frac{1}{\lambda n} \sum_{i=1}^{n} \sum_{i \neq j} I_h(d_{ij}),
\]

(2.44)

where \( d_{ij} \) is the distance between \( i \)th and \( j \)th observed events, \( I_h(d_{ij}) \) is an indicator which has the value 1 if \( d_{ij} \leq h \), 0 otherwise and \( \hat{\lambda} = n/a(R) \). Moreover \( w_{ij} \) is the edge corrector like the one introduced in [134]. It is the inverse of the proportional part of the circumference of the circle with center on \( x_i \) and with radius \( d_{ij} \). This estimator uses all the points of a point pattern.

Considering the \( K(h) \) function of the Poisson point process as benchmark we can use the estimation to test the CSR hypothesis. In case of independently distributed points and Poisson distribution, the number of events within distance \( h \) is \( \lambda \pi h^2 \), and thus \( K(h) = \pi h^2 \). The CSR test may be performed by comparing the empirical \( \hat{K}(h) \) with \( K(h) \). In particular, \( \hat{K}(h) > \pi h^2 \) indicates some degree of clustering on scale \( h \) and \( \hat{K}(h) < \pi h^2 \) indicates some degree of dispersion on scale \( h \).

To standardize the K-function it is possible to define the L-function:

\[
L(h) = \sqrt{\frac{K(h)}{\pi}}
\]

(2.45)

Under CSR the L-function is zero. Further, \( L(h) > 0 \) indicates clustering on scale \( h \), whereas \( L(h) < 0 \) indicates dispersion on scale \( h \).
In a real world the homogeneous Poisson process is not suitable to model random point patterns. Most of the time the point patterns exhibit different intensity over the space – i.e., non-stationarity and inhomogeneity.

An extension of second order analysis of Ripley’s $K$-function method, for the inhomogeneous Poisson processes is proposed in [8]. For such an inhomogeneous Poisson process with $\lambda(x)$ as the known first-order intensity, the estimator of $K(h)$ is:

$$
\hat{K}(h, \lambda) = \frac{1}{a(A)} \sum_{i=1}^{n} \sum_{j \neq i} w_{ij} I(d_{ij} \leq h) \frac{\lambda(x_i)\lambda(x_j)}{\lambda(x_i)\lambda(x_j)}
$$

(2.46)

where $I(d_{ij} \leq h)$ is the same indicator of Equation 2.43, $w_{ij}$ is the edge corrector. Normally, the intensity function $\lambda(x)$ is unknown and needs to be estimated.

### 2.2.4 Extension to Spatiotemporal Data

When it is necessary to model the time dependent and dynamic point pattern we should consider the spatiotemporal point pattern in a finite spatiotemporal region. This consists of a set of time-ordered events $(x_i, t_i) : i = 1, ..., n$ and the underlying stochastic model is called the spatiotemporal point process.

Spatiotemporal point processes may be temporal continuous or discrete and spatial continuous and discrete. Based on the nature of the two dimensions we can separate these into different kinds of spatiotemporal point processes [149] such as the earthquake process in case of events that once occur at a particular location in a particular time; the explosion process for events having a stochastic spatial realization in a specific temporal point and the birth-death process (or space-time survival point process [36]) for events that are born in a time $t_a$ and live in a specific location for a period $t_l$.

In [51], the authors proposed an extension of the second order analysis for the spatiotemporal inhomogeneous point process, directly extending the work in [8]. This point process model is suitable to fit the inhomogeneity spatial and temporal nature of some phenomena.

The first-order spatiotemporal intensity function $\lambda(x, t)$ now defines the expected number of events in the volume centered on $x$ and $t$ per unit time and unit space while the $\lambda_2((x_i, t_i)((x_j, t_j)))$ is the second-order intensity function. Inspired by [51], we apply the extension to spatiotemporal inhomogeneous processes assuming that spatiotemporal point pattern is bounded in time and space. This means that each $e_i \in A$, where $e_i = (x_i, t_i)$ and $A = X \times S$ where $T$ is a time interval and $S$, is a bounded planar area.
2.2. Exploratory Analysis of Spatial Point Pattern

The benchmark for testing the complete spatiotemporal randomness (CSTR) is the spatiotemporal inhomogeneous Poisson point process. Ripley’s $K$-function for this distribution is $K(u,v) = \pi u^2 v$ [51], where $u$ is the Euclidean distance in the planar region $\|x_i - x_j\|$ and $|t_i - t_j|$ the time interval between two points $x_i$ and $x_j$. An estimator for the spatiotemporal inhomogeneous $K$-function is given by [51] as follows:

$$\hat{K}(u,v) = \frac{1}{a(S,T)} \frac{n}{n_v} \sum_{i=1}^{n_v} \sum_{j=1; j>i}^{n_v} \frac{1}{w_{ij}} \frac{1}{\lambda(x_i)\lambda(x_j)}$$ \hspace{1cm} (2.47)

This equation is similar to Equation 2.43, even though it provides an edge corrector in two different spaces. For the locational space it uses the same corrector proposed in the spatial $K$-function, while to deal with the temporal edge effects (for point temporally on the border), in Equation 2.43 it is specified by the value $n_v$. For each $v$, this value specifies the number of events for which $t_i \leq (T_1 - v)$

The CSTR test is performed from the estimated $\hat{K}$-function from the empirical data. These data are compared with the benchmark $K$-function of the inhomogeneous spatiotemporal Poisson process – i.e., values of $\hat{K}(u,v) > \pi u^2 v$ indicates spatiotemporal aggregation at spatial and temporal distance $u$ and $v$, while $\hat{K}(u,v) < \pi u^2 v$ indicates regularity.
Chapter 3

Related Work

Although the topic of event detection in image management is not very old, the event detection as a topic has existed since the TDT (Topic Detection and Tracking) initiative was proposed [2, 3]. This chapter discusses how event detection has been used in other works related to the approaches presented in this thesis.

Extracting pictures related to real-life events is an active research field [55, 168], and in the past few decades, the detection of events from textual document streams and databases has been extensively treated in the literature [3, 19]. Still, despite being active, event-related image retrieval and matching for photo-sharing applications is not a fully mature field. To put the present research into perspective, I briefly discuss some approaches that are related to the approach addressed here.

Most related approaches in event retrieval and matching have been aimed at extracting events from different kinds of datasets. To my best knowledge, only a few works have addressed the problems of retrieving events related to media-sharing. Most of these approaches were presented in the Social Event Detection (SED) task at MediaEval 2011 [117], where the main objective was to develop event retrieval systems for Flickr pictures. The work by Trad et al. [168] is very interesting. The authors proposed a method to match a given (query) picture representing an event to pictures representing the same events in a picture collection. The query image is provided with both temporal and spatial information, and the matching algorithm is based first on visual similarity, followed by a re-ranking step based on geo-temporal coherence. In order to handle the scalability, they use Map Reduce in the content analysis and indexing process, and conducted their experiments on a set of around 1 million pictures, from the LastFM-Flickr dataset [170]. The work presented in this thesis differs from the [170] work in that rather than applying visual features, only use textual data, temporal and geographical data are used here. This enables work on a much larger dataset.
3.1 Event Detection within Photo-Sharing Applications

Several works address the event detection problem from community photo collection such as the ones proposed in [30, 129, 119].

The work presented in [30] uses both temporal and spatial information (GPS position as latitude and longitude) to analyze the tags. However, besides applying wavelet transform on data to reduce the noise, this approach differs from the one proposed in the present thesis in several aspects. The approach presented in [30] consists of two main steps. The first is event tag detection to extract tags related to events based on their spatiotemporal distribution. The second is the generation of events, to group tags related to the same event together, by using semantic similarity and considering spatial distance between two tags. This distance is computed based on the KL-divergence between the two densities representing the two tags. As part of this step, a pre-step process is performed to differentiate tags related to periodic and aperiodic events. Then, the retrieved events are associated to a set of photos (still differentiating the periodic and aperiodic events) after determining the time and location of each photo. This kind of approach is called Feature-Pivot approach [50]. Here the main focus is on the determination of bursty events – i.e., events that are “hot” in a certain period of time – in a chronologically ordered text stream. Bursty events consist of a set of bursty features, and are mainly used for assisting with text classification. Thus, the main goal of the presented approach is to extract bursty features to detect any related bursty events.

In both the approaches presented in [129] and [119] the authors proposed methods for detecting groups of events and landmark pictures from community photo collections, by applying a clustering step and followed by a classification step. The main difference is the way the clustering step is carried out. While the former used an agglomerative clustering algorithm, the latter is based on community detection clustering algorithm. Nevertheless, at a first glance, these approaches seem to be related to the present approach in that we can apply the event detection part in the event retrieval process. However, the focus is different in that we are most interested in directly retrieving event-related pictures without having to cluster and classify them first. Also, both of the approaches are based on visual features. In addition to these two approaches, the work by Becker et al. [14] is another approach on extracting events from community photo collections. Here, the authors have mainly focused on event clustering.

Other works addressing event detection from Flickr images were presented in [39] and [133]. In [39] the focus was on the extraction of location and event semantic
for tags assigned to Flickr photos. In [133], the authors also presented an approach for detecting events tags in a user collection photos. However, here only temporal features were considered. The idea is to capture the picture-taking behavior of a user, using time-series modeling – e.g., number of images per day. By analyzing this behavior, their algorithm can extract significant events from the image collections. To do this, they adopted the ARIMA (Auto-Regressive Integrated Moving Average) model to predict the user's picture-taking behavior over time. Then by assuming that significant events are normally those that do not fit with any typical behavior, they used the differences between the predicted behavior and the actual user behavior to identify significant events.

3.2 Event Detection from Textual Documents

Most previous papers have focused on detecting events from document streams instead of image collections. In particular the two main detection tasks – retrospective detection and online detection – were defined in [182]. The former aims at discovering previously unidentified events, while the latter aims at identifying new events from a stream of news. Existing papers on retrospective detection are [3], [19], [190], and [67], whereas the approach on online detection was proposed in [182]. More specifically, in [182] the authors developed a group-average clustering (GAC) algorithm for retrospective detection and Incremental Clustering (INCR). This is a single pass algorithm for both retrospective and online detection. For a given set of documents, the GAC algorithm produces a forest of cluster trees. The idea is to exploit the temporal proximity of the different stories by giving more priority in grouping temporally consecutive stories. The event detection approach proposed in this thesis can be compared to this work in that the focus is also on the detection of events in a set of documents gathered in temporal windows. However, the documents here are images surrounded by metadata representing contextual information such as textual annotation, and geographical and temporal information, and not document streams, per se.

Some other approaches have been proposed to use other information than time in the event detection process. For example, in [160] and [71], location information was used to improve the effectiveness of TDT. In the first work a retrospective event detection on unstructured history documents was presented. Statistical measures were used to analyze the frequency of co-occurrences between date and place names over sentences and paragraphs. Based on this information, events can be extracted and ranked. In the second work, the authors presented an analysis of the contribution of place and time information in the event tracking domain. Here, place names were extracted automatically from newspaper articles. The main idea of this work is to
apply named entity recognition to gather place information from documents. Such an approach has not yet been extensively treated in TDT. However, while TDT could benefit from the place information, the authors recognized that place names are difficult to process because of the incurred ambiguities – e.g, Washington as a person vs. a place name.

Further, in addition to Feature-Pivot approach in Section 3.1, there are other works that have focused on the document content, which can be categorized as Document-Pivot approaches [182, 3, 19, 169]. As opposed to Feature-Pivot approaches, the main idea is to group documents based on events extracted from the document content. This means that a groups of similar documents form an event.

3.3 Event Detection and Extraction from Microblogs.

Due to the advance of the Internet-based social community, much effort has been put on developing approaches to identify and extract events from different social community resources such as Microblogs [95, 29, 146, 178, 15], and image-sharing applications [143, 129, 119, 14]. Focusing on Microblogs, such as Twitter, the user contributes to the social media by posting text messages that are generally short and tagged with a temporal tag. The most important differences of these types of text compared with textual documents are the average length of the textual messages and the noises which such messages contain. Work on event detection in this domain have tried to tackle the above two characteristics in different ways. For example, Long et al. [95] propose a language-independent approach for detecting, summarizing and tracking events from Twitter posts. Further, Chakrabarti and Punera [29] suggested a real-time approach to summarize the Twitter posts as events, using a modified variant of the Hidden Markov Model to model the hidden state representation of an event. Other examples of real-time approaches were presented in [178, 15, 146]. In [178], the goal was to detect events from tweet posts by leveraging on their geographical and temporal tags. In [15], the authors presented a method composed by a clustering step, followed by a classification step to group tweets and separate event clusters from non-event clusters, respectively. Finally, Sakaki et al. [146] investigated the possibility to detect events such as earthquake using the real-time stream of tweet posts as sensors. For this, the authors proposed a specific spatiotemporal model based on the Kalmann filter to detect such a kind of event.
3.4 Event Detection using Visual Features

There are few approaches that use visual features to track events. An example of these is the work by Le and Fei-Fei [87]. In this paper, the authors proposed a system to recognize events in static images using visual contents. Classification was performed on a picture of sport events interpreting semantic elements in the image, and by integrating scenes and object categorization. Another example is the work proposed in [97]. In this paper, the authors proposed a novel approach for automatic generation of albums – also called automatic albums – from a collection of personal photos. In particular, the system presented includes an event clustering algorithm that works on two levels, date/time and visual content. The goal of automatic albums consists of helping users to organize their pictures in a story. A story is here an organized set of photos with the appropriate context information, that participate in the interpretation of the photo. Personal photos are then classified using the combination of date and time, and the correlation between the visual contents of the pictures.

In addition to the above approaches, the work proposed in [52] is worth discussing. Here, the authors proposed using visual features to recognize interesting objects from a picture, and apply these features to build a visual vocabulary. Further, they built a reference database by crawling Web-based community photo collections. Based on the combination of the resulting collections, they managed to automatically label mined objects, such as landmarks, arts and scenes. This means that the main goal of their approach was to detect objects from photos, and give the objects correct labels. From this perspective, the present work can be related to this work in that the method in [52] could also be used to label events. However, the main differences is that rather than annotating photos at the object level, the proposed event detection method here focuses on grouping and detecting photos representing the same event.

3.5 Other Related Approaches

In [185] event categorization was treated as a multiclass classification problem. The focus is first on discovering and mining compositional features, and then applying that for classification. GPS traces were used as series of Geotags and fused with the visual content to classify images into predefined events. This approach uses the AdaBoost classifier for classification purposes [185].

In the field of multimedia tagging in general, there is a lot of work that can implicitly be related to the work presented in this thesis. Most notable are those which have been carried out in the area of video tagging, presented in the MultimediaEval
Tagging Task [83]. Examples of such work were proposed in [177] and [63], where the main idea was to exploit the metadata associated with videos in video retrieval processes. Both of these approaches use information retrieval methods to generate tags from video metadata. However, the focus of these approaches is different from the present one. Even though the present approach could benefit from adopting the ideas in these methods – e.g., in terms of identifying relevant terms for candidate events, they will be included in future work.

Yet another related topic, part of multimedia tagging is Geo-tagging. This means automatically assigning a geographical tag to a picture by using different features [61, 153]. Specific examples of such works were introduced in the MultimediaEval Placing Task [83] and in [191]. Even though the focus of these works is different from the present work, the event detection approach presented in our thesis could benefit from adopting the method introduced in these works. For example, automatic tagging of a specific location of a specific picture would be interesting to explore. Such an approach could improve the mining process by also including non-Geotagged pictures, since their geographical position could be estimated in advance.

Finally in the past few years there has been an effort to use ontology to support the exploration of events in media collection. In [46, 170] the authors presented the LODE ontology used to represent event descriptions and to ensure interoperability among different media ontologies. In this respect, the method presented in these papers could be used to provide events with more informative labels and link them to external resources such as news and other related websites. Nevertheless, more work is still needed to disambiguate the labels and the possible links.

\footnote{This paper is the winner of the "Where I am?" ICCV 2005 contest. See also \url{http://research.microsoft.com/en-us/um/people/szeliski/visioncontest05/default.htm}.}
Part II

Mining and Detect Event in Social Media
Chapter 4

Event Mining of Social Media
Pictures By Suffix Tree Structure

As specified in the previous chapters, the event detection problem, which is closely related to clustering, has gained a lot of attentions within event detection for textual documents. However, although image clustering is a problem that has been treated extensively in both Content-Based Image Retrieval (CBIR) and Text-Based Image Retrieval (TBIR) systems, event detection within image management is a relatively new area. Having this in mind, in this chapter we propose a novel approach for event extraction and clustering of images, taking into account textual annotations, time and geographical positions. Our goal is to develop a clustering method based on the fact that an image may belong to an event cluster. Here, we stress the necessity of having an event clustering and cluster extraction algorithm that able to work on large scale dataset and allow online applications. To achieve this, we extend a well-known clustering algorithm called Suffix Tree Clustering (STC), originally developed to cluster text documents using document snippets. The idea is that we consider an image along with its annotation as a document. Further, we extend it to also include time and geographical position so that we can capture the contextual information from each image during the clustering process. This has appeared to be particularly useful on images gathered from online photo-sharing applications such as Flickr. Hence, our STC-based approach is aimed at dealing with the challenges induced by capturing contextual information from Flickr images and extracting related events. We evaluate our algorithm using different annotated datasets mainly gathered from Flickr. As part of this evaluation we investigate the effects of using different parameters, such as time and space granularities, and compare these effects. In addition, we evaluate the performance of our algorithm with respect to mining events from image collections. Our experimental results clearly demonstrate the effectiveness of our STC-based algorithm in extracting and clustering events.
4.1 Introduction

The proliferation of Web photo-sharing applications has resulted in a large amount of personal photos being available for public access. As specified in Chapter 1, today Flickr alone contains more than 5 billions\(^1\) photos, of which 100 million are geotagged\(^2\). Although, recent developments in search technology have made the access to these photos easier than before, users still need to browse through a big amount of information, or do text-based searches, which are often imprecise and time-consuming. Because clustering allows us to group similar images together, image clustering can help the user with such processes [53]. First, since images can be accessed at a group level, it allows for better and more user-friendly navigation. Second, since searches can be done at a cluster level, we get a reduced search space and improved re-ranking of the search results.

Images from this kind of collections, may be clustered using their tags, and a tag-based clustering algorithm has already been proposed in Flickr\(^3\). However, to our best knowledge, this clustering algorithm does not take the time or the geographical information into account, but seems to be based on statistics computed on the tags only.

To illustrate why this is a limitation, a specific image taken at a specific place and a specific time may belong to several semantic levels. For example, a photo of a group of people in front of the Tour Eiffel during the New Year’s celebration may belong to three semantic layers. The first layer is related to the event of New Year’s celebration itself, identified by the time information. The second layer is related to the fact that it represents a tourist attraction, and it is strictly related to the visual features. The third layer is on the geographic location, the Paris area. In other words, an image can be placed in a two-dimensional space, represented by time and space, which calls for a new clustering approach. Traditional text-based clustering approaches often only consider the annotations, which can be inaccurate and often incomplete due to their subjectivity and limitation in coverage.

The main objective of the framework proposed in this chapter is to develop a method that can overcome the above limitations, by including both time and locational dimensions, as well as the textual annotations, in the event extraction for the clustering process. We achieve this by extracting event clusters from large datasets such as Flickr images, using temporal and spatial information, and by adopting and extending an incremental clustering algorithm. For this purpose, we have chosen the Suffix Tree Clustering (STC) algorithm [187] as our core clustering algorithm. Due to its

---

\(^1\)See http://blog.flickr.net/en/2010/09/19/50000000000/.

\(^2\)See http://blog.flickr.net/en/2009/02/05/100000000-geotagged-photos-plus/.

\(^3\)See http://blog.flickr.net/en/2005/08/01/the-new-new-things/.
incremental nature, STC is particularly suitable for dynamic and a large amount of
data such as the Flickr image collection, and it appear suitable for online applica-
tion. And, as we will discuss later, STC has been shown to have good performance
characteristics [187].

Most work in event detection has mainly been done on text documents, where an
event has been defined as something happening in a certain place at a certain time
[182]. Focusing on our context, this definition is too limited. Thus, we extend this
event definition to something happening in a certain place at a certain time and
tagged with a certain term.

Bearing this in mind, the main contributions are as follows. First, we suggest a new
event extraction-based clustering approach that (1) addresses the above challenges
and the limitations of image annotations, and (2) taking into account the noises
in the annotations. As part of this, we extend a dictionary tailored to the Flickr
images. Using this dictionary, we can analyze the image tags and remove noises
from the annotations. A result is that we gain a reduced term space. Second, we
propose to extend the well-known clustering algorithm STC to deal with both the
textual annotations, and time and locational information. Here we also use the set of
annotations in an image as the snippet of a document and apply this in the clustering
process. Third, a clustering algorithm such as STC was originally proposed for text
document clustering only. To our best knowledge, the way we use and extend this
algorithm in our approach is new. Hence, we believe this, in combination with
event extraction on a large set of images, is in itself a contribution. Forth, we do
an analysis of the behavior of the algorithm using different granularity of time and
space, and evaluate its performance on two different datasets gathered from Flickr.

The remainder of this chapter is organized as follows. Section 4.2 formally defines
the problem addressed. Section 4.3 discusses other work related to ours. Section 4.4
gives an overview of the STC algorithm and briefly discusses how it is used in this
work. Section 4.5 describes in detail the principle behind our approach. Section 4.6
presents and discusses the results from the evaluation of our method. Section 4.7
concludes the chapter and outlines directions for further research.

4.2 Problem Definition

This section presents the problem that we address in this chapter. First, to give
an overview of the problem, we describe the input data to our system. Then, we
explain the intuition behind our ideas. Finally, we elaborate on the main goal of our
approach and the intended output.
4.2.1 Input Data

Let $F$ be our dataset consisting of a collection of images downloaded from Flickr. We assume that every image is surrounded by metadata containing different kinds of contextual information related to the picture. In addition, we consider the temporal information related to when a picture was taken, and the geographical information regarding where a picture was taken. Hence, each image $I \in F$ is represented as

$$I = \{T, g, d_t\},$$

(4.1)

where $T$ represents the set of annotations for $I$, including the title of the image, $d_t$ denotes the date and time when the photo was taken, and $g = \{\text{lat}, \text{lon}\}$ is the pair of real numbers representing latitude and longitude coordinates. It can be assumed that every Flickr image included in our dataset contains all this information.

4.2.2 Hypothesis and Core Idea

According to our definition of events in media-sharing applications mentioned in Section 8.1, an event can be characterized by three parameters: the time and geographical information, and one or more tags associated to the actual image. Images related to an event are grouped in the same time slot, in the same geographical area and preferably associated with one or more tags. Thus, formally an event cluster can be defined as follows:

**Definition 4.2.1** An event cluster $E = \{I_1, \ldots, I_N\}$ is a set of images, all related to the same event. If $E$ is an event cluster, then it can be identified by a time interval $V$ and a geographical area $G$ in where all the images of a specific event cluster are enclosed and tagged by the same tag $t$. In other words,

$$E : \text{event cluster} \Rightarrow \{\forall(I_i, I_j \in E), G_i = G_j \land V_i = V_j \land t_i = t_j\}. \quad (4.2)$$

Note that the converse can never be true. This means that not all sets of images grouped in a time interval, geographical area and annotated by a tag, represent a specific event. In fact, if we consider some pictures related to a popular landmark (e.g., Piccadilly circus in London), then such pictures can be distributed over some (unrestricted) periods of time, but pictures of an event are concentrated in a specific time interval. This is despite the fact that both sets of pictures might be spatially aggregated from a certain geographical area. Fig. 4.1 illustrates two different distributions in space and time of both a landmark-related tag and an event-related tag. As depicted, on one hand both of the spatial distributions of the two tags are concentrated in a specific area (see Fig. 4.1a and Fig. 4.1b). On the other hand, the
temporal distribution of the [piccadilly circus]-pictures is spread over a long period of time -i.e., on year (see Fig. 4.1c), but for the [chelsea flower show]-pictures, the distribution has a peak on the time the event took place (see Fig. 4.1d).

To find a definition that make the backward implication in Definition 4.2.1 also to be true (i.e., true double implication), we establish an ideal assumption for the pictures related to an event in the following definition:

**Definition 4.2.2** For an event, the set of images $S_{VGt}$ taken in a specific area $G$ at a given time interval $V$ with a particular tag $t$ is the same set of images $S_{Gt}$ taken
in the same area $G$ with the same tag (without taking the time into account). In this case, the event is identified by the triplet \( \{V, G, t\} \). This means that

$$E : \text{event cluster} \Leftrightarrow \exists (V, G, t) : S_{VGt} = S_{Gt}$$

Fig. 4.2 further explains the idea of Definition 4.2.2. As depicted, for a set of images representing an event, the ideal case is that the set of images are grouped by the spatial area where the picture were taken, the temporal interval when the event took place, and a tag $t$ is the same for the set of images grouped by only the same geographical area and tag $t$.

![Figure 4.2: Illustration of the ideal assumption.](image)

With the above ideal assumption, we can use Definition 4.2.2 as our formal concept of event in the domain of media-sharing applications. This definition will thus be the core of the algorithm proposed in this chapter.

### 4.2.3 The Goal of the System

The goal of our system is to extract the set of event clusters \( \mathcal{H} = \{\mathcal{E}_1, \ldots, \mathcal{E}_M\} \) from a set of geotagged and labeled images \( \mathcal{F} \), according to our event definition in Definition 4.2.2. Every \( \mathcal{E}_i = \{I_1, \ldots, I_k\} \) will group the set of pictures representing an event together. To achieve this, our approach consists of two essential steps: (1) event clustering and (2) event extraction. The former aims at properly grouping pictures belonging to the same event, from a dataset containing images of different events. The latter aims at extracting clusters of pictures not only of the same event but also from an heterogeneous collection of photos (i.e., both event-related and non-event-related). Each \( \mathcal{E}_i \) will be labeled with a set of representative terms.
4.3 Comparison with Previous Approaches

As can be inferred from the discussion in Chapter 3, this domain has lately been a subject for intense research. In this section, we further elaborate on two specific approaches that we believe the most related approaches, as they seem to address the same problem discussed here - i.e., extracting and grouping event pictures, within the domain of community photo collections. More specifically, these two approaches are those proposed by Quack et al. in [129], and Papadopoulos et al. in [119].

First, in [129], the authors present a method divided into two main steps to mine objects and events from Flickr: clustering and classification. In the clustering step, they employed a hierarchical agglomerative clustering algorithm to cluster so-called interesting entities. They built two full distance matrices based on textual similarity, including picture tags, titles and descriptions, and visual similarity, relying on the number of local and scale invariant SURF features matching between two pictures [13]. Further, the main goal of the classification step is mainly to classify the content of the cluster into objects (i.e., landmarks) and events. For this step they use the temporal information and the user information connected to each picture. In particular, the classification is based on two features: time span – i.e., the number of distinct days at which the pictures in a cluster were taken, and number of users – i.e., the number of unique users contributing that have taken the pictures in the cluster, normalised on the cluster size.

Second, similar to the above approach the method by Papadopoulos et al. [119] also divided the entire process into two main steps, consisting of clustering and classification. However, the way they applied clustering and classification is different. In one hand, this approach seems to be inspired by that of Quack et al. [129]. On the other hand, they introduced new features to address additional problems that were not completely addressed before. For example, the clustering step was done through community detection algorithm rather than agglomerative clustering algorithm. More specifically, this approach works as follows. First, the pictures form two similarity graphs, one by employing textual similarity as image similarity and the other by considering the visual similarity. Then, the clustering is performed on the hybrid graph, a result from combining the two similarity graphs. Moreover, the subsequent classification is performed by employing SVN and kNN based on event tag profile features and landmark tag profile features, as well as the time span for the cluster and the normalised number of unique users. In particular the two tag features are based on a training set of labeled landmark and event pictures. From this set, common tags are discarded before the classification, and then two basis vector are defined according to which class the tags belong - i.e., either landmark class or only to event class. After the classification, landmark classified clusters are
merged based on the spatial information so that two or more clusters belonging to
the same landmark class become one cluster. Finally, when this is done each cluster
is assigned a meaningful label.

Based on their respective experimental results, both of the approaches above have
achieved good and promising results with respect to both landmark and event de-
tection. However, to our knowledge, they both rely on a time consuming off-line
step. For example, the approach by Quack et al. [129] needs such a step to extract
the visual features from the pictures, building the full distance matrix and cluster
with hierarchical agglomerative clustering. And, for the approach by Papadopoulos
et al. [119] such an off-line step is necessary for creating the similarity graph and
extracting clusters.

One of the main motivations for this work has been to cope with this challenge by
not applying neither off-line standard clustering such as the agglomerative clustering,
nor off-line classification that requires training. Instead, we suggest a method relying
on a formal definition of an event as discussed in Section 4.2.2 and then use a data
structure based on this to allow us to deal with frequent changes in the picture
collection, as well as the scalability issues. More specifically, we index our dataset
in a scalable suffix tree. Then, we extract the clusters and the events from this tree-
based index. Since users may add new pictures frequently, we stress the importance
of enabling the execution of the entire process on-line and incrementally, including
cluster extraction, as well as dealing with large dataset (scalability). This is why
we adopt and extend the Suffix Tree Clustering (STC) algorithm. With respect to
performance, as we will discuss in the next section, this algorithm has been proven
efficient, with a computational time cost of $O(n)$ for the construction of the index
tree, where $n$ is the number of documents to index, and time complexity for the
search of a pattern string composed by $k$ words being constant $O(k)^4$. So, although
we need to do the indexing off-line, we believe it is not as time-consuming as the
previous methods, and that the index can be extended or updated on the fly, since
the insertion of a new document to the index has constant time complexity.

### 4.4 The Suffix Tree Clustering Algorithm: An Overview

As already mentioned the core of our system is based on the Suffix Tree Clustering
(STC) algorithm, mainly used to cluster Web documents or textual documents.
Hence as can be inferred by the name, this clustering algorithm is based on the

---

*In our method, the maximum value of $k$ is 3*
4.4. The Suffix Tree Clustering Algorithm: An Overview

*Suffix Tree* data structure. In this section we briefly describe the data structure and the clustering algorithm.

### 4.4.1 Suffix Tree Data Structure

A suffix tree is a tree structure mostly used for representation of a string [9]. It is built with all the suffixes of the string – i.e., given a string $S$, a suffix tree is a compact tree containing all the suffixes of $S$ (see Fig. 4.3 for an illustration). The structural characteristics of this tree are as follows: (1) it is a rooted directed tree, (2) each internal node other than the root has at least two children, and (3) each edge leaving from a particular node is labeled with a non-empty substring of $S$.

![Figure 4.3: Example of suffix tree built over the string $S=\text{Papua}$](image)

Due its characteristics, the suffix tree data structure is suitable in many fields such as Information Retrieval and Bioinformatics, involving string applications like Pattern Matching, Longest Common Substring and Genome Sequence Alignments. The most important advantage is its linear time computational cost, which according to the *Ukkonen’s algorithm* [173], is $O(m)$, where $m$ is the number of character in the string. However, the main drawback is that computational cost in space is quadratic, but this has also been tackled in several works [81]. When more than one string are stored in a tree, a *Generalized Suffix Tree* can be created, which is built over all the suffixes of all strings from an input.

### 4.4.2 The Clustering Process

The STC was first introduced by Zamir and Etzioni in [187], and was originally used for clustering text and Web documents [58]. The original version of the clustering algorithm consists of three different steps [187]:

1. **Preprocessing**: Convert the input text documents into a suffix tree representation.
2. **Cluster Identification**: Use the suffix tree to identify clusters of similar sequences.
3. **Postprocessing**: Refine and clean the clusters to improve their quality.

These steps are explained in detail in subsequent sections of the document.
1. **Document Cleaning**: Given a set of documents \( C = \{ D_1, D_2, ..., D_N \} \), a suffix tree is built over all suffixes of the documents in \( C \).

2. **Base Clusters Identification**: Nodes representing more than two documents will be selected as *Base Cluster* \( B \). A score \( s(B) \) is assigned to each base cluster as
   \[
   s(B) = |B| \cdot f(|P|),
   \]
   where \( |B| \) is the number of documents in the base cluster \( B \) and \( P \) is the length of the common phrase.

3. **Base Clusters Merging**: The base cluster is ranked according to the score \( s(B) \), and the top \( k \) clusters are combined in a single-link clustering algorithm. Next, the similarity graph is created, with the base clusters as nodes and their similarities being used to generate the edges. This means that two base clusters \( B_i \) and \( B_j \) are connected if and only if their Jaccard coefficient is bigger than 0.5 – i.e.,
   \[
   J(B_i, B_j) = \frac{|B_i \cap B_j|}{|B_i \cup B_j|} > 0.5.
   \]

Though the STC algorithm was first introduced more than a decade ago, to our best knowledge, it has not been used in approaches like ours before. Nevertheless, the algorithm has been applied in textual Web documents with great success [187, 27].

One of the main reasons we adopt STC in our work is because it is *snippet tolerant*. Being snippet tolerant, STC can produce high quality clusters from the document snippet, instead of the whole document (see also below), which fits very well with our type of data. Here, every set of image annotations is considered as the snippet for the clustering process.

Compared to conventional clustering approaches – e.g., k-means or hierarchical agglomerative clustering – previously used for image clustering, the designated STC algorithm has several advantages. For example, in contrast to conventional clustering, with STC we can perform the clustering without needing to know any initial number of clusters. In addition, the following features make STC more optimal to our problem than conventional clustering approaches:

- **Phrase based model**: STC does not treat document as *bag of words* but as *sequence of words*. Hence, this clustering algorithm groups documents with common sub-phrases together.

- **Linear time tree construction**: As mentioned before, the construction of the Suffix Tree is linear in time with respect to the number of items.

- **Snippet-tolerance**: When clustering documents, the algorithm only uses *snippet* rather than entire document to produce high quality clusters. Snip-
pets are short representative part of the text document. In our experiments an image is represented by the title and the textual annotations, which in turns form a set of words representing that image.

- **Overlap**: Documents may be shared among different clusters. This means that they can belong to different topics. In the image context, an image can contain more than one semantic concepts, and thus it may be labeled or classified with several different labels/concepts.

Note that as we base our approach on the hypothesis in Definition 4.2.2, to our knowledge, it is not straightforward to use conventional clustering. Further, referring to previous researches [129, 119], to extract event clusters we may need a classification stage, which would make our approach more complicated and less suitable for online applications.

### 4.5 Event Extraction and Clustering

Referring to the discussion in the previous section, image event extraction and clustering is performed on a dataset composed by a textual representation of pictures. This textual representation consists of textual annotations that are enriched with temporal and spatial information.

Keeping this in mind, the basic idea underlying our approach is as follows. First, the clustering process produces a set of candidate event clusters. These are groups of images that are likely to represent event clusters. Second, this set of clusters are filtered following our hypothesis to produce the event clusters. Third, after filtering, semantic similar clusters are merged to one cluster. As part of this, an analysis of all spatiotemporal adjacent subtrees is done to get images belonging to events that were left out from the previous step. Forth and finally, the event clusters obtained are marked with labels.

Fig. 4.4 shows the architecture of our system. It is divided in two parts: an offline part, and an online part. The former deals with the offline data collection and construction of support structures, while the latter deals with the construction of clusters and extraction of event clusters. This part also includes a refinement step, to merge spatiotemporal adjacent subtree clusters and event annotations.

In addition to the main parts, the system has submodules that we elaborate in the following.
4.5.1 Cleaning

The first step after acquiring the data is a pre-processing step, mainly consisting of cleaning the image annotations. The representation of the image $I = \{T, g, d_i\}$ is transformed into $I' = \{T', g, d_i\}$, where all stopwords are removed from the annotation, and stemming is performed. First, all capitals in the textual annotation are converted to lowercase. Next, many annotations in Flickr images have semantically irrelevant terms, or they are so common that they do not contribute much in the discrimination of clusters. For example, camera names like nikon, canon or other terms like jpeg and geotag are very common and may safely be removed. Other common terms that can be omitted are those referring to time information such as january, february or the short versions as jan, feb. Further, we can safely remove frequent words containing digit such as oct2009 and 12may. Although they contain intrinsic temporal information, they are not useful in the clustering process. For this reason, we also remove all terms containing digits.

We divide our extended stopword list in three classes: (1) temporal terms, (2) camera-brand related terms and (3) general noise. These list came from the observation of the most frequent used tags in the annotation of pictures in our datasets. The list of terms in the stopword vocabulary can be found in Table 4.1.

As can be inferred, the main benefit of this step is the reduction of the space needed to construct the Suffix Tree. This is in addition to the fact that stopword removal also contributes to improve the quality of the clustering process, by avoiding noises in the
4.5. Event Extraction and Clustering

Temporal terms

january, february, march, april, may, june, july, august, september, october, november, december, jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec, summer, winter, autumn, fall, spring

Camera related terms

agfa, canon, nikon, tamron, sony, powershot, pentax, eos, reflex, polaroid, epson

General Noise

flickr, photo, image, picture, geotagged, geotag, geo, jpeg, cc, stockphoto, resolution, iphone, ipod, img, dsc, raw, lat, lon, jpg, gif

<table>
<thead>
<tr>
<th>Table 4.1: Extended Stopword List</th>
</tr>
</thead>
</table>

annotations. Further, by applying stemming we may both reduce the search space and improve the retrieval performance [75]. Here, we apply the Porter Stemmer algorithm [127] to reduce inflected and derived words to their stem.

Section 4.6 provides a more detailed analysis of the general effects of this cleaning step.

4.5.2 Annotation Expansion

After the cleaning is done, the tags in the images are reduced in size. Thus, \( I' = \{T', g, d_t\} \) is now smaller than the original \( I \). However, to further improve the clustering process, we have to extend the tags again. So in this step we extend the textual annotations by including the information about time and location from the EXIF\(^5\) data. Note that this expansion will not have any negative effects on the performance of the algorithm since we can construct the Suffix Tree in linear time, and the search within the tree has a logarithmic cost.

Formally, let \( I' = \{T', g, d_t\} \) be the image representation where \( T' = \{t'_1, t'_2, ..., t'_L\} \) is the set of tag associated with the picture \( I' \). Then \( t'_i \in T' \) may be a term or a sequence of terms. We will apply a tag extension process on the set \( I' \). The output

of that process will be the set $I'' = (T'', g, d_t)$ where $T'' = \{t''_1, t''_2, \ldots, t''_L\}$ and

$$t''_i = \text{concat}(s_1(d_t), s_2(g), t'_i),$$

where $\text{concat}$ is the string concatenation function, and $s_1$ and $s_2$ denotes the transformation from temporal data and geographical position, respectively, to strings – i.e., time and space discretisation. The function $s_1$ and $s_2$ will thus define the granularity into which time and space are divided.

To summarize, Fig. 4.5 shows an example of how tag expansion is performed. In Section 4.6, we also analyze the effects of event extraction performance as function of the granularities of time and space.

![Figure 4.5: Example of annotation expansion. The original tag set $T'$ is shown on the left side. After the extension with temporal-string and location-string, the set becomes $T''$.](image)

4.5.3 Suffix Tree Construction

At this point, the set $T''$ in the triplet denoting $I''$ is the set of snippets representing an image. The Suffix Tree is built on this set and stored on the file system. It has the same characteristics as the generalized suffix tree explained in Section 8.3.

Fig. 4.6 illustrates our suffix tree. As mentioned before, every image is represented by a collection of tags. In our representation, the first two term will represent the temporal and spatial information, and the third part of the textual annotation. The suffix tree will group pictures sharing the same suffixes together. The annotation expansion step enriches the existing annotations of the pictures with a temporal
and a spatial information. In our suffix tree, we distinguish between the following three kinds of branches, along with their corresponding nodes: **temporal branches**, **positional branches** and **tag branches**.

**Temporal branches** are those labeled with strings representing the temporal information. They come from the transformation of the time information of the pictures applying the function $s_1$. Nodes having an in-link of this kind will group pictures taken in the same temporal interval, according to the granularity expressed in $s_2$.

**Positional branches** include all branches labeled with a string representing a geographical area. Nodes with this kind of in-link will group images taken in the same geographical area, according to the function $s_2$ used for the transformation of the geographical information.

**Tag branches** include all branches labeled with a textual tag – i.e. those provided in Flickr by the owner of the images. Nodes with an in-link branch of this
kind will create groups of all pictures tagged with the same specified term.

As can be inferred from this, each node of the Suffix Tree collects all pictures with the same suffix in the annotation into a cluster. Recall that in the original algorithm, each node representing a possible cluster is called base cluster, and it has a unique label. In our algorithm, the label for a specific node consists of a sequence of labels for all the nodes that have to be passed through when traversing the tree from the root to this node.

### 4.5.4 Candidate Event Clusters Extraction

In contrast to the original STC, to extract our base clusters, we do not need to consider all nodes of the suffix tree, though the nodes group two or more documents. Instead, we first prune the set of nodes according to the definition of an event given in Section 4.2. That is, if it does not obey the hypothesis, then it is not likely to be event, and will be removed.

All nodes labeled with a temporal label, a geographical label and a tag label are possible events. These constitute what we call candidate event clusters, denoted by \( S_{VGt} \). They are extracted by traversing through all nodes of the Suffix Tree that are reachable by the sequence of temporal branch, positional branch and tag branch. Each such a cluster is tagged with the sequence of labels of the branches through which the algorithm passes. This means that each candidate event cluster is a collection of pictures grouped by a certain time slice, a certain geographical area, and tagged with a certain term.

Recalling Definition 4.2.1 in Section 4.2, the collected candidate event clusters make the right side of the implication to be true. Note, however, that this is not enough to obtain an event cluster. Based on Definition 4.2.2, we must compare \( S_{VGt} \) with the set of images grouped by the same geographical area \( G \) and the same tag \( t \), \( S_{Gt} \).

This hypothesis always holds because a tag representing an event can only belong to a single combination of date/time and a geographical area. It still holds even if we have a situation where an object (image object) or a place appear in several images taken over a long period of time. Although the tags for these images have the same geotags but different time tags, their combinations are still unique.

Ideally, to capture this uniqueness, the images in \( S_{VGt} \) must be the same as those of \( S_{Gt} \) in our tree structure. Thus all candidate event clusters \( S_{VGt} \) that are labeled with a temporal label, a geographical label and a tag label will be compared with the set \( S_{Gt} \). In practice, considering the noises in the tags, and in the temporal and spatial information associated to a picture, we can relax this assumption. This
means that for a given \( S_{VGt} \), the assumption for an event cluster become

\[
\frac{|S_{VGt} \cap S_{Gt}|}{\max(S_{VGt}, S_{Gt})} \geq K,
\]

where \( K \) is a parameter. With \( K = 1 \), we get \( S_{Gt} = S_{VGt} \). Using a smaller \( K \) value, we can control this assumption with respect to noises in the dataset, and the spatio/temporal granularity used for the extension of the tags. In Section 4.6, we discuss the effects of this parameter in details.

### 4.5.5 Merging Candidate Event Clusters

Let \( \varepsilon = \{e_1, ..., e_n\} \) be the set of events extracted from the previous step. Each event \( e_i \) consists of a set of images labeled with \( L_{e_i} = \text{concat}(s_1(d_i), s_2(g), t_i') \) directly derived from the subtree associated with the event cluster \( e_i \) from the root to the leaf, and by collecting the label on the branch/node.

Now, the issue to be solved is that there may exist two or more different event clusters representing a single event. In the original STC algorithm, where all the branches have the same semantic meaning, this happens because documents can share more than one phrases. In fact, the STC algorithm may produce overlapping clusters since the base clusters can share the same documents (i.e., one or more documents can belong to more than one group, and thus appearing in more than one cluster) [127]. A merge step was introduced to solve this issue. A binary similarity measure between base clusters is used to measure the overlapping degree of the documents sets:

\[
\frac{|B_i \cap B_j|}{|B_j|} > 0.5 \text{ and } \frac{|B_i \cap B_j|}{|B_i|} > 0.5,
\]

where \( B_i \) and \( B_j \) are the two base clusters. The problem of this similarity measure is that it is not suitable when the ratio, \( |B_i|/|B_j| \), of the size between the two base clusters is too small or too big. In such cases, if a smaller set is subset of a bigger one, they must readily be merged, but they will not satisfy the Eq. 4.8. To deal with this problem, and to enhance the similarity measure by including the non-overlapping parts, we can apply a combination of the two following measures to measure the overlap between two clusters, which were originally proposed in [56]:

\[
S(B_i, B_j) = \alpha \text{overlap}(B_i, B_j) + (1 - \alpha) \text{Sim}(B'_i, B'_j)
\]

where:

\[
\text{overlap}(B_i, B_j) = \frac{|B_i \cap B_j|}{\min(|B_i|, |B_j|)}
\]
and the second part is related to the cosine similarity between the base clusters where \( B'_i = B_i - |B_i \cap B_j| \) and \( B'_j = B_j - |B_i \cap B_j| \).

Once the similarity measure is defined the merging process can be performed by first constructing a graph where the nodes are the base clusters. Two nodes are connected if they satisfy Eq. 4.8. Connected components are detected to extract the merged base clusters. In citeDBLP:conf/eiswt/2007 an iterative process was performed to calculate the pairwise similarity over all the base clusters according to Eq. 4.10, and then by selecting the maximum \( S(B_i, B_j) \) greater than a parameter \( k \) and merging the clusters \( B_i \) and \( B_j \). The parameter \( k \) is tuned to fit with the number of merged clusters.

As we explained before, our Suffix Tree, is built by using three different kinds of labels, each of which with a different semantic meaning. In addition, we use a novel way to explore the Suffix Tree to detect the candidate event clusters. In our case, we can have two different needs to merge candidate event clusters: (1) **Semantic Similarity** and (2) **Spatiotemporal Neighborhood**.

**Semantic Similarity**

Like in the original STC algorithm, we need to merge all candidate event clusters that share the same pictures – i.e., merging tags related to the same event. To do this, we determine the similarity among candidate clusters, by computing the overlap function as defined in the Eq. 4.10. We refer to this function as \( \Psi(e_i, e_j) \) where \( e_i \) and \( e_j \) are two event clusters extracted in the previous step – i.e.,

\[
\Psi(e_i, e_j) = \frac{|e_i \cap e_j|}{\min(|e_i|, |e_j|)}.
\]

(4.11)

Inspired by [187], we build an event cluster graph to facilitate the merging of similar clusters based on Eq. 4.11. The nodes of this graph will be the event clusters. If two nodes for two event clusters \( e_i \) and \( e_j \) have the similarity value such that \( \Psi(e_i, e_j) = 1 \), then we add an edge between \( e_i \) and \( e_j \). So, to find out which clusters to be merged into single clusters, we traverse the graph to find all nodes that are connected, and then merge these.

As for our structure, we do not need to explore all the pairwise similarity \( \Psi(e_i, e_j) \) between the event clusters. In fact, since this merging step merges clusters that share the same pictures, they must implicitly belong to the same geographical area and temporal interval, depending on the chosen granularity. The graph and the connected components are detected only from event clusters belonging to the same temporal interval and spatial area. We call this step in-spatiotemporal merging and the output is a set of merged event clusters \( \tilde{\varepsilon}_i = \{E_{i1}, ..., E_{iK}\} \), where \( E_j = \{e_{j1}, ..., e_{jL}\} \). The label for the new cluster is now a label derived from the merging of the labels for the overlapping clusters.
4.6. Evaluation

Spatiotemporal Neighborhood
Depending on the granularity chosen in the spatial and temporal dimensions, it may happen that only a part of the pictures of an event is detected. This is because they may have a (limited) shaped spatial distribution or be extended in an area/interval bigger than the dimension of the temporal and spatial granularity. In this case, we must perform a spatiotemporal neighborhood analysis. And, inspired by the well-known DBSCAN clustering algorithm [45], we apply a refinement step.

This refinement step explores the spatiotemporal neighborhood in a recursive way to find a specific group of pictures that can be included in an event cluster. Algorithm 1 shows the pseudocode of our refinement method which is executed when the \texttt{Refine}(N, C, T) method is called. Here $N$ is a set of the 26-connected neighbors of the event clusters $E_i$, in the spatiotemporal dimensions and $C$ is the set that will eventually contain the merged pictures. The method \texttt{GetPictures} returns the set of pictures in $N_i$ annotated with one of the tags in $T$. The set $T$ will contain the textual annotation related to the different $e_{ik} \in E_j$, i.e., the tag of the tag branch of the associated node. The function \texttt{GetNeighbors}(P) returns all the 26-connected spatiotemporal neighbors that have not been visited before. We call this step \textit{neighbor-spatiotemporal merging}.

**Algorithm 1** \texttt{Refine}(N, T, C)

1: Mark all $N_i \in N$ as visited
2: for \ do $N_i \in N$
3:  $I_T = \text{GetPictures}(N_i, T)$
4:  if ( then $I_T \neq \emptyset$)
5:    $M = \text{GetNeighbors}(N_i)$
6:    if ( then $M \neq \emptyset$)
7:      \texttt{Refine}(M, T, C)
8:    end if
9:  Add $I_T$ in cluster $C$
10: end if
11: end for

4.6 Evaluation

In this section, we present the results from our experiments. The main goal is to test the performance of our approach. First, we present the two datasets. Second, we describe the evaluation measures applied. Third, we analyze the effect of the cleaning process mentioned in the previous section. Forth, we study the effect of varying the
time and space granularity. Fifth, to find out about the behavior of the parameter \( K \) in Eq. 4.7, we assess how the different values of \( K \) affect the clustering and event detection performance. Sixth and finally, we present a user-based evaluation of our approach.

### 4.6.1 Dataset

We evaluate our approach using two different datasets. In both datasets, we will use *tag* to refer a single annotation of the set of annotations for each picture – e.g., *Tour Eiffel* is one tag, and *term* as a single word for the tag – i.e., the tag *Tour Eiffel* contains two terms.

Now, the first dataset consists of pictures taken in the area of San Francisco, collected from Flickr using the Flickr Api\(^6\). For simplicity, we refer to this dataset as *San Francisco* dataset. To build this dataset, we gathered all images uploaded between 1 April 2008 and 31 October 2010. The spatial area is a rectangle with a minimum latitude of 37.650223 and a maximum latitude of 37.829388, and a minimum longitude of \(-122.534\) and a maximum longitude of \(-122.349\) (see also Fig. 4.7). Within this time frame and spatial rectangle, the total number of images collected is 231,222.

![San Francisco area map](image)

**Figure 4.7:** The area in *San Francisco* considered for gathering the Flickr photos of the San Francisco dataset

In this dataset the pictures are tagged with a total of 1,752,756 tags, where 207,986

---

\(^6\)http://www.flickr.com/services/api
of these are unique. Further the number of terms is 3,187,223 of which 94,442 are unique. The average number of terms per tag is 1.85, while it is 3.78 considering only the title. The average number of tags per picture (considering the title) is 7.53. Finally, 12,311 pictures do not contain any title, while 45,234 pictures do not contain tags. All images in our dataset contain at least or title or tags. This statistic is summarized in Table 4.2.

<table>
<thead>
<tr>
<th></th>
<th>San Francisco</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images</td>
<td>231,222</td>
</tr>
<tr>
<td>Tags</td>
<td>1,752,756</td>
</tr>
<tr>
<td>Unique Tags</td>
<td>207,986</td>
</tr>
<tr>
<td>Terms</td>
<td>3,187,223</td>
</tr>
<tr>
<td>Unique Terms</td>
<td>94,442</td>
</tr>
<tr>
<td>Terms in Title</td>
<td>828,223</td>
</tr>
<tr>
<td>Unique Terms in Title</td>
<td>94,442</td>
</tr>
<tr>
<td>Avg.Tags/Pictures</td>
<td>7.58</td>
</tr>
<tr>
<td>Avg.Terms/Tags</td>
<td>1.82</td>
</tr>
<tr>
<td>Avg.Terms/Title</td>
<td>3.78</td>
</tr>
<tr>
<td>Images without title</td>
<td>12,311</td>
</tr>
<tr>
<td>Images without tags</td>
<td>45,234</td>
</tr>
</tbody>
</table>

Table 4.2: Summary of the information related to the San Francisco dataset

The second dataset is the same as the one used in [118], which was used for the same purpose. It is composed of around 200K pictures collected in the area of Barcelona. Here, we only consider pictures containing geographical information, and after performing the preprocessing step such as tag cleaning, the number of pictures in this dataset becomes 160,123. For simplicity, we will refer to it as Barcelona dataset.

We mainly use the San Francisco dataset to investigate the effects of cleaning of tags and to analyze the performance of the extraction of the event clusters using different values of the parameter $K$, as presented in Eq. 4.7. Whereas, we mainly use the Barcelona dataset to do the user-based evaluation. For both datasets, we evaluate the performance over different space and time granularity. And, for each image in the datasets, we consider the title of the pictures as part of the tag set, and we only included pictures with geo-tags.
4.6.2 Evaluation Measures

As can be inferred by our discussion above, we perform two separate evaluations on the two datasets. In both evaluations, however, we evaluate the clustering performance using information retrieval metrics. This is because we use the clustering results to retrieve event clusters and rank them according to a score function based on the size of each cluster. Examples of previous approaches that have adopted such an evaluation principle are those presented in [187] and [58].

Note that none of the datasets has any ground truths. For this reason, we checked each cluster to decide whether they were event clusters or not. And, this lack of ground truths is part of the reasons for performing our user-based evaluation (see Section 4.6.6). For this purpose, we use precision at $k$ position - i.e., $P@k$ [22] as the main measures. $P@k$ is the fraction of the first $k$ extracted clusters that are correctly identified as events. Thus, this measure can be defined as follows:

$$P@k = \frac{|C_k \cap R_k|}{k},$$  \hspace{1cm} (4.12)

where $C_k$ is the set of top-$k$ extracted clusters, $R_k$ the set of relevant clusters from the set $C_k$ - i.e., the set of all clusters are events, and $|C_k \cap R_k|$ is the number of extracted cluster that are correctly identified as events.

4.6.3 Dataset Tag Cleaning

As discussed before, the main purpose of the cleaning process is to remove noises, and to keep the most representative tags and title for each picture. As part of the cleaning process, we first perform a tokenization step on the title and on the picture tag to extract terms from the text. After this, we run a stopword removal step, using both a standard English stopword list and the extended stopword list discussed in Section 4.5.1, and listed in Table 4.1. Further, for the event extraction, low frequency terms are not representative since they do not have a relevant spatio-temporal distribution. In our datasets, these are terms used less than 5 times.

The top 20 most frequent terms are removed since this list contains terms related specifically to the area only, and thus they are not discriminative for event detection (see Table 4.3). Further, we only keep terms containing letters and remove those with digits. Finally we apply stemming on each term using the Porter Stemmer algorithm. Note that after the tag cleaning process is completed, we may safely remove all images that no longer have tags.

To summarize, Fig. 4.8 shows the effect of reducing the number of tags and terms after the cleaning process, in percentage.
### Table 4.3: Top 20 frequent terms in 4.2(a) San Francisco Dataset and 4.2(b) Barcelona dataset

<table>
<thead>
<tr>
<th>Term</th>
<th>Frequency</th>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>san</td>
<td>151,265</td>
<td>barcelona</td>
<td>171,090</td>
</tr>
<tr>
<td>francisco</td>
<td>147,889</td>
<td>spain</td>
<td>70,807</td>
</tr>
<tr>
<td>california</td>
<td>67,720</td>
<td>de</td>
<td>51,043</td>
</tr>
<tr>
<td>2010</td>
<td>37,757</td>
<td>catalunya</td>
<td>39,972</td>
</tr>
<tr>
<td>sf</td>
<td>28,386</td>
<td>la</td>
<td>32,193</td>
</tr>
<tr>
<td>united</td>
<td>27,039</td>
<td>españa</td>
<td>30,719</td>
</tr>
<tr>
<td>states</td>
<td>26,886</td>
<td>catalonia</td>
<td>29,957</td>
</tr>
<tr>
<td>img</td>
<td>26,638</td>
<td>art</td>
<td>22,788</td>
</tr>
<tr>
<td>of</td>
<td>22,268</td>
<td>live</td>
<td>22,621</td>
</tr>
<tr>
<td>s</td>
<td>22,114</td>
<td>2009</td>
<td>21,700</td>
</tr>
<tr>
<td>street</td>
<td>20,204</td>
<td>cataluña</td>
<td>21,337</td>
</tr>
<tr>
<td>2009</td>
<td>17,546</td>
<td>17</td>
<td>17,672</td>
</tr>
<tr>
<td>mission</td>
<td>17,498</td>
<td>tamron</td>
<td>16,667</td>
</tr>
<tr>
<td>sanfrancisco</td>
<td>16,796</td>
<td>sagrada</td>
<td>16,299</td>
</tr>
<tr>
<td>jpg</td>
<td>16,735</td>
<td>music</td>
<td>16,222</td>
</tr>
<tr>
<td>usa</td>
<td>16,573</td>
<td>architecture</td>
<td>15,735</td>
</tr>
<tr>
<td>ca</td>
<td>15,488</td>
<td>hara</td>
<td>15,459</td>
</tr>
<tr>
<td>bay</td>
<td>14,880</td>
<td>gaudi</td>
<td>15,360</td>
</tr>
<tr>
<td>at</td>
<td>14,253</td>
<td>2009</td>
<td>15,241</td>
</tr>
<tr>
<td>iphone</td>
<td>13,326</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.8: Effect of the cleaning on the tags and terms frequency for *San Francisco* dataset
4.6.4 Event Clusters Extraction Analysis

As mentioned previously, we extended the tags with time and location strings. The two functions $s_1$ and $s_2$ transform the real values of space and time into these two strings, meaning that the time is discretized in a time window slice and the location coordinates in a square.

Our analysis of the algorithm performance was done on two different time granularity and four different discretization of the space. We considered a time slice of 1 day and 1 week. For the location, we considered squares of 0.001, 0.002, 0.005, 0.01 decimal precision in the latitude and longitude unit measures – i.e., squares with side lengths of around 100, 200, 500 and 1000 meters (m).

When doing the experiments, we had 8 different combinations to analyze, which we compared against each other. The precision values were then computed at rank level 5, 10, 20 and 50. In addition, we did the computations for each combination of granularity of time and space.

4.6.5 Evaluation Based on Values of K

![Figure 4.9: P@50s at different value of K over different granularity in space and time (e.g., $D100$ means day as temporal granularity and 100 meters as spatial granularity)](image)

Using our San Francisco dataset we now analyze the behavior of the parameter $K$ introduced in Eq. 4.7 to relax the hypothesis of event clusters. Here, we perform the algorithm using $K = 1$ and $K = 0.8$. The results of the experiments at different space and time granularity and with different $K$ are shown in Table 4.4 and Table 4.5. A summary of the differences between the behavior with different values of $K$ is also given in Fig. 4.9.
4.6. Evaluation

Due to the lack of ground truth, though recognising the issues with subjectiveness, we decided to perform a user-based evaluation to get more independent results. As already mentioned, we use the Barcelona dataset for this evaluation process. Note, however, that even though we used the same dataset as that of [118], we could not compare our results with their results due to the differences in the scope and focus. Moreover, they also carried out their evaluation in a subjective way. Hence, to get a real comparison, we would need to use the same assessors as they used. Further, they evaluated the quality of the clusters by a geo-spatial cluster coherence measured as the mean of the geodesic distances between the center of the cluster and its members [118]. Our algorithm, on the other hand, automatically groups pictures by locations, which would, to our knowledge, give positive results anyway in this respect. Also, this would be true even if we extended the same measure over the temporal space.

To carry out our user-based evaluation, we invited at least 4 highly educated human assessors\(^7\), aged between 25 and 35. They were provided with the top-50 clusters that were detected and extracted by our algorithm, each of which could be a true event, a possible event, or not an event at all. This means that for each cluster the

\(^7\)The number of persons that participated in our evaluation varied between 4 and 8.
possible answers were yes – i.e., the cluster surely represented an event, probably – i.e., the cluster was maybe an event, and no – i.e., the cluster was clearly not an event. Each answer was given a score depending on the chosen alternative: 1 for yes, 0.5 for probably, and 0 for no. Then, the final relevance score for each cluster was calculated as the average of all scores given by the different assessors for that cluster. We used the values of this score as basis for our precision-oriented evaluation metrics. For example, suppose a cluster obtained a set of scores \( \{1, 1, 1, 0.5\} \) from four different assessors. Then, the average score of 0.875 would be the final score for this cluster. Moreover, if the top-5 cluster average scores were 1, 0.875, 0.875, 1 and 0.875, the \( P@5 \) would be the average of these, which is 0.925.

To generalise, let \( \varsigma_{ij} \) be the score of the cluster \( c_i \) given by a user \( u_j \). Then \( \varsigma_{ij} \) is specified as follows:

\[
\varsigma_{ij} = \begin{cases} 
1 & \text{if } c_i \text{ is surely an event} \\
0.5 & \text{if } c_i \text{ is probably an event} \\
0 & \text{if } c_i \text{ is not an event}
\end{cases}
\]  
(4.13)

Further, let \( \sigma_i \) denote the final score of \( c_i \), and that the number of users evaluating \( c_i \) is \( n_i \). Then, the score for \( c_i \) is

\[
\sigma_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \varsigma_{ij},
\]  
(4.14)

and the precision value at \( k \) position is given by

\[
P@k = \frac{1}{k} \sum_{i=1}^{k} \sigma_i.
\]  
(4.15)

Using the above information, we evaluated the performance of our algorithm by using the measures described in Section 4.6.2. More specifically, as with the San Francisco dataset, we computed the precision values \( P@5, P@10, P@20 \) and \( P@50 \) based on the above average relevance score values. To get our results, we executed eight runs, using different time and space granularities and \( K = 0.8 \) – the \( K \) value that give the best performance results in the previous experiment.

Table 4.6 summarises the precisions values from this experiment. Further, Fig. 4.10 shows the complete results over different granularities in time and space, and Fig. 4.11 depicts the min and max of the the average precision values over different time and space granularities.
Table 4.6: Precision of event clusters extraction over different granularity of space and time with parameter $K = 0.8$ for Barcelona dataset

<table>
<thead>
<tr>
<th></th>
<th>100 mt.</th>
<th>200 mt.</th>
<th>500 mt.</th>
<th>1000 mt.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WEEK</td>
<td>DAY</td>
<td>WEEK</td>
<td>DAY</td>
</tr>
<tr>
<td>P@5</td>
<td>0.93</td>
<td>0.93</td>
<td>0.90</td>
<td>0.95</td>
</tr>
<tr>
<td>P@10</td>
<td>0.92</td>
<td>0.88</td>
<td>0.90</td>
<td>0.93</td>
</tr>
<tr>
<td>P@20</td>
<td>0.91</td>
<td>0.83</td>
<td>0.90</td>
<td>0.87</td>
</tr>
<tr>
<td>P@50</td>
<td>0.83</td>
<td>0.82</td>
<td>0.84</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Figure 4.10: Precision of the algorithm on the Barcelona dataset over different time and space granularities

Figure 4.11: Min and Max of the average precisions at different time and space granularities

4.6.7 Discussion

The Results using San Francisco Dataset
As can be derived from the experimental results presented in the previous section, our event cluster extraction approach achieves globally satisfactory results. For
example, for all the combinations in granularity of space and time for $P@5$ and $P@10$, we measured a 100% of precision. At $P@50$ we measured an average precision of 0.9275 with $K = 1$ and an average precision of 0.96 at $K = 0.8$.

Figure 4.12: Part of the ranked list of the extracted event clusters with $K = 1$

Fig. 4.12 and Fig. 4.13 show two examples of cluster extraction with a *week* as temporal granularity and 1000 m as spatial granularity, and applying two different values of the $K$ parameter, respectively. As can be observed from this experiment, with $K = 1$ the extracted clusters that were judged *not events* were only a large personal collection of pictures taken of a person in a special situation, maybe during
4.6. Evaluation

Figure 4.13: Part of the ranked list of the extracted event clusters with $K = 0.8$

a visit in some place. The first three at rank 16, 34 and 39 consist of pictures taken in the same period of time, the same place and with a small subset of tags. They represent a collection of pictures of a user visiting certain places – e.g., a science park, a zoological park and a city. The second case at rank 41 represents a set of pictures that were taken at an exposition of dresses. With $K = 0.8$, on the other hand, the above types of clusters were pushed further down in the ranked list, which was the main reason for the increase of the precision. This increase of precision with the smaller value of $K$ was as predicted. There were a larger number of candidate event clusters available that tended to be merged with stronger candidate event clusters – i.e., those that were closest to satisfy our formal definition of event clusters. At the same time, event clusters representing collective events, such as protest, conference, parade were push up in the ranked list.
As part of our evaluation, we also calculated the average number of tags for each cluster, and the average number of clusters merged for the top 50 clusters with respect to $K$. The results from this are summarized in Table 4.7. We separated these two values – i.e., the average number of tags and the average number of clusters, since the labels for two candidate event clusters can be a subset of one another, as for example picket line and picket. As shown in Table 4.7, we counted the number of clusters after the tag merging procedure. We can observe that with $K = 0.8$, the event clusters detected were in average composed by more candidate clusters than with $K = 1$, and each merged event cluster was included in a larger set of annotations. As a result, we got an increased descriptiveness.

<table>
<thead>
<tr>
<th></th>
<th>$K = 1$</th>
<th>$K = 0.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of Merged Clusters</td>
<td>4.3</td>
<td>5.65</td>
</tr>
<tr>
<td>Average number of Tags</td>
<td>2.95</td>
<td>3.1</td>
</tr>
</tbody>
</table>

Table 4.7: Summary of the average number of merged clusters and average number of tags for each event cluster detected for different values of $K$. These are results from the top 20 event clusters.

The Results using Barcelona Dataset
For the Barcelona dataset, the results are equally promising and confirm the performance of the algorithm obtained using the San Francisco dataset. The precision values are a bit lower because there would always be some possibilities that the assessors or users used in the evaluation process could misinterpret the meaning of an event. In such cases, they might chose wrong answers. Nevertheless, the average precisions until $P@20$ were higher than 0.86 and the precisions until $P@10$ were higher than 0.9. Focusing on the best case in term of space and time granularity, the values of $P@20$, $P@10$ and $P@5$ were all higher than 0.91. The difference between the $P@50$ values of the Barcelona dataset and the San Francisco dataset could principally be due to the difference in size of the two datasets, as well as the aforementioned possible user misinterpretation. For instance, after the cleaning step, the San Francisco dataset is around 15% bigger.

4.7 Conclusions
We have presented a novel approach to extract event clusters from a large set of Flickr images. Our idea has mainly been to consider a Flickr picture as a document snippet. This has been possible since the annotations are short textual description, especially
4.7. Conclusions

after the noises have been removed. Here we analyzed the content of the annotations to identify such noises. This includes removal of stopwords and other common words that do not contribute well in cluster discrimination. As partly a result of this, we were successfully able to use an incremental clustering approach, originally proposed to cluster text documents only. Hence, we have been able to verify our hypothesis with respect to the use of a text clustering algorithm on extended tag sets. More specifically, we have adopted and extended the suffix tree clustering (STC) algorithm to achieve this. As part of our approach, we defined a function to extract groups of pictures representing events. Further, we designated a refining spatiotemporal merging function, inspired by the well known DBSCAN, to refine the merging process.

Compared to existing work, the uniqueness of our approach is the way we use a fast, incremental algorithm, originally proposed for document clustering to group large collections of similar images together. This is achieved by expanding the annotation such that we can both deal with the incomplete annotations and at the same time apply more semantic onto the clustering process.

Our evaluation has shown that this approach has indeed a very good potential. The high average $P@50$ with both datasets and both evaluation methods, over different granularity for the extracted clusters gives us a good indication of this. This also support our observation that each extracted cluster has a high consistency with respect to the number of false positive hits.
Chapter 5

Semantic Social Event Detection

In this chapter we present the framework used for the Social Event Detection (SED) task at MediaEval 2011 challenge. MediaEval is a benchmarking initiative and its main purpose is to evaluate novel algorithms for multimedia access and retrieval. Its emphasis is on multimedia and it also focuses on human and social aspects of multimedia tasks. This initiative attracts participants interested in multimodal approaches to multimedia involving different tasks such as speech recognition, audio analysis, multimedia content analysis, user-contributed information (tags, tweets), social networks, temporal and geo-coordinates and viewer affective response.

5.1 Introduction

The work presented in this chapter is part of the MediaEval 2011, for the Social Event Detection (SED) task challenge. The general purpose of the challenge was to propose an event retrieval system given two specific queries. In particular, we proposed a system to solve two specific event extraction challenges. In the first challenge, the main purpose was to retrieve all soccer events in Rome and Barcelona and in the second challenge, we were asked to retrieve all events from two specified venues in Amsterdam (NL) and Barcelona (ES) within a certain temporal range. The results of the queries were presented as groups of images - i.e., one group per event. More specific details of the challenge can be found later in the chapter.
5.2 Challenges and Dataset

The SED task proposed the two following challenges:

- **Challenge 1**: Find all soccer events taking place in Barcelona (Spain) and Rome (Italy) in the test collection. For each event provide all photos associated with it.

- **Challenge 2**: Find all events that took place in May 2009 in the venue named Paradiso (in Amsterdam, NL) and in the Parc del Forum (in Barcelona, Spain). For each event provide all photos associated with it.

For the first challenge was required to find pictures associated to soccer event – i.e., social event centred around the concept of soccer. Examples are pictures of supporters inside the stadium, pictures of the match, pictures people arriving and leaving the stadium before and after the match. The images must be related to the specified event happened in the cities of Rome and Barcelona.

The second challenge required to find all the social events happened in May 2009. In particular this events must be located in two specific venues in Amsterdan and Barcelona. The events happening in the first venue, are generally music event and several bands or player playing in same evening are considered as single event. On the other side, the second venue can host social event such as festivals with a more than one day duration.

For both the challenges, the request was to give as output a set of picture clusters. Each cluster had to group images associated to the same single event.

The dataset is composed by 73,645 pictures gathered from Flickr. The picture was provided with surrounded metadata – i.e., tags, description, timestamp, location information, etc., in form of XML files. From 80% of the pictures has been removed the gps information for simulating the lack of these information in the real Flickr world. The pictures, all geotagged, referring to the city of Barcelona, Amsterdam, Paris, Rome and London, and taken in May 2009 has been incorporated with other not geotagged pictures from [170] referring to the same period and the same cities.

5.3 System Overview

In this section we give a detailed presentation of our proposed algorithm. Figure 5.1 shows an overview of our system.
5.3. System Overview

5.3.1 Query Expansion

In a social event retrieval context, a query can be splitted and mapped in three different parts according to the general parameters characterizing an event: (1) what, i.e., which kind of event we are looking for, (2) where, i.e., the venue, name of place, city or region where the event that we are looking for takes place, (3) when, i.e., the time, interval when the event happens. In both challenges the where part of the query is expanded in the first block. For Challenge 1 the where part is created with all the stadium names in Rome and Barcelona, in all languages while for Challenge 2, it was created with the names of the two venues specified in the challenge. For both challenges, the geographical location (latitude and longitude) are also extracted. In order to retrieve these information, a set of SparQL queries are submitted to DBpedia\(^1\) database by using the Jena\(^2\) interface for java. To be more specific, for Challenge 1, names in different languages and geographical information are extracted by selecting from the DBpedia category Football_venues_in_Italy

\(^1\)http://dbpedia.org/
\(^2\)http://jena.sourceforge.net/
the occurrences based on the city of Rome and Barcelona. For Challenge 2, the geographical location and names related to the requested venues were extracted using the LastFM API\(^3\) and used as query into the LastFM database. The output of this block is a set of queries \(Q = \{Q_1, ..., Q_N\}\), where each subset \(Q_i = \{q_{i1}, ..., q_{iM}\}\) refers to all queries related to a venue and each \(q_{ij} = \{T, g\}\) is composed of two different parts: a textual part with different names of the venues, and a spatial part with a pair of real numbers representing the latitude and longitude of the venues.

### 5.3.2 Search

The queries are submitted to the search engine over the dataset. In our work, we use Solr search engine to index the dataset and perform the search. The search is done as a mix of spatial (by using latitude and longitude values) and textual search. The data are indexed based on the textual metadata, including Title, Description and Tags. The search is then performed over all the three different metadata. In particular for Challenge 2 the queries are Boolean queries with all the terms in AND, while in the Challenge 1 these conditions are more relaxed and the terms of each query are composed with the boolean operator OR. The reason for this is that in this challenge, a categorizer is provided as next step to filter out non-relevant retrieved occurrences.

### 5.3.3 Categorization

The input of this block is a list of pictures with their metadata. This module is used only for Challenge 1 to extract pictures related to a soccer event. The categorization is performed over the three textual metadata for each picture, i.e., Title, Description and Tags. The different runs will exploit the descriptiveness of each kind of metadata in the categorization process (see Section 5.4). To categorize the pictures the SemanticHacker API\(^4\) over the different textual metadata was used. The categories produced are based on the Open Directory Project\(^5\). The pictures are filtered by only keeping those categorized with a category that has radix Sports/Soccer.

\(^3\)http://www.last.fm/api
\(^4\)http://textwise.com/api
\(^5\)http://www.dmoz.org/
5.4. Experiments and Results

5.3.4 Clustering and Merging

The previous block returns a set of filtered pictures (Challenge 1) related to soccer events or pictures taken in the venues specified in the search step (Challenge 2) and grouped based on the venues. In this step temporal information will be used to group the temporal related pictures. In that way the resulting clusters are finally grouped according to their temporal and locational information. To perform the clustering process the Quality Threshold Clustering (QT) algorithm is used [64]. This algorithm does not require to specify in advance the number of clusters and even it is computationally expensive, it is used only on retrieved documents. The resulting clusters may be semantically related and belonging to the same event. To merge semantically similar clusters a graph is built, where the nodes of the graph are the clusters, and two nodes are connected if they share at least a tag representing a named entity of an event or of an artist. To extract the named entities, we use the tags and submit them as queries to LastFM for the artist names and DBPedia for event names. Clusters are merged by finding the connected component as in [143].

5.3.5 Refinement

The resulting clusters may be incomplete, i.e. the dataset may contain other pictures related to the event clusters extracted but not retrieved in the search step. The refinement module is used here to query the dataset by using the (1) top-k frequent tags and (2) top-k frequent entity names (artists and events). The results of the refinement step can still be filtered to avoid retrieving non-relevant occurrences.

5.4 Experiments and Results

This section we will first present the detail of our submitted results and second the results of the entire challenges.

For evaluating the submitted results two specific metrics has been employed: the harmonic mean (F-score) of the precision and recall for the pictures retrieved and the normalized mutual information (NMI) for comparing two collections of photo clusters.

The ground truth has been built by considering first the pictures coming from the Event Media dataset and the ones with specific machine tags, such as the lastFm tag (lastfm:xxx). For the other images has been performed a manual lookup.
5.4.1 Submitted Results

In this section, we present the different runs and their evaluation over the two specified metrics. Table 5.1 provides a summary of our results.

Challenge 1
Two different runs were performed in the first challenge. In the first run (Run 1) all the workflow of the system is performed excluding the refinement step and semantic merge between clusters. For Run 1 the categorization is performed by using only Tag metadata, while for the second run (Run 2), we also include the Title and Description metadata. From the results obtained (see Table 5.1) we can observe that including other metadata than tags resulted in a decrease of precision, probably due to the lack of descriptiveness of the other metadata.

Challenge 2
For this challenge, three different runs were performed. The Run 1 (the baseline run) executed the algorithm without including the semantic merge and refinement steps. In both Run 2 and Run 3 the semantic merge and refinement steps were performed. Semantic merge was done by considering each cluster represented by the named entity representing events or artists. Moreover in Run 2, refinement is performed by querying the top-100 tags and the temporal range in which each cluster is closed. In Run 3, we used the entity names representing artists or events extracted from the set of tags of each cluster.

<table>
<thead>
<tr>
<th></th>
<th>Challenge 1</th>
<th>Challenge 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Run 1     Run 2</td>
<td>Run 1     Run 2     Run 3</td>
</tr>
<tr>
<td>Precision</td>
<td>94.26     92.47</td>
<td>74.70     77.91     78.85</td>
</tr>
<tr>
<td>Recall</td>
<td>38.48     43.16</td>
<td>37.99     55.06     56.83</td>
</tr>
<tr>
<td>F-Measure</td>
<td>54.65     58.65</td>
<td>50.36     64.52     66.05</td>
</tr>
<tr>
<td>NMI</td>
<td>0.4613    0.4752</td>
<td>0.4101    0.5049    0.6448</td>
</tr>
</tbody>
</table>

Table 5.1: Evaluation measures of the runs

5.4.2 Results of the challenges

The challenge involved other six groups of participants – i.e., ANU (the Australian National University), ITI (Informatics and Telematics Institute), EURECOM, VTT technical research center of Finland, QMUL (Queen Mary University of London), LIA (Laboratoire Informatique d’Avignon). In their approaches, all the groups made use a knowledge database. For example all the participant groups used Last.fm for
detecting and extracting information related to artists and venues for the second challenge. Only three groups (ITI, EURECOM and ANU) enhanced their methods using the visual features of the pictures. The results of all the groups are summarized in Table 5.2 for the first challenge and 5.3 for the second.

<table>
<thead>
<tr>
<th>ANU</th>
<th>ITI</th>
<th>EURECOM</th>
<th>LIA</th>
<th>NTNU</th>
<th>QMUL</th>
<th>VTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>84.86</td>
<td>90.58</td>
<td>97.75</td>
<td>7.49</td>
<td>92.47</td>
<td>76.81</td>
</tr>
<tr>
<td>Recall</td>
<td>52.54</td>
<td>67.58</td>
<td>42.38</td>
<td>15.62</td>
<td>43.16</td>
<td>62.11</td>
</tr>
<tr>
<td>F-Measure</td>
<td>64.90</td>
<td>77.40</td>
<td>59.13</td>
<td>10.13</td>
<td>58.85</td>
<td>68.68</td>
</tr>
<tr>
<td>NMI</td>
<td>0.237</td>
<td>0.618</td>
<td>0.247</td>
<td>0.026</td>
<td>0.475</td>
<td>0.414</td>
</tr>
</tbody>
</table>

Table 5.2: Results of the first challenge among the groups

<table>
<thead>
<tr>
<th>ANU</th>
<th>ITI</th>
<th>EURECOM</th>
<th>LIA</th>
<th>NTNU</th>
<th>QMUL</th>
<th>VTT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>70.79</td>
<td>54.31</td>
<td>70.99</td>
<td>14.37</td>
<td>78.85</td>
<td>42.14</td>
</tr>
<tr>
<td>Recall</td>
<td>43.90</td>
<td>80.61</td>
<td>67.01</td>
<td>37.99</td>
<td>56.83</td>
<td>27.13</td>
</tr>
<tr>
<td>F-Measure</td>
<td>50.44</td>
<td>64.90</td>
<td>68.95</td>
<td>12.44</td>
<td>66.05</td>
<td>33.01</td>
</tr>
<tr>
<td>NMI</td>
<td>0.448</td>
<td>0.385</td>
<td>0.617</td>
<td>–0.013</td>
<td>0.645</td>
<td>0.498</td>
</tr>
</tbody>
</table>

Table 5.3: Results of the second challenge among the groups

We globally obtained promising results, comparing with the results of the other groups. In particular, our approaches shown, in the their runs, high values of precision.

5.5 Conclusions

We have presented a system to extract events for the given two challenges. As described in this chapter, the best result in terms of precision was obtained in the first challenge by using only the tags for the categorization step, while the other evaluation measure were better when using all the textual metadata. In the second challenge the best result was obtained using the complete workflow of the algorithm, i.e. using refinement step, in particular using the entity names in the refinement query for each cluster. Our future experiments, especially for the first challenge, will include the use of the refinement step and semantic merge over the totality of the results (instead of applying it over groups of results coming from the query).
Part III

Exploring Geo-Temporal Distribution of Tags in Social Media
Chapter 6

Event-Related Image Search: Timestamped Query

Media-sharing applications – e.g., Flickr and Panoramio, contain a large amount of pictures related to real life events. Providing effective tools to retrieve event-related pictures within these media-sharing applications is an important but challenging task. One interesting aspect is to search pictures related to a specific event with a given annotated image. Most existing methods have focused on doing this by extracting visual features from the pictures. However, pictures in media-sharing applications increasingly come with location information, such as geotags. Therefore, we stress the importance of exploring the possibility to leverage on the geographical and temporal distribution of terms in a tag-based search process, within event-related image retrieval. Specifically, we propose extended query expansion models that exploit the information about the temporal neighborhoods among pictures in a collection, and leverage on the geo-temporal distribution of the candidate expansion terms to re-weight and expand the initial query. To evaluate our approach, we conduct extensive experiments on a dataset consisting of pictures from Flickr. The results from these experiments demonstrate the effectiveness of our method with respect to retrieval performance.

6.1 Introduction

Although recent developments and technological advances have helped the user to access public photos on the Web – e.g., through media-sharing applications, the amount of available information makes the access to these photos still a less straightforward task. To partly address this challenge, the development of event-related
image retrieval systems has been proposed [117]. An event-related image retrieval system is a retrieval system optimized to retrieve all pictures related to a specific event. Here, an event has a specific semantic meaning. Focusing on media-sharing applications, from Chapter 4, an event can be “something happening in a certain place at a certain time and tagged with a certain term”. So in an event-retrieval system, the intent of a user might be to retrieve resources related to a particular event, or to use a given tagged photo representing an event to retrieve other photos related to any similar events from a large image collection. Our main focus is on the latter.

Due to their characteristics, images in photo-sharing applications are particularly interesting. As previously specified, pictures in such applications are accompanied by contextual metadata, containing heterogeneous fields, such as camera-specific data – e.g., the Exchangeable Image File Format (EXIF) data, title, tags, description, temporal information – i.e., capture and upload time, and geolocation – i.e., geotags. In this work we study how we can exploit the above metadata to retrieve event-related pictures. In doing so, we aim at addressing the following challenges. First, not all pictures may contain reliable tags, description or title. They may either be missing or have no relation to the content of the picture. As a result, they may not contribute much in a retrieval task. Second, tags are unstructured, subjective and full of noise, thus worsening the retrieval performance. Third, many of the queries are short – i.e., pictures containing only few tags. Dealing with short queries is in itself a challenge. Fourth, a complete collection of images from photo-sharing applications is inherently large, and handling large datasets is in itself an important challenge.

In summary, the main goal of the work proposed in this chapter is to deal with the above challenges, focusing on situations where a user searches for pictures related to a specific event, each of which is represented by an image with a possibly small number of tags. We believe this area is still not mature, and that only few approaches are available – e.g., [168, 117, 92]. Further, within information retrieval, existing work has mainly been focused on applying temporal information in the retrieval models [73]. At the same time, the most related approaches, such as [168], are promising with respect to retrieval performance, but seems to be mainly based on using the visual features. We also considered using visual features as part of our approach. However, we early learned that performance (speed) could be a challenge with large data sets. Further, due to the characteristics of event-based pictures, also pointed out by Brenner and Izquierdo [20], we decided to mainly focus on using the metadata. As shown in the work presented in this chapter, we manage to get good retrieval performance, even without using visual features.

That is, we show that by mining and extracting the geo-profiles of terms from textual
tags, we can further improve the retrieval performance. To our best knowledge, this has not been explored in depth before. Existing work has mainly been concerned with point-of-interests (POI) extraction [145, 132] and trajectory mining [183]. With the constantly increasing number of geotagged pictures in – e.g., Flickr\(^1\), exploring this dimension is important. To this end, our main contributions are as follows. First, we conduct a study comparing the effectiveness of different retrieval models when using only the textual metadata in event-related image retrieval. As part of this, we thoroughly analyze how different combinations of textual fields affect the retrieval effectiveness, depending on the adopted retrieval model. Second, we propose a new weighting model for a query expansion step based temporal proximity in combination with existing term weighting and similarity models. Third, we develop a new extended model that also includes the mined spatial profile for terms in the textual tags. Our extensive evaluation shows that using both of our new models yields better retrieval performance than the baseline models, especially with short queries – i.e., pictures with only one to three tags.

This chapter is organized as follows. Section 6.2 gives an overview of the related work. Section 6.3 outlines some preliminary theory that our approach is based on, and defines the problems addressed in this chapter. Section 6.4 elaborates on our new weighting model for query expansion, accompanied by the query expansion models that we use as baselines for our experiments in Section 6.5. Section 6.6 presents the result from these experiments. Finally, in Section 6.7 we conclude the chapter and outline our future work.

### 6.2 Relation to Other Work

In the past decades, detection of events from textual document streams and databases has been extensively treated in the literature [3, 19]. However, although mining and retrieving pictures related to real-life events is an active field, it is still a less mature research domain [144, 55, 168]. Most existing related approaches have been aimed at extracting events from different types of datasets. To our best 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\(^2\) [117], where the main objective was to propose event retrieval systems for Flickr pictures.

Query expansion (QE) has been proven to increase retrieval effectiveness, where

\(^1\)In 2009, more than 3.3% (approx. 100 Million pictures) were geotagged.

See also [http://goo.gl/fvjPg](http://goo.gl/fvjPg)

an often applied approach is so-called Pseudo Relevance Feedback [188, 102, 142]. The use of temporal information in Information Retrieval (IR), has been previously widely investigated both for ranking models [88] and query expansion [33]. Approaches incorporating geographical context in query expansion has, on the other hand, been mainly proposed in the field of Geographical Information Retrieval (GIR) systems [21, 49, 121, 128]. In particular in [49], the authors propose an expansion process by deriving the spatial query footprint from SPIRIT\(^3\) ontologies, while in [21], the term suggestion is supported by Wordnet\(^4\). The main difference to our work is that in both [49] and [21], the query expansion process uses similarity derived from ontologies, where as, in our work we measure the geographical co-occurrence using the dynamic context of social media resources such as Flickr. This also allows us to use free text search rather than relying on queries with specific query structure, such as the triplet \(<\text{what}, \text{relation}, \text{where}>\), often used in GIR [21]. Note that tag suggestion can be related to the topic of query expansion. For example in the work in [159] an [111], the authors proposed tag recommendation methods for Flickr pictures. The limitation of these approaches is that the input query image must be necessarily geotagged. However, with our approach we only require the tag-based search to be performed with timestamped textual queries.

### 6.3 Problem Definition

The main focus of our work is on a tag-based search of event-related pictures from a photo collection. Here, we assume a query to be a set of tags from a picture – e.g., a Flickr picture tagged with textual information, including Title, Tag and Description, as well as a timestamp specifying when the picture was taken.

So, consider a collection of Flickr images as our target dataset \(\mathcal{D}\), where each image \(\mathcal{I}\) comes with metadata consisting of information about when the picture was taken and the textual annotations. Then, each image \(\mathcal{I} \in \mathcal{D}\) can be represented as \(\mathcal{I} = \{\mathcal{T}, d_t\}\), where \(\mathcal{T} = \{\text{Title}, \text{Description}, \text{Tags}\}\) denotes the set of textual annotations for \(\mathcal{I}\), and \(d_t\) is the timestamp for when the photo was taken. With the aforementioned challenges in mind, we want to investigate approaches to deal with the situation in which a user wants to retrieve a set of pictures representing a specific event, given a picture representing the same event. Formally, if we let \(\hat{\mathcal{S}}\) be a set of pictures representing the target event-related to the user query intention, and \(\mathcal{I}_q \in \hat{\mathcal{S}}\) denote our query image, then our problem is to retrieve all \(\mathcal{I} \in \hat{\mathcal{S}}\) representing the same event as \(\mathcal{I}_q\).

\(^3\)http://www.geo-spirit.org/
\(^4\)http://wordnet.princeton.edu/
As part of the solutions to our problem, we will answer the following research questions: First, how do different tag fields of a picture from a media-sharing application such as Flickr affect the retrieval effectiveness? Second, can a query expansion step be useful in retrieving event-related pictures, if we have a query consisting only of the metadata for a single picture? Third, which temporal and spatial features can be useful to improve the search effectiveness in retrieving event-related pictures? Forth, can we still improve the retrieval effectiveness when applying queries with small number of tags?

As we explain in Section 6.4, we aim particularly at exploring the temporal proximity between term distributions and considering the spatial profile of tag terms in retrieving event-related pictures. Further, to partly answer the above questions, in our evaluation we will first perform a set of baseline experiments in which we explore the effectiveness of different retrieval and query expansion models. Then, we will evaluate the retrieval effectiveness of our query expansion model based on temporal and spatiotemporal re-ranking of the retrieved list.

6.4 Query Expansion for Event Retrieval

Query expansion (QE) is a post-processing step in retrieval systems, aiming at ensuring good retrieval performance when the query is too short, poor and does not contain all the terms, and therefore does not sufficiently reflect the user’s search intent [9]. The effectiveness of QE has been proved in many works [180, 62, 102]. One of the most used QE approaches is pseudo-relevance feedback [9]. The main idea is to assume that a top-k ranked list of retrieved documents are relevant to a specific query. Then, we perform QE by extracting terms from these documents, and use them to re-weight and extend the terms in the original query. Depending on the method being used, the choice of the terms can be done by comparing the distributions of terms in the retrieved (or feedback) documents and the entire collection. Note that since we want to tackle the challenges connected to searching event-related pictures using metadata – assuming timestamped pictures with small number of tags, it is necessary to improve and adapt existing query expansion techniques. In the following, we elaborate on how we do this after giving an overview of the baseline QE approaches.

6.4.1 Baseline Query Expansion Approaches

Generally speaking, a query expansion approach is a two step approach consisting of (1) choosing the terms to be used in the expansion, and (2) assigning the
weight to the chosen terms. Focusing on (1), there are several approaches that have been suggested. Among these, we have specifically considered two methods that has been proven to be very effective: the Kullback-Liebler (KL) divergence-based approach [26] and the Divergence from Randomness (DFR) model [5]. With the KL divergence approach, the idea is to analyze the term distributions, and maximize the divergence between the distribution of terms from the top-k retrieved documents and the distribution of terms over the entire collection [26]. The terms chosen for the query expansion are those contributing to the highest divergence – i.e., the highest KL-score [26]. This means that expansion terms with low probability in the entire collection and high probability on the retrieved top-k documents are given more weights than other terms. The following equation is used to calculate the KL-score for a given term \( t \) in the feedback (top-k) documents [26]:

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

where \( P_{Rel}(t) \) and \( P_{Coll}(t) \) are the probability that \( t \) appears in the top-k documents and in the collection, respectively. Here, \( P_{Rel}(t) \) can be estimated by the normalized term frequency of \( t \) in the top-k documents, whereas \( P_{Coll}(t) \) can be computed as the normalized frequency of \( t \) in the entire collection. With the DFR model, on the other hand, the idea is to weight the expansion terms by calculating the divergence between the distribution of terms in the feedback documents (the top-k documents) and a random distribution [5]. In our work, we have chosen to implement this method based on the Bose-Einstein statistics (Bo1), which has been shown to be one of the most effective approaches. Bo1 is computed as follows [5]:

\[
Bo1(t) = t_{feedback} \log \left( \frac{1 + P_n(t)}{P_n(t)} \right) + \log [1 + P_n(t)],
\]

where \( t_{feedback} \) is the frequency of term \( t \) in the feedback documents, and \( P_n(t) = F/N \) is the ratio between the frequency \( F \) of \( t \) in the entire collection and \( N \) the size of the data set. After the expansion terms have been selected using one of the approaches above, we can proceed to step (2) – i.e, re-weighting the terms in the query. One of the classical approach to re-weight query terms is the Rocchio’s algorithm [142]. In particular we use the Rocchio’s Beta equation [123] as follows:

\[
\hat{w}(t_q) = \frac{tf_{q,t}}{\max tf_q} + \beta \frac{w(t_q)}{\max w},
\]

where \( \hat{w}(t_q) \) is the new weight of a term \( t_q \) of the query, \( w(t_q) \) is the weight from the expansion model – i.e., \( KL(t_q) \) or \( Bo1(t_q) \), \( \max w \) is the maximum weight from the expanded weight model, \( \max tf_q \) is the maximum term frequency in the query and \( tf_{q,t} \) is the frequency of the term in the query.
6.4.2 Extended Query Expansion Models for Event Retrieval

In this section we propose a set of methods to extend the above baseline models. Our main goal is to exploit the temporal and geographical information encapsulated in the picture tags. Previous approaches have focused on investigating the application of the temporal information in pseudo relevance feedback approaches. For example, the approaches by Keikha et al. [77], and Efron and Golovchinsky [44] proposed methods to incorporate time into the relevance model by Lavrenko and Croft [84]. In contrast to this, our objective is to use the characteristics of an event, in combination with the temporal proximity of the term distribution as features in the term selection process for a query expansion framework. We assume that all pictures in our collection contains a temporal annotation identifying when the picture was taken – i.e., a timestamp. Further we hypothesize that pictures related to the same event have some temporal proximity or temporal closeness. This means that the more temporally close to the query the retrieved pictures are, the more likely that they are related to the same event. Such a property is useful in a query expansion framework, since we can use the temporal information to decide the term weights. For example, we can give higher weights to terms having higher probability to appear in a document and being temporally close to the query. With this in mind, in the following we propose a query expansion model to improve the retrieval of events.

**Temporal-Proximity Re-Ranking**

As a first improvement, we explore the effectiveness of using a ranking function that considers both the textual similarity and the temporal proximity of the document, in the query expansion process. The idea is to push documents with higher temporal proximity up in the top-k feedback documents. Note, however, that the temporal similarity and the textual similarity are not two unified measures. Therefore, the scores assigned by performing two queries, one with textual query and another with the temporal data, are not straightforward to merge by a score-based ranked list fusion. For this reason, we merge the two ranked lists by adopting \( rCombMNZ \) [86], which is the ranked-based version of \( CombMNZ \) [155], given by

\[
\text{score}^{R_i} = h(d, R) \sum_{R_i \in [R_1, R_2]} g^{R_i}(d),
\]

(6.4)

where \( d \) is a document of a ranked list, \( R_1 \) and \( R_2 \) are the two ranked lists and \( h(d, R) \) is the rank hits representing the number of ranking lists in which the document \( d \) is present. Further, \( g^{R_i}(d) \) denotes the normalized ranking score of the document \( d \) in the ranked list \( R_i \).

**Temporal-Proximity Aware KL Divergence**

As a second improvement, we actively use the assumption about temporal proximity, mentioned before. In both of the presented baseline query expansion models, the
core premise is that a query expansion word should be more common in the feedback documents and less common in the whole collection. This means that we have a high divergence between the distribution of the candidate term expansion in the feedback document set, and the distribution of the same term in the whole collection. Hence, our intuition is the following: the distribution of a good candidate expansion term should commonly co-occur as much as possible in documents that are temporally close to the query picture and less common in the whole collection. This is the same as having a high divergence between the distribution of the co-occurrence of the candidate expansion terms and the query terms in the set of temporal neighbors pictures, and the distribution in the whole collection. The idea is that in addition to the original KL-divergence computation, our weighting process also considers the divergence of the term distributions within a time slice \( L \), centered in the timestamp of the query image, and the co-occurrence with the query terms within the same time slice. Now, let \( \theta^L_{[t,t_i]} \) be the distribution of the co-occurrence between the candidate expansion term \( t \) and the query terms \( t_i \in Q \) within the set of temporal neighbors, and \( \theta^{Col}_L(t,t_i) \) denote the distribution of the co-occurrence terms in the whole collection.

Then, our temporal-aware KL score can be computed as follows:

\[
KL^L(Q,t) = \sum_{t_i \in Q} KL(\theta^L_{[t,t_i]} || \theta^{Col}_L(t,t_i))
\]

\[
= \sum_{t_i \in Q} P_L([t|t_i]) \log \left[ \frac{P_L(t|t_i)}{P^{Col}_L(t|t_i)} \right].
\]

In this re-weighting process, the new weight of a candidate expansion term \( t \) is the sum of the divergence between \( \theta^L_{[t,t_i]} \) and \( \theta^{Col}_L(t,t_i) \), for all the \( t_i \in Q \). In other words, a candidate expansion term gets a higher weight if the divergence between these two distributions \( \theta^L_{[t,t_i]} \) and \( \theta^{Col}_L(t,t_i) \) is high. Further, \( P_L(t|t_i) \) is the co-occurrence probability of the terms \( t \) and \( t_i \) within a time interval \( L \), and \( P^{Col}_L(t|t_i) \) is the co-occurrence probability of the terms \( t \) and \( t_i \) within the whole collection. We evaluate the co-occurrence probability as proposed in [158] by adding a normalization factor:

\[
P_L(t|t_i) = \left[ \frac{n^L_d(t,t_i)}{n^L_d(t)+n^L_d(t_i)} \right], \text{ and}
\]

\[
P^{Col}_L(t|t_i) = \left[ \frac{n^{Col}_d(t,t_i)}{n^{Col}_d(t)+n^{Col}_d(t_i)} \right],
\]

where \( D \) is the whole dataset and \( D_L \subset D \) is a subset of \( D \) composed by documents having timestamp within the time interval \( L \). This means that \( n^L_d(t,t_i) \) and \( n^{Col}_d(t,t_i) \) are the number of documents in the set \( D_L \) and \( D \), respectively, in which the terms \( t \) and \( t_i \) co-occur. Similarly, \( n^L_d(t) \) and \( n^{Col}_d(t) \) are the number of documents in the set \( D_L \) and \( D \), respectively that are tagged with the term \( t \).
Example: To explain the motivation behind Equation 6.6, consider the tag scores in Table 6.1. This table shows the results of two re-weighting processes: (1) by using the baseline KL divergence in Section 6.4.1, and (2) by using the temporal KL in Equation 6.6. Here, our query was a picture with the tags \{atmedia, london, ajax\} and the timestamp (27.09.2007), referring to a periodic conference event, "atmedia", in 2007. As shown in Table 6.1, our dataset at least contains pictures from the 2008 and 2009 conferences.

In this example, we can make the following interesting observations. First, with the baseline approach, although several tags may refer to the same periodic event – e.g. the tag atmedia2008 and atmedia2009, different times may lead to different scores. Second, using our temporal KL divergence approach, generic event-related terms in the user query Q, such as event, conference and session, get higher scores than with the baseline approach. This is because the distribution of the co-occurrences of such terms with the query terms have a higher divergence in the set of temporal neighbors, compared to the divergence of the same distribution in the whole collection. To further illustrate the usefulness of applying the temporal infor-
Table 6.1: The scores of the query expansion terms after baseline KL divergence ($KL(t)$), and temporal KL divergence ($KL^c(Q,t)$).

<table>
<thead>
<tr>
<th>Term</th>
<th>$KL(t)$</th>
<th>$KL^c(Q,t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>atmedia</td>
<td>1.400</td>
<td>1.316</td>
</tr>
<tr>
<td>london</td>
<td>1.168</td>
<td>1.266</td>
</tr>
<tr>
<td>ajax</td>
<td>1.108</td>
<td>1.135</td>
</tr>
<tr>
<td>atmedia2009</td>
<td>0.270</td>
<td>0.400</td>
</tr>
<tr>
<td>atmediaajax</td>
<td>0.089</td>
<td>0.244</td>
</tr>
<tr>
<td>javascript</td>
<td>0.077</td>
<td>0.182</td>
</tr>
<tr>
<td>atmedia09</td>
<td>0.055</td>
<td>0.146</td>
</tr>
<tr>
<td>media</td>
<td>0.050</td>
<td>0.130</td>
</tr>
<tr>
<td>web</td>
<td>0.047</td>
<td>0.097</td>
</tr>
<tr>
<td>conference</td>
<td>0.040</td>
<td>0.048</td>
</tr>
<tr>
<td>atmedia2008</td>
<td>0.030</td>
<td>0.046</td>
</tr>
<tr>
<td>event</td>
<td>0.025</td>
<td>0.033</td>
</tr>
<tr>
<td>presentation</td>
<td>0.020</td>
<td>0.031</td>
</tr>
<tr>
<td>brendaneich</td>
<td>0.015</td>
<td>0.021</td>
</tr>
<tr>
<td>session</td>
<td>0.014</td>
<td>0.021</td>
</tr>
<tr>
<td>johnresig</td>
<td>0.013</td>
<td>0.015</td>
</tr>
<tr>
<td>christianheilmann</td>
<td>0.012</td>
<td>0.014</td>
</tr>
<tr>
<td>patrickgriffiths</td>
<td>0.012</td>
<td>0.009</td>
</tr>
</tbody>
</table>

Combining the KL Divergences

To include the influences of both scores in the calculation of the final expansion weight, the last two models can be mixed in a linear combination, given by

$$KLT(Q,t) = \gamma KL(t) + (1 - \gamma)KL^c(Q,t), \quad (6.9)$$

where $\gamma$ is factor used to determine the amount of influence each score has on the final weight. In our experiment, we will analyze the retrieval effectiveness as function of the values of $\gamma$ on the weighting step, in the query expansion process. This gives us also the possibility to evaluate the impact of the proposed temporal weighting model.

In the table, Figure 6.1 shows how the temporal distributions of two tags conference and atmedia, and their co-occurrences look like within a given time interval. Also, the results of our experiments in Section 8.5 demonstrate that our observation also apply to most cases.
6.4.3 Exploring Term Spatial Distribution

As explained in our hypothesis, pictures related to the same event tend to appear in a limited geographical area. In this approach we mainly consider query pictures that are not geotagged. There are two main reasons for this. First, we believe the problem would be less challenging to solve when having both the temporal and geographical information available. Second, we aim at making our approach as generic as possible, and thus enabling it for media-sharing applications and social media in general. For example, in the Flickr dataset only 3.3% of the pictures are geotagged. As a conclusion, although the probability to have a geotagged picture is low, the portion of pictures that are geotags can still be useful to extract geographical profile of the terms.

With this in mind, we propose a method to find a good expansion term \( t \), given a set of query terms \( Q = \{t_i\}_{i} \). Including the geographical dimension, a good expansion term is a term related to the same event of the query picture. In particular it is a term that commonly co-occur in documents that are temporally close to the query picture and in a geographic delimited area, and less common in the whole temporal timeline in the same delimited area. To define a realistic problem, the query picture is not geotagged.

The method presented is based on the discretization of the world map. We first divide the world map in \( M \) tiles \( \Theta = \{\mathcal{T}_k\}_{k=1..M} \) of size one degree as proposed by Zhang et al. [189]. This means that the tiles does not have the same size. This is because, on the world map, the size corresponding to one degree varies depending on the latitude values; spanning from 0 Km on the poles, to 100 Km close to the equator. This approximation is suitable to use since most of the highly populated areas are closer to the equator than the poles.

To include the spatial dimension in the candidate expansion term score, we use a similar hypothesis to the one proposed in Section 6.4.2 as a starting point. This means that a good expansion term \( t \) is the one for which there is a high divergence between the distribution of the pictures tagged with the query term and the expansion term in a temporal time slice \( \mathcal{L} \) and a tile \( \mathcal{T}_k \), and the distribution of the terms in the same geographical tile \( \mathcal{T}_k \) but covering the whole timeline.

Formally the new divergence are computed using KL-divergence as follow:

\[
KL^L_{\mathcal{T}_k}(Q, t) = \sum_{t_i \in Q} KL(\theta^L_{|t_i, \mathcal{T}_k}|\theta^{Coll}_{|t_i, \mathcal{T}_k})
\]

\[= \sum_{t_i \in Q} P_L(t|t_i, \mathcal{T}_k) \log \left[ \frac{P_L(t|t_i, \mathcal{T}_k)}{P^{Coll}(t|t_i, \mathcal{T}_k)} \right]. \tag{6.11}\]
Here, $P_L(t|t_i, T_k)$ is the co-occurrence probability of the query term $t_i$ and expansion term $t$, within a time interval $L$ and a geographical tile $T_k$. Similarly, $P_{Col}(t|t_i, T_k)$, is the same probability without the temporal restriction. We approximate these probability as follows:

$$P_{Col}^G(t|t_i, T_k) = \left[ \frac{n_{Col}^G(t,t_i|T_k)}{n_{Col}^G(t|T_k) + n_{Col}^G(t_i|T_k)} \right]$$

(6.12)

$$P_L^G(t|t_i, T_k) = \left[ \frac{n_L^G(t,t_i|T_k)}{n_L^G(t|T_k) + n_L^G(t_i|T_k)} \right]$$

(6.13)

We calculate the pair of probabilities $P_{Col}^G(t|t_i, T_k)$ and $P_L^G(t|t_i, T_k)$ for each tile $T_k \in \Theta$. We calculate the divergence between the two distribution values, tile by tile. We consider the maximum as the final score. In order to include the influence of KLT, we mixed the models in a linear combination, given by

$$KLST(Q,t) = \sigma KLT(Q,t) + (1 - \sigma) \max_{T_k} \{ KL_{T_k}^L(Q,t) \}$$

(6.14)

### 6.4.4 Scalability of the Method

Recall that the purpose of our work is to improve the tag-based search effectiveness of event-related resources, such as Flickr pictures, by improving the keyword-based ranking models in IR. In our approach, the images are indexed based on their textual metadata (the tags), using inverted index structure. It is a data structure that efficiently store and retrieve textual resources, and has been proven scalable [6]. As for our framework, the temporal and spatial dimensions are included in the ranking model, and our query expansion method does not need extra data structure. Thus, the only bottleneck might be the increased size of queries. However, as we mentioned before, we assume that the query size is normally small. Therefore, this would not be an issue.

Nevertheless, our expansion algorithm is depicted in Algorithm 2. To further understand the scalability of our approach, let us analyze the computation cost of this algorithm. As can be observed, to compute the final score, the algorithm requires $N = |E| \times |Q| \times |\Theta|$ steps, where $|E|$ is the number of expansion terms, $|Q|$ is the size of the query and $|\Theta|$ denotes the number of tiles. Since $|\Theta|$ is a finite number and that not all tiles contain images, plus $|Q|$ is normally small, it is safe to assume that our algorithm has a complexity of $O(|E|)$.

In general situations where the above sizes are unlimited, we can parallelise the core of the algorithm – i.e., Step 6 to Step 8 in Algorithm 2. Moreover, computing
Algorithm 2 Pseudo code of the QE procedure that incorporates geo-temporal dimensions.

1: $\mathcal{L} \leftarrow$ time interval centred in the query timestamp
2: **Query** $Q$ by using ranking model $r$ and get the $D$ set 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:  for $t_i$ in $Q$ do
6:    for $T_k$ in $\Theta$ do
7:      Calculate $KLT_{T_k}(t_i, e_j)$
8:    end for
9:  Calculate $KLST(t_i, e_j)$
10: end for
11: Calculate $\sum_{t_i \in Q} KLST(t_i, e_j)$
12: end for
13: **Rank** $e_j \in \mathcal{E}$ terms according to $KLST(Q, e_j) \rightarrow \mathcal{E}_{Rank}$
14: **Re-build** $Q$ with the top-k terms from $\mathcal{E}_{Rank} \rightarrow \hat{Q}$
15: **Query** $\hat{Q}$ by using ranking model $r$

Equation 6.10 is done with a query limited in a spatial area (the tile). During the computation, this area is fixed for any queries. In such a case, scalability would not cause any problem.

As a final note, to perform our experiments, we indexed and run our queries using Terrier$^5$ for the text search and Solr for the spatial search, both providing features for searching and storing Web-scale indexes. Further, we defined three random test queries with one keyword, two keywords and three keywords. Then, we measured the response time as function of the size of the dataset – i.e., the number of indexed documents. We performed the experiments on an Intel i7-950 Processor, with 24Gb RAM and 1Tb Hard Disk. Figure 6.2 summarizes the results of our experiments, showing the average response time of the baseline QE method and the average response time of our QE approach. As can be derived from these results, even though the size grew, the execution times did not follow the increase of the dataset size. Note that the code written to perform the experiments was not optimized, and thus this lack of optimization might affect the response time, in general. More specifically, we did not perform any parallelization of the queries in Step 6-8 in Algorithm 2. We did not optimize Solr neither, but used standard tuning values. Finally, we did not warm up the cache of the Solr system before each experiment – i.e., the cache was empty at each query processing.

$^5$See [http://www.terrier.org](http://www.terrier.org)
Figure 6.2: Response time for our QE executions with the random test queries as function of the dataset size.

6.5 Experimental Setup

6.5.1 Dataset

To evaluate our method, we use the Upcoming dataset [14] as the ground-truth for our experiments. This dataset consists of 270,425 pictures from Flickr, taken between 1st of January 2006 and 31st December 2008, each of which belongs to a specific event from the Upcoming event database. 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. Further, since the size of this dataset is relatively small for our purpose, and due to the lack of other datasets that are very large, we decided to build an additional dataset by merging the Upcoming dataset with other pictures gathered from Flickr covering a time period from 01.01.2006 to 31.12.2010 and without spatial restrictions. Our final dataset now consists of 88,257,485 pictures, of which 18,861,585 pictures are without any tags and around 23.5% with 1 to 3 tags (see also Figure 6.3 for more information about the distribution of the number of tags). This further illustrates the necessity of supporting short queries, as mentioned in Section 8.1. Also, this shows that both the ground-truth and the final dataset contain sufficiently enough portions of short queries.

Before performing our experiments, we first indexed all image tags using Terrier. As part of the dataset preparation we run a preprocessing step consisting of tokenization—i.e., UTFTokenization based on whitespace and punctuation marks, and English stopword removal. Then, we randomly selected set of pictures from each event cluster in the Upcoming dataset and use these as queries.
6.5.2 Evaluation Methodology

To assess the effectiveness of our approach, we compare our models with existing models, which also serve as baseline for our evaluation. Our baseline models are the Vector Space Model (TFIDF) [9], Okapi BM25 (BM25) [140], Hiemstra Language Model (LM) weighting model [65] and KL divergence retrieval model (KLDM) [82]. For both BM25 and LM, we use the default parameter values – i.e., for BM25 we set $k_1 = 1.2$, $k_3 = 8$ and $b = 0.75$, and for LM is $c = 0.15$.

To evaluate the retrieval performance, we use standard in information retrieval evaluation metrics, including the Mean Average Precision (MAP) and R-Precision (RP) [9]. 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. Therefore, any stated improvements in this chapter are all statistically significant, unless otherwise specified.

6.5.3 Considerations Related to Query Expansion

Studying our dataset, we observed that more than one picture related to the same event have been annotated with the same set of tags by the same user. This is because many users in Flickr often copy and paste the same set of tags for pictures related to the same events or same group of pictures. To illustrate this, Figure 6.4 shows the difference between the number of picture retrieved and the number of...
unique pictures in the retrieved set, using our query set presented above and with a BM25 retrieval model.

This histogram shows that a set of retrieved documents generally contains a high percentage (around 80% in all the cases) of pictures with duplicated set of tags. This observation is useful when performing a query expansion on the type of dataset as ours. Further, when extracting candidate expansion terms from the top-K retrieved documents, it can happen that a high number of duplicates of tag sets are in the documents (pictures) within the top-K positions. This would reduce the space of candidate expansion terms. To avoid this problem, we decide to remove the duplicates from the retrieved document set during the process of selecting the top-K retrieved pictures. So in our experiments, the number of pictures in the top-k retrieved set used to select the candidate expansion terms is the number after removing the duplicates.

Finally, to handle noisy and non-informative tags, we first filter candidate expansion terms from the whole dataset that do not comply with $tf < 100$. Then, we remove candidate terms that do not match $tf_{T_k} < 50$, where $tf_{T_k}$ is the term frequency of a term extracted from images taken within the geographical tile $T_k$.

### 6.6 Results

Aiming at answering our research questions in Section 6.3, we analyze the effectiveness of each textual field in the pictures to find out which of the fields contributes to the best retrieval performance. Thereafter, we perform different sets of experiments
to study the effectiveness of our proposed query expansion model with respect to different parameter values.

6.6.1 Field Effectiveness

Our first experiment aims at exploring the effectiveness of using Flickr images as queries. To assess this effectiveness and analyze the role of the fields in the metadata, we use different combinations of the textual metadata as queries and document representation. Specifically, we evaluate how Title, Tag and their combination affect the retrieval effectiveness. To do this, we first use Title only as a document, then Tag only, and finally Description only. Thereafter, we test different combinations of these fields as follows: Title and Tag; and Title, Tag and Description.

Note that the efforts of the TREC community on retrieval of structured and unstructured documents – i.e., the INEX benchmarking for XML information retrieval, and the field-based retrieval models such as BM25F [137] can seem related to this part of our work. However, because the focuses of these are more on full text contents, they are beyond the scope of this work.

The set of queries is formed by randomly selecting 1 picture from each event cluster in the Upcoming Dataset. Here, we only consider event clusters containing more than 500 pictures from a total of 50 clusters. Thus, the total number of queries is 50 for each sample. This random sampling is repeated 5 times to obtain five sets of 50 queries, which means that the total number of queries submitted are 250.

Table 6.2, Table 6.3 and Table 6.4 summarize the results from the experiments for our the retrieval effectiveness analyses. Here $TAG_{TAG}$ means that we use the tag field in both the indexing and the query, whereas $TAG_{TIT}$ means we apply tag $(TAG)$ in the indexing but title $(TIT)$ in the query, and so on. $TT$ stands for tags and title combination, while DES is the description field. The numbers 1, 2, 3 and four in superscript in the tables indicate the statistical significance improvements on the dataset indexed with $TAG$ field, TIT field, DES field and TT fields, respectively.

<table>
<thead>
<tr>
<th>Comb</th>
<th>TFIDF</th>
<th></th>
<th>BM25</th>
<th></th>
<th>LM</th>
<th></th>
<th>KLDM</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAP</td>
<td>RP</td>
<td>MAP</td>
<td>RP</td>
<td>MAP</td>
<td>RP</td>
<td>MAP</td>
<td>RP</td>
</tr>
<tr>
<td>$TIT_{TAG}$</td>
<td>.498</td>
<td>.502</td>
<td>.500$^{234}$</td>
<td>.506$^{234}$</td>
<td>.484$^{234}$</td>
<td>.492$^{234}$</td>
<td>.0503$^{234}$</td>
<td>.510$^{234}$</td>
</tr>
<tr>
<td>$TIT_{TIT}$</td>
<td>.350</td>
<td>.358</td>
<td>.324</td>
<td>.332</td>
<td>.357</td>
<td>.364</td>
<td>.353</td>
<td>.360</td>
</tr>
<tr>
<td>$TIT_{DES}$</td>
<td>.550$^{124}$</td>
<td>.559$^{124}$</td>
<td>.459</td>
<td>.467</td>
<td>.460</td>
<td>.468</td>
<td>.460</td>
<td>.468</td>
</tr>
<tr>
<td>$TIT_{TT}$</td>
<td>.113</td>
<td>.129</td>
<td>.106</td>
<td>.124</td>
<td>.127</td>
<td>.140</td>
<td>.130</td>
<td>.147</td>
</tr>
</tbody>
</table>

Table 6.2: MAP and RP by querying using the Title field
With the results in these three tables, we can make the following observations. First as shown in Table 6.2, querying using the title resulted in the lowest MAP and R-precision (RP) values compared to querying with the title and the tags. Further, looking at the best results in each table, for each retrieval model, the most representative field for each picture was the Tag field, with which the MAP and the RP values were the highest. Finally in Table 6.3 and Table 6.4 we can see that in all the cases, the highest MAP and RP values were obtained when the same fields were used both to represent the documents/images and to generate the set of queries. In summary, since these results are conclusive, we can safely base our experiments to test our query expansion (QE) step using the combination \( TAG_{TAG} \).

### 6.6.2 Short Queries vs Long Queries

In this section we compare the retrieval effectiveness of using query pictures with less than three tags and query pictures with more than four tags. To do this, we randomly select 40 query pictures with less than three tags and 40 query pictures with more than four tags. To make the experiment more realistic, we consider only event clusters containing more than 100 pictures. This is because a small number of
users normally contribute to small clusters. Thus, there would be a high probability that a high percentage of the pictures would be annotated with the same tags.

To perform this experiment, as well as executing the standard models, we also applied the Query Expansion (QE) models described in the previous section. Specifically, we used the the Rocchio’s framework weighting model, with both the Kullback-Leibler divergence model (KL), and the Bose-Einstein weighting scheme (Bo1) to choose the expansion terms. For each QE run, we used the default values – i.e., setting $\beta = 0.4$ and choosing the first $n$ terms of the top-$K$ documents for the Rocchio’s Beta weighting model. The values of $K$ – i.e., the number of pseudo relevant documents, were chosen from $\{30, 60, 90\}$, and $n$ – i.e., the number of selected terms, from $\{8, 18\}$.

![Figure 6.5: MAP values with respect to using different retrieval models](image)

<table>
<thead>
<tr>
<th></th>
<th>Bo1 Long</th>
<th>Bo1 Short</th>
<th>KL Long</th>
<th>KL Short</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFIDF</td>
<td>2.75%</td>
<td>5.79%</td>
<td>2.75%</td>
<td>5.78%</td>
</tr>
<tr>
<td>BM25</td>
<td>2.62%</td>
<td>5.71%</td>
<td>2.54%</td>
<td>5.83%</td>
</tr>
<tr>
<td>LM</td>
<td>-0.33%</td>
<td>3.98%</td>
<td>-0.32%</td>
<td>3.98%</td>
</tr>
</tbody>
</table>

Table 6.5: Short vs. long queries: Percentage improvements using the query expansion model compared to the standard retrieval model, in terms of MAP values.

Figure 6.5 and Table 6.5 present the results of our comparisons of the effects of using short and long queries. Specifically, in Figure 6.5 we focus on comparing the effects of short and long queries on the retrieval effectiveness when using only standard IR models. In Table 6.5, on the other hand, we compare the impacts of the query lengths when applying the two different query expansion models, Bo1 and KL. Here,
we summarize the percentage improvements from standard IR models to applying the query expansion models.

As Figure 6.5 shows, with all standard IR models, we obtained the highest MAP and R-precision (RP) values with long queries. In contrast to this, as shown in Table 6.5, when applying the query expansion step, we generally get the best results with the short queries. More specifically, apart from the Language Model (LM), where long queries resulted in decreased MAP values, applying the query expansion step yielded two times higher improvements with short queries than using long queries.

As a conclusion, if we only use standard retrieval models, we get the best results with long queries. The reasons for this is that (1) the use of a higher number of tags make the query more effective, and (2) many users usually annotate groups of pictures with high number of tags. Because we extract the expansion terms from a list of Top-\(K\) documents, thus making most of the query terms either an excess or more important, short queries with the expansion steps give the best results. For this reason, we focus on improving the query expansion models based on short queries.

### 6.6.3 Evaluating the Extended QE Models

In this experiment, we evaluated the approaches proposed in Section 6.4.2. As in the previous experiment, we first randomly selected 100 queries from the event clusters, containing more than 100 pictures. Then we selected pictures with less than three tags.

In the first set of experiments, we compared the results obtained by performing the retrieval process followed first by the standard KL divergence for query expansion (KL), and thereafter by the proposed proximity based temporal KL (KLT). In the second set of experiments, we tried the combination of the two proposed methods – i.e., selecting the expansion query terms by considering the pseudo-relevant top-\(K\) documents and weighting the terms extracted applying KL and KLT, using the linear combination in Equation 6.9. In the third experiment, we compared the previous models with the one based on spatial distribution of terms (KLST). In the fourth experiment set, to assess the effectiveness of KLT, we compared the effectiveness of KL and KLT, when doing a re-ranking step as explained in Section 6.4.2, with Equation 6.4 being either applied or not applied. Here, the QE was performed on pseudo-relevant pictures, still using Bo1 and KL in the Rocchio’s Beta framework (RB), with the same default values of \(\beta\).

Now, to perform a complete set of experiments, we considered different values of the following parameters. First, as query expansion parameters, we varied the value of
$K$ such that $K \in \{30, 60, 90, 120\}$ and the values of $n$ such that $n \in \{8, 18\}$. Second, as a parameter for KLT, we varied the time slice $L$ in the following set: \{1 day, 3 days, 7 days\}. Third, for the re-ranking step, we varied the $R$ values – i.e., number of top-$R$ documents to re-rank, in the set \{1000, 2000, 3000, 4000\}.

In addition to the above models, we also implemented the Mixture Model \[188\] and the Relevance Model \[84\]. However, the results were comparable to the KL and Bo1 query expansion models. Thus, due to the space constraints, we did not include them in this chapter.

The Impact of $\gamma$ on Mixed KL

With this set of experiments, we tested the impact of the parameter $\gamma$ in Equation 6.9 used to linearly combine the KLT and the standard KL divergences. We varied its values from 0 to 1 such that $\gamma \in \{0, 0.25, 0.5, 0.75, 1\}$, where 0 means that we only have the contribution of KLT and to 1 we only have the contribution of KL. We repeated this experiment for the six combinations of the number of query expansion terms $n$, and the number of top-$K$ documents considered in the query expansion process.

Figure 6.6 shows the impact of varying the values of $\gamma$ on the MAP values. As can be seen in this figure, for all the combinations of $K$ and $n$ values, the MAP values decreased when we increased the $\gamma$ value. This means that mixing both of the contributions was not very effective with respect to retrieval performance, but the most important contribution came from our KLT divergence.

KL vs KLT

To further assess the performance of our KLT approach, we compared it with the baseline approach, using the linear combination in Equation 6.9, with $\gamma = 0$.

First, we compared KL and KLT without any re-ranking step. The result from this experiment is summarized in Figure 6.7, showing comparison between the retrieval effectivenesses of our QE models and the baseline models. As can be observed, by using BM25 and TFIDF retrieval models in the initial retrieval step, our KLT outperforms KL, with all combinations of $K$ and $n$. With LM and $n = 30$, the KLT also outperforms the baseline model. With $n = 60$, the KLT still outperforms KL but in this case the query expansion process is not very effective. Overall, we can conclude that our query expansion (QE) models are better than the baseline QE model, and that all presented improvements are statistically significant at 95% confidence interval.

We carried out our next experiment to assess the effectiveness of our KLT compared to the baseline query expansion, including the re-ranking step. Specifically, we evaluate the impact of $R$ – i.e., the number documents re-ranked with respect to the temporal proximity. Here, we performed the retrieval process, first by re-ranking and
Figure 6.6: The MAP values as function of the value of $\gamma$ for three different standard retrieval models as base for the query expansion models. In each figure from (a) - (c), each graph represents a combination of $K$ and $n$ values, expressed as $\{K\}_{\{n\}}$.

then applying the KL divergence for query expansion ($\text{RERANKING+KL}$), and second by re-ranking and then applying our proposed temporal KL ($\text{RERANKING+KLT}$).

As before, we varied the values of $n$ and $K$.

Figure 6.8 presents the results from this experiment. It depicts several graphs comparing the retrieval performance of the above approaches, using different combinations of the size of the feedback document set $K$ and the number of candidate query expansion terms. So Figure 6.8a shows the results from running QE with TFIDF as a base retrieval model\(^6\), and with $K = 30$ and $n = 8$, and so on. As we can observe in this figure, in all our tests, our proposed KLT with the re-ranking outperforms the standard KL. Moreover, we can see that in all the cases, we obtained the highest MAP values with $R=4000$. And, as before, all the improvements of KLT are statistically significant at 95% confidence interval.

\(^6\)A base model is the retrieval model we run prior to a QE step.
Figure 6.7: The MAP values as function of the values of $K$ and $n$ – expressed as $\{K\}_n$, with the three different retrieval models as bases for the different query expansion models.

**KLT vs KLST**

In this subsection we compare the temporal-aware query expansion model with the spatiotemporal-aware query expansion model KLST. We use the values of $\gamma = 0$ for the linear combination between KL and KLT, which has been shown to yield the best result. Further, we set $\delta = 0.5$ to compute $KLST(t)$ as given by Equation 6.14. Due to the space limitation, we do not present any tuning process for the $\delta$ value.

Table 6.6 summarizes our comparison experiments. In this table we show how much our proposed query expansion models, KLT and KLST, improve the retrieval performance – i.e., the MAP and RP values, as compared to the base IR model BM25 and the baseline KL. The temporal window used is 3DAYS. For both the MAP and RP values, the first columns of the table is the percentage improvement from BM25 to KL; the second column is the percentage improvement from BM25 to KLT; while the third column is the percentage improvement from BM25 to KLST.
Figure 6.8: The MAP values as function of the of number of documents to re-rank, different retrieval models and different values of $K$ and $n$

Note that as mentioned previously, we have omitted the results from applying the Bo1-based QE model because they were not significantly different from the KL results. Further, we chose to include BM25 as the base IR model here since it was the model that gave us the overall best results, compared to TFIDF and LM (See also Section 6.6.3).

Analyzing the results in Table 6.6, we can observe that for all combinations of the number of feedback documents $K$ and number of expansion terms $n$, our geo-spatial and temporal-based QE model, KLST, outperforms both the baseline KL and our temporal-based, KLT, re-weighting model. Specifically, KLST is from 10.6% to 19% better than the baseline method. Moreover, with the best MAP and RP values, KLST is six times better than the baseline KL and around 50% better than the KLT model. To give a better overview of our comparison, Figure 6.9 depicts the differences between the four models with respect to the MAP and RP values. As this figure shows, the effectiveness of our KLST model is noticeably better than the three other models, which also specifically answers our third research question in Section 6.3.
6.7. Conclusions

Table 6.6: Percentage of improvement of MAP and RP using different re-weighting model on BM25. The numbers 1, 2 and 3 in superscript in the table indicate the statistical significance improvements on the baseline, KL and KLT re-weighting models, respectively.

<table>
<thead>
<tr>
<th></th>
<th>ΔMAP(%)</th>
<th></th>
<th>ΔRP(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>KL KLT</td>
<td>KLST</td>
<td></td>
</tr>
<tr>
<td>30_8</td>
<td>2.97</td>
<td>8.20</td>
<td>10.84^12</td>
</tr>
<tr>
<td>60_8</td>
<td>4.57</td>
<td>10.66</td>
<td>15.06^123</td>
</tr>
<tr>
<td>90_8</td>
<td>2.91</td>
<td>13.19</td>
<td>16.77^123</td>
</tr>
<tr>
<td>120_8</td>
<td>2.61</td>
<td>13.34</td>
<td>16.94^123</td>
</tr>
<tr>
<td>30_18</td>
<td>3.62</td>
<td>8.58</td>
<td>11.44^123</td>
</tr>
<tr>
<td>60_18</td>
<td>3.62</td>
<td>12.25</td>
<td>15.55^123</td>
</tr>
<tr>
<td>90_18</td>
<td>2.95</td>
<td>14.33</td>
<td>16.85^123</td>
</tr>
<tr>
<td>120_18</td>
<td>2.84</td>
<td>14.52</td>
<td>17.25^123</td>
</tr>
</tbody>
</table>

Figure 6.9: Comparison of MAP and RP values of KL-ST against other query expansion models, as function of the values of K and n (expressed as \( \{K\}_{\{n\}} \)), using BM25.

6.7 Conclusions

Photosharing applications, such as Flickr, contain many pictures related to real life events, and many of them are annotated with time and location information. The main goal of this work has been to improve existing retrieval models by exploiting this information within event-related image retrieval. Our main idea has been to use picture metadata to emulate a query-by-example analogy. To achieve this goal, we have proposed an extended query expansion model that exploits the temporal information of pictures and the spatial distribution of terms. We thoroughly evaluated our approach by first analysing the retrieval effectiveness with respect to different combinations of metadata fields, and using different standard retrieval models.
Then we conducted several experiments to assess the effectiveness of our two proposed query expansion models; one based on temporal proximity of tag terms and the other based on spatial distribution of tag terms. We compared both methods with existing baseline approaches. The results of these experiments have shown that our approach outperforms the state-of-the-art query expansion models, and that the improvements were statistically significant at a $p < 0.05\%$ level. In particular, we demonstrated that our method is effective even when the amount of information surrounding a picture is small. Finally, by testing our approach on a large dataset, and still getting good results, we can conclude that our approach can handle large-scale data.
Chapter 7

Spatial Features from Geographical Distribution of Social Tags

The availability of a huge amount of geotagged resources on the Web can be exploited to extract new useful information. We propose a set of estimators that are able to evaluate the degree of clustering of the spatial distribution of terms used to tag such geotagged resources. We introduce the concept of *tag point pattern* to derive indexes from the exploratory analysis by taking advantage of the second order Ripley’s $K$-function, previously used in epidemiology, geo-statistics and ecology. The derived model estimates the degree of aggregation of the geotagged resources, taking into account the heterogeneity of the spatial distribution of the underlying population. Further, thanks to subsampling techniques, our approach is able to handle large datasets. Without losing of generality, we perform our experiments on a dataset derived Flickr pictures, as a use case. This dataset consists of tags that were extracted from a set of 1.2 million pictures. We evaluate our proposed indexes with respect to their ability to extract tags related to geographical landmarks and hotspots. Our experiments show that we get good results using our estimators.

7.1 Introduction

In media-sharing applications, such as *Flickr*, the user can add, to their photo, a geotag — i.e, the GPS position, to the resources in addition to the date of creation, and other metadata. In almost all cameras, this metadata are stored directly in the EXIF file. As a result, we can associate tags with related geographical distributions
and positions from a collective contribution.

The exploration of the spatial structure of these tag distributions, such as the clustering tendency can reveal some interesting properties in the semantic of the tags — e.g., tags being related to a landmark or a point of interest. However, there are two most important challenges that we have to tackle when exploring these spatial structures. First, we have to deal with a large amount of data, thus resulting in a high requirement on scalability. Second, the underlying geographic distribution of the data in the dataset may not be uniform. For example, for tagged pictures from Flickr, some areas may have a high number of pictures, while others have a low number of pictures. Hence, the density of pictures over a specific region may not be homogeneous. This may, in itself, be a challenge, since it may cause difficulties in capturing the clustering tendency.

To tackle these challenges we propose a method to derive optimal estimators from the second order Ripley’s $K$-function [134] ($K$-function for short). This is a function used for global test of clustering and quantifying the intensity of spatial patterns at different scales. Using in a case study analysis, $K$-function allows us to analyze the spatial structure and clustering tendency with respect to an underlying population with non-homogeneous intensity. Further, our estimators measure the degree of aggregation of the spatial distribution of picture tags on a geographical map. These kinds of indicators may, for example, be used as features in a query expansion framework, by including the spatial characteristics of a term. As shown later, in our experiments we employ the estimators to extract tags that are related to geographical hotspots from a large set of geotagged pictures in Flickr.

In view of this, the main goal of this work is to propose a method that is able to (1) define a set of estimators that can be used to indicate the clustering tendency based on a given spatial distribution, and (2) capture the spatial characteristics of the spatial distribution. Having a set of estimators that are able to output a real value based on the clustering degree of the tag point pattern may be useful not only for extraction of landmark-related tags, but also for ranking and comparison of different tags.

To our best knowledge, exploratory analysis of tags as this has not been done in depth as we do before. Previous work such as the one by Rattenbury et al. [133] has focused on directly extracting tags related to places, without considering the information about the underlying spatial structure. Earlier, exploratory analysis was adopted in epidemiology [85] to test the existence of clusters of disease and in ecology [124] to understand the ecological process by studying and characterizing the spatial structure of ecological data. Existing methods for evaluating significant clustering from cases has been classified in two different families[164]: cluster detection test and global test. The former, such as the Scan Statistic [80], are a family of clustering
test able to localize significant clusters. The latter, such the diagnostics based on the Ripley’s K-function [135], are generally used to evaluate if the spatial case tends to be located close to each other. Essentially, these methods assess the clustering tendency of the point pattern over the entire considered area.

In summary, the main contributions of the work presented in this chapter are as follows. First, we propose a set of estimators to measure the degree of aggregation of a spatial tag point pattern. These estimators are based on a stochastic model, considering the inhomogeneity of the underlying spatial distribution of the population. Second, we derive our estimators from the exploratory analysis to explore the spatial structures of a tag point pattern, previously used only in geo-statistics and epidemiology. Third, we employ the proposed estimators to extract tag representing a landmark from a large dataset of more than 1 million pictures. For each tag, we also associate a value indicating the distance of maximum significant clustering. Fourth, we perform extensive comparative analysis between the estimators and evaluate their efficiency with respect to spatial tag pattern analysis.

The rest of the chapter is organized as follows. In Section 7.2, we give an overview of the background theory underlying our work and formulate our problem. In Section 7.3, we describe our approach in more detail. In Section 5.4, we evaluate our approach and discuss the results from our experiments. To put our work in a perspective, in Section 7.5, we briefly summarize existing work related to ours. Finally, in Section 7.6, we conclude the chapter and discuss the directions for our future research.

7.2 Problem Definition

In this chapter we extended the concepts of exploratory analysis discussed in Section 2.2, in the domain of collection of images gathered from media-sharing applications. Our target dataset is composed by geotagged resources, annotated with tags. Without losing of generality, we consider a set \( \mathcal{D} = \{ P_1, ..., P_N \} \) of \( N \) pictures gathered from Flickr. Further we assume that each picture \( P_i = \{ g, t, T \} \) is associated with some metadata. This includes locational information represented by a pair of real numbers of latitude and longitude, \( g = (lat, lon) \), temporal information \( t \) represented by the picture timestamp, and the set of tags, consisting of a collection of term \( T \). Hence, it is safe to presume that the pictures can be placed in a spatiotemporal domain, or bounded in a spatial area and a particular time slice. We also define a vocabulary of \( M \) tags \( \mathcal{V} = \{ w_1, ..., w_M \} \), composed of all the social tags used to annotate the pictures in the dataset \( \mathcal{D} \).

Considering the above pictures as our geotagged Web resources that we analyze, we
can model the spatial distribution of pictures taken in a certain geographical area as:

**Definition 7.2.1 (Picture Point Processes)** A Picture Point Process is a Point Process, modeling the spatial distribution of pictures taken in a 2-dimensional study region \( \mathcal{R}^2 \). Any realization of the random variable modeling the Picture Point Process is called Picture Point Pattern.

Therefore, for each term \( w_i \) of the vocabulary \( \mathcal{V} \), we can consider to have a set of \( M_{w_i} \) points \( \{s_{1w_i}, ..., s_{M_{w_i}}\} \) representing the spatial distribution of the tag in the study region. Thus, we now get what we call Tag Point Pattern, formally defined as follows:

**Definition 7.2.2 (Tag Point Pattern)** A Tag Point Pattern \( \mathcal{D}_{w_i} \) is a subset of a Picture Point Pattern \( \mathcal{D} \). It is composed by the geographical position of the pictures annotated with the tag \( w_i \).

In view of this, the problem addressed in this chapter is finding a way to use the exploratory analysis of spatial data to derive estimators that are able to indicate clustering tendency of a tag point pattern with respect to a Picture Point Process. In addition, we want to employ the proposed estimators to rank and extract social tags from the vocabulary \( \mathcal{V} \) that are related to geographical hot-spots, landmarks or Point-of-Interests.

### 7.3 Ranking and Extracting Spatial Hot-Spot Terms

In this section we show how we can use our concept of tag-point pattern in spatial analysis to rank and extract terms related to geographical hot-spots.

Figure 7.1 and Figure 7.2 illustrate the spatial distributions of the Flickr pictures in our dataset \( \mathcal{D} \), displayed on a map. As can be observed, the distribution over the geographical space is not homogeneous. Thus, the density \( \lambda \) for these pictures is not constant but varies depending on a location \( x \). Recalling from our discussion in Section 8.3, this means we have \( \lambda(x) \) to denote the location-dependent density. The reason is that the distribution follows the characteristics of the Flickr dataset, which is not homogeneous. Some regions are highly populated in terms of number of taken pictures, while other regions are sparsely populated or not populated at all. Knowledge about such population characteristics is highly useful when analyzing the pattern structures and clustering tendency.
In the following, we show how our tag pattern point analysis and our proposed approach can tackle the inhomogeneity of the picture population.

### 7.3.1 Case-Control Analysis

As can be inferred from Section 8.1 our method has been inspired by the case-control analysis from epidemiology [85], where the main ideas is to analyze and detect the concentration of diseases based an underlying population. To be able to a case control analysis, we need a diagnostic function. Thus, here we consider the diagnostic function $D_{w_i}(h)$ with respect to the overall spatial distribution of pictures for a specific tag $w_i$. Since the overall distribution of the pictures is not
homogeneous, some tag-point patterns may show aggregation properties, that only depend on the distribution of the population.

So, as defined in [42, 43], our diagnostic function $D_w(h)$ can be computed based on the difference between the $K$-function $K_{w_i}$ for the tag point pattern and the $K$-function $K_{all}$ for the pictures point pattern – i.e.,

$$\hat{D}_w(h) = \hat{K}_{w_i}(h) - \hat{K}_{all}(h),$$

(7.1)

where both $K_{w_i}$ and $K_{all}$ are given by Equation 2.44. This also means that a plot of $D_w$ would depict the interaction between the tag-point pattern and the overall distribution of the pictures. And, for some values of $h$, $D_w(h) > 0$ indicates aggregation in a circle within a radius $h$.

Normally, the significance of a case control analysis is evaluated by calculating an envelope for the function. To do this one can perform Monte Carlo test for null hypothesis with $N=99$ realizations and $\alpha=0.05$ confidence interval – i.e., 95% confidence envelope obtained by Monte Carlo simulations. A $p$-value the test is obtained by estimation. This test calculates a maximum and minimum confidence based values of the envelope for $K(h)$, which can, in turn, be used to test the clustering (the upper bound) or the separation (the lower bound) tendency of a point pattern. Note that the cost of this test step, especially the calculation of the envelope, is quadratic, and is generally not suitable for large datasets.

As an example, Figure 7.4 shows the use of the $K$-function for homogeneous point pattern. Here, we calculated the $K$-function for the point pattern related to the tag **night** (see Figure 7.3). As can be observed from this figure, the values of $K(h)$ are
greater than the upper envelope of the function related to the Poisson Process that we use as benchmark. This means that we have a clustering tendency of the point pattern for all the values \( h \). Note that since the tag \textit{night} is a general tag, its tag point pattern follows the underlying distribution of the population, thus showing aggregation itself, as depicted in Figure 7.3.

### 7.3.2 Proposed Estimators

We propose a method that is able to (1) tackle large set of pictures – i.e., being scalable, (2) produce a real value indicating the degree of aggregation, and (3) determine the radius with highest significant clustering tendency of the tag point pattern. As already mentioned, the main goal of our method to produce a ranked list of terms based on their aggregation capabilities.

Inspired by the sub-sample similarity hypothesis in [79, 105], to estimate the envelope related to the confidence band of \( \hat{K}_{\text{all}}(h) \), we perform the following steps:

1. Obtain \( \{D_i^*\}_{i=1...C} \), sub-sampling the control dataset \( D \) \( C \)-times.
2. Calculate the \( K \)-functions \( \{\hat{K}_{\text{all},i}(h)\}_{i=1...C} \) for each sub-sampled set \( D_i^* \).
3. Estimate the upper envelope as
   \[
   \hat{K}_{\text{MAX}}(h) = \max\{\hat{K}_{\text{all},i}(h)\}_{i=1...C} \quad (7.2)
   \]
4. Calculate the diagnostic function
   \[
   \hat{D}_{w}(h) = \hat{K}_{w}(h) - \hat{K}_{\text{MAX,all}}(h) \quad (7.3)
   \]

As can be inferred from this, we are only interested in tag point patterns that show aggregation. This is why our test only considers the upper envelope \( \hat{K}_{\text{MAX}}(h) \), and not \( \hat{K}_{\text{MIN}}(h) \) which gives a lower limit for testing the significance of the dispersion of a spatial point pattern.

So, once \( \hat{K}_{\text{MAX}}(h) \) is estimated, we propose the following two estimators, to measure the aggregation capabilities of the tag point pattern:

\[
\hat{I}_{\text{SUM}}(w_i) = \sum_{k=1}^{m} \left[ \frac{\hat{D}_{w_i}(h_k)}{\sqrt{\text{Var}(\hat{D}_{w_i}(h_k))}} \right], \quad \text{and} \quad (7.4)
\]
\[
\hat{I}_{\text{MAX}}(w_i) = \max_{k=1...L} \left[ \frac{\hat{D}_{w_i}(h_k)}{\sqrt{\text{Var}(\hat{D}_{w_i}(h_k))}} \right], \quad (7.5)
\]

where the extent of clustering is estimated by the maximum distance and the sum of the distances from the upper envelope of the \( K \)-function for the population data.
More specifically, the estimator $\hat{I}_{SUM}(w_i)$ sums the differences between the two $K$-functions. A high value of $\hat{I}_{SUM}(w_i)$ indicates a point pattern showing strongly aggregation of the tag point pattern in relation to the underlying pictures point pattern. This is because it covers a higher positive area under the function $D_{w_i}(h)$. Further, the value of the estimator $\hat{I}_{MAX}$ determines the highest positive difference between the $K$-function of the tag point pattern and the pictures point pattern over different scale values $h_{kk=1...L}$. Thus, its objective is to give more importance on the maximum distance between the two pattern points, over different scales. In Section 5.4, we study the effects of using the two proposed estimators in ranking of a list of tags.

In addition to these, we need to estimate the most significant scale value of the clustering tendency. This can be done as follows:

$$h_{MAX} = \arg \max_{h_k} \left[ \frac{\hat{D}(h_k)}{\sqrt{\text{Var}(\hat{D}(h_k))}} \right]$$ (7.6)

It worth to note that the cost of estimating both estimators is suitable for large datasets, since for both $\hat{I}_{SUM}$ and $I_{SUM}$, we can avoid using Monte Carlo estimation. In addition, we can start from a sub-sample similarity hypothesis and thus considering only a sub-sample of the control data.

In summary, since one of the main purposes of this work is to extract tag representing geographical hot-spots or landmarks, we further consider the number of occurrences of the pictures tagged with the extracted terms. The use of this parameter to order or rank the output list of terms is elaborated on in Section 5.4.

7.4 Experiments

7.4.1 The Dataset

Before presenting our experiments and the results, we give an overview of the dataset that we base our experiments on.

As mentioned before, we mainly used Flickr pictures to do our experiments. More specifically, these consist of 1.073.364 images that were taken around London between January 1st, 2007 and January 1st, 2011, in a square area between 51.395778 and 51.649555 in latitude, and between -0.341263 and 0.151062 in longitude. Recall from Figures 7.1 and Figure 7.2 that the pictures in our dataset are not uniformly distributed in space over the area we have selected.
Further, Figure 7.5(a) shows the distribution of the numbers of tags per picture in the dataset. The number of tags per pictures has a mean of 7.104 and median of 5. The maximum number of tags per picture is 75.

Because our dataset is based on pictures uploaded by different users, the tags may contain noise. In particular, the temporal tags may be inconsistent. To deal with this, we had to filter out or remove pictures which the upload time differs from the capture time\(^1\) with more than 1 month. This has resulted in 212,024 or 20 % of the total number of pictures being filtered out. In addition, we removed another set of 9,402 pictures that have capture date greater than upload date, which can also safely be seen as noise. To summarize, Figure 7.5(b) shows the distribution of the differences between the capture dates and upload dates for the pictures in our

\(^1\)Capture time is the time when a picture was taken.
dataset.

The geographical information in our dataset is in form of latitude and longitude coordinates. Since they are not in euclidean space, to be able to use this information in our approach, we had to transform each locational information entry into Universal Transverse Mercator (UTM)\(^2\). This, in turn, makes us able to see each picture location as a point in the euclidean space, and thus allowing for more precise distance calculation.

### 7.4.2 Pre-Processing Step

Data gathered from Flickr may contain pictures with the same latitude and longitude values. This happens because users sometime copy geotags and set of textual tags for more than one photos of the same groups. Therefore, for each tag point pattern, we removed such duplicated occurrences of geotags. Further, to perform the CSR test we assume that the point processes must have the property of *simplicity*. This means that all geographical coordinates representing an element of the point pattern must be different, without duplication and that multiple coinciding points cannot occurs. Formally, this means \( \not\exists (i, j) \mid x_i = x_j \), where \( x_i \) and \( x_j \) are elements of the picture point pattern \( D \).

As part of the dataset preparation we perform a preprocessing step consisting of tokenization – i.e., UTFTokenization based on whitespace and punctuation marks, english

7.4. Experiments

stopword removal, and stemming using the well-known Porter Stemmer algorithm.

7.4.3 Results

In this section, we present the results from our experiments.

Spatial Hot-Spot Tag Extraction

In this experiment we evaluate our proposed indexes $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$ as measure to check the aggregation capabilities of a point pattern. The purpose of our analysis to study the quality of the extracted ordered list of tags representing geographical hotspots and landmarks.

We manually evaluate the top-100 tags extracted from the ordered list. Thereafter, to consider the influence of the number of occurrences of the tag point pattern, we also re-order the first 150 tags by the number of pictures tagged. Table 7.1 and Table 7.2 show the top-10 terms extracted based to the different estimators.

Since the output of our mining method is an ordered list we examine the precision at rank $n$ ($P@n$) [9] with $n = \{5, 10, 20, 50, 100\}$, representing the percentage of relevant tags (term representing an hot-spot/landmark) in the top-$n$ tags:

$$ P@n = \frac{|R_n|}{n}, $$  

where $R_n$ is the number of relevant tag at rank $n$. To facilitate the evaluation of the tags, we query each ranked tag with the term london, on Wikipedia\(^3\) to check if the tag refers to a landmark. The results of this experiment are summarized in Table 7.3.

<table>
<thead>
<tr>
<th></th>
<th>$\hat{I}_{SUM}$</th>
<th>$I_{ord}^{SUM}$</th>
<th>$\hat{I}_{MAX}$</th>
<th>$I_{ord}^{MAX}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P@5$</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$P@10$</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>$P@20$</td>
<td>0.95</td>
<td>1.0</td>
<td>0.90</td>
<td>1.0</td>
</tr>
<tr>
<td>$P@50$</td>
<td>0.92</td>
<td>1.0</td>
<td>0.92</td>
<td>0.96</td>
</tr>
<tr>
<td>$P@100$</td>
<td>0.92</td>
<td>0.94</td>
<td>0.92</td>
<td>0.92</td>
</tr>
</tbody>
</table>

Table 7.3: Precisions values using the different indexes

From Table 7.3, we can observe that both indexes show promising $P@n$ values. The lowest $P@100$ value is 0.92 for both $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$, whereas the highest $P@100$

value is 0.94. For both indexes, the best result is obtained when the tags are also ordered by their number of occurrences in the population.

All Wikipedia pages corresponding to relevant hot-spot or landmark tags revealed from the lists were also tagged with a GPS coordinate. Analysing the 7 non-relevant tags, we noticed that three of them, chelseafowershow, cansfestival and rhs, referred to events that used to happen in a particular geographical area. All of these tags were in the list and ordered by $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$, but only one of them was in $\hat{I}_{ord_{MAX}}$. Note that the list was also ordered by number of occurrences. Another two tags, eventhorizon and eros, referred to a sculpture located in Trafalgar Square and a statue in Piccadilly Circus in London, respectively. The sixth tag va is an acronym for Victoria Albert Museum. Finally, the last tag, mashed, does not have any immediate reference to any landmark term in Wikipedia.

In conclusion, based on the precision values and the observation of the tags considered not relevant, both indexes, $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$ can be said to be robust. Comparing the two indexes, $\hat{I}_{SUM}$ gave the overall best $P@n$ values. Table 7.3 also shows that improvement in term of higher precision values can be obtained when ordering the ranked list by number of occurrences of the tags.

**Comparison between Indexes**

To further analyze the differences between the point patterns ordered using our two indexes, in this section we perform a more in depth comparison of the effects of using the indexes, $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$. To do this, we compare the two resulting ranked lists of tags.

![Figure 7.6: Rank distance $R_{diff}$, between the top-50 tags ordered by the two indexes $\hat{I}_{SUM}$ and $\hat{I}_{MAX}$, respectively.](image)
7.4. Experiments

First, we analyze the differences between the ranking of tags ordered by \( \hat{I}_{SUM} \) and \( \hat{I}_{MAX} \). Figure 7.6 shows the differences for the top-50 tags ordered by the indexes, respectively.

From the Figure 7.6, we can see that the tags in the first position of the list, ordered by \( I_{MAX} \) are pushed further down, compared to the list ranked by \( I_{SUM} \). This means that the two estimators capture different characteristics of the point patterns. This effect is also supported by the observation that in average, the same tags have lower ranks in the list ordered by \( I_{SUM} \) than by \( I_{MAX} \).

In the following, we further analyze these differences. First, we compare the values of the significant aggregation radius of the tag point pattern. The comparison is performed over the top-50 extracted using the list ranked by the two indexes, respectively. More specifically, we compare the cumulative average of the radius of significant clustering as follows:

\[
AVGrad(n) = \frac{\sum_{i=1}^{n} h_{max}(w_i)}{n}
\]  

(7.8)

Figure 7.7 summarizes this result. We observe that using the two indexes the first difference is that \( I_{MAX} \) tends to give higher values to tag point patterns showing significant agglomeration at higher scale values.

![Figure 7.7: Cumulative Average radius of significant clustering for the top-50 tags extracted. AVGrad(n) is expressed in kilometers.](image)

To further illustrate the differences, we analyze the characteristics of the tags that have the highest ranking distance between the lists ordered by \( I_{SUM} \) and \( I_{MAX} \), respectively. Table 7.4 and Table 7.5 shows the top-5 tags that were obtained first by ordering using the indexes, and then by ordering the top-50 using rank differences
Table 7.4: Top-5 tags extracted using rank differences in ordering the list by $I_{MAX}$

<table>
<thead>
<tr>
<th>TAG</th>
<th>$h_{max}$</th>
<th>$R_{I_{MAX}}$</th>
<th>$R_{I_{SUM}}$</th>
<th>$R_{diff}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>wwt</td>
<td>0.85</td>
<td>17</td>
<td>116</td>
<td>-99</td>
</tr>
<tr>
<td>greatportlandstreet</td>
<td>0.91</td>
<td>21</td>
<td>113</td>
<td>-92</td>
</tr>
<tr>
<td>zsl</td>
<td>0.81</td>
<td>10</td>
<td>96</td>
<td>-86</td>
</tr>
<tr>
<td>grosvenorsquare</td>
<td>1.01</td>
<td>30</td>
<td>114</td>
<td>-85</td>
</tr>
<tr>
<td>oldroyalnavalcollege</td>
<td>1.01</td>
<td>30</td>
<td>110</td>
<td>-80</td>
</tr>
</tbody>
</table>

Table 7.5: Top-5 tags extracted using rank differences in ordering the list by $I_{SUM}$

<table>
<thead>
<tr>
<th>TAG</th>
<th>$h_{max}$</th>
<th>$R_{I_{MAX}}$</th>
<th>$R_{I_{SUM}}$</th>
<th>$R_{diff}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>westkensington</td>
<td>0.36</td>
<td>153</td>
<td>48</td>
<td>105</td>
</tr>
<tr>
<td>astoria</td>
<td>0.39</td>
<td>132</td>
<td>34</td>
<td>98</td>
</tr>
<tr>
<td>madametussauds</td>
<td>0.26</td>
<td>109</td>
<td>27</td>
<td>82</td>
</tr>
<tr>
<td>britishlibrary</td>
<td>0.94</td>
<td>123</td>
<td>47</td>
<td>76</td>
</tr>
<tr>
<td>eros</td>
<td>0.78</td>
<td>87</td>
<td>15</td>
<td>72</td>
</tr>
</tbody>
</table>

Figure 7.8: Cumulative Average radius of significant clustering for the top-$n$ tags extracted. $AVGrad(n)$ is expressed in kilometres.

in the two lists. Further, Figure 7.8 shows plots of the distributions of the cumulative average of the $h_{max}$ values for the first 10 tags of Table 7.5 and Table 7.4.

Figure 7.9 depicts plots of the diagnostic $D$ for the tags in Table 7.4 and Table 7.5. This shows the capabilities of the index $I_{MAX}$ in that it gives more weight to point pattern having evident peaks in the shape of the related $K$-function and showing a low significant aggregation radius as we can observe also in the $h_{max}$ values in Table 7.4 and in Table 7.5. This confirms our observation in Figure 7.7 and Figure 7.8.
7.5 Related Work

Geographical information from Web resources has been applied for different scopes in the past. For example, Kennedy and Naaman [78] proposed a search system that is able to retrieve representative pictures for a given landmark by using location and tags, as well as visual features. In addition, Zheng et al. [192] employed geographical knowledge for similar purpose to recognize and model landmarks from geotagged pictures in a Web-scale environment.

Similar to our work, the researches by Chen and Roy [30] and [133] aimed at extracting semantics from geotagged pictures annotation by applying both temporal and geographical annotation in the images. In particular, Rattenbury et al. [133] proposed a mining algorithm to extract tags related to places based on the geographical information of pictures. In contrast our approach, although they also applied clustering, they did not do any explicit exploratory analysis to analyze the spatial clustering or cluster tendency.

Finally other applications based on the geographical knowledge extracted from Web resources are trajectory mining and travel Itinerary mining for tourist purposes [40, 126] and activity recommendation [74]. In addition, Silva and Martins [159] developed a tag suggestion method based on analysis of collective knowledge, spatial proximity extracted from picture collections and automatic assignment of geographical location to a tagged picture.
7.6 Conclusion

In this chapter we have proposed the concept of tag point pattern based on the spatial point pattern concept from Statistics. Our main idea has been to develop two estimators from exploratory analysis to measure the clustering tendency of the tag spatial point pattern. This has allowed us to not only perform a clustering test, but also to produce an output value that can be used to measure the clustering tendency of a spatial point pattern. Further, we have studied how we can apply our estimators to extract terms related to geographical hotspots or landmarks in a specific geographical area. Our focus has been to use our approach on geotagged Web resources that have mainly been gathered from social media-sharing applications.

Our extensive experiments have shown the effectiveness of both of our proposed estimators, when applying on dataset gathered from Flickr. Further, with our comparative analysis, we have been able to compare the two estimators with respect to their ability to capture spatial point patterns with different characteristics. This comparison has been conclusive. Moreover, to allow using our method on a large dataset we have applied subsampling techniques, which also have shown that our method is scalable.

Nonetheless, we believe there are still rooms for improvements. In the next chapter, we will find a way to exploit the capabilities of our indexes within Information Retrieval – i.e., query expansion techniques that combine these spatial estimators with standard scores related to text or term distributions.
Chapter 8

Event-Related Image Search: Non-Timestamped Queries

In Chapter 6 we presented a query expansion model exploring and leveraging the geo-temporal profile of the terms within the temporal neighborhood given a timestamped query. In this chapter we explore the more general case of a tag-based event retrieval system given a textual query without any temporal information associated. 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 this kind 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. Our main idea is to use these features in a machine learning-based approach to query expansion - i.e., to select the best expansion terms. Specifically, we apply rigorous statistical exploratory analysis of spatial point patterns to extract these geo-spatial features. With these features, we are 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 - 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 our expansion terms. To evaluate our method, we perform several experiments, including well-known feature analyses. Such analyses reveal the contributions of our proposed geo-spatial features on the overall retrieval performance. The results from our experiments demonstrate the effectiveness and viability of our method.
8.1 Introduction

The main goals of this work are to build a framework to extract a set of geographical features from geographical raw data of documents or pictures, and 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 the integration of a machine-learning-based approach in our retrieval system. As part of this, we analyze how these features can be used in a query expansion framework. As we elaborate later, here we are especially interested in the contributions of these features on the selection of expansion terms from feedback documents.

Toward this end, we propose a novel framework that improves the retrieval effectiveness when doing tag-based search of pictures by including the geographical profile of terms. We have developed a new method to extract a set of features from 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. Although the concept of geographical-based or spatiotemporal-based retrieval might no longer be new, to our best knowledge, geographical features like ours have only been used and explored in few works. Existing approaches have mainly been concerned with point-of-interests (POI) extraction [132] and trajectory mining [183]. With the constantly increasing number of geotagged pictures in – e.g., Flickr\(^1\), exploring the geographical raw metadata has become crucial.

In summary, the main contributions of the work presented in this chapter are as follows. First, we propose a new robust set of geographical features that can be used to (1) analyze the geographical structure of the geographical distribution of tags to reveal their clustering tendency, and (2) analyze the tag relatedness between two tags by exploring the correlation between their geographical distributions. To do this, we have developed new measures derived from a well-founded Exploratory Analysis within Statistics. More specifically, we adapt the Ripley’s \(K\)-function and Ripley Cross-\(K\) function (\(K\)-function and cross-\(K\) function for short) [134] as part of our approach to extract the geographical features. Second, we show how these 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, our experimental eval-

\(^1\)Around 220M of Flickr pictures are geotagged. See also \url{http://www.flickr.com/map/}
8.2. Relation to Other Work

A research area closely related to the work presented in this chapter is Pseudo Relevance Feedback (PRF). Generally speaking, PRF refers to techniques to average top-retrieved documents to automatically expand an initial query. Pseudo Relevance Feedback has been studied widely in information retrieval both to extend existing retrieval models [84, 188, 166, 24], and as part of Query Expansion (QE) frameworks [26, 142]. Specifically, Lavrenko and Croft [84] and Zhai and Lafferty [188] propose two methods; the Relevance Model and the Mixture Model, respectively, to include feedback information in the KL-divergence retrieval model [82]. 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 able to improve the standard retrieval models with respect to retrieval effectiveness [101]. This has also been the main motivation for why we have chosen to include them in our study.

Cao et al. [24] present a classification approach to automatically select good expansion terms from a set of candidate terms in 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 are composed by 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. [89] 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\textsuperscript{2} social bookmarking Web service, as a source for extracting useful expansion terms. Such use of social annotation as source for improving retrieval performance has also previously been investigated by Zhou et al. [193]. These approaches can be related to ours in that we also use classification to select good expansion terms. However, to our knowledge, none of these include either temporal, geo-spatial or geo-spatiotemporal features.

As discussed later in this chapter, we are interested in investigating the contributions of the temporal characteristics of a term in the pseudo relevance feedback context. Within event retrieval, the usefulness of the temporal information is evident. Also within general information retrieval, results from existing works have proven its usefulness. For example, Dakka et al. [37] and Jones and Diaz [73] show how the temporal profile of queries can be used to improve existing retrieval models; whereas Keikha et al. [77] and Whiting et al. [179] 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, focusing on event retrieval, we argue the necessity of including the spatial profile of social tags, as well as the temporal profile. It seems that the combination of both temporal and spatial features of social tags to improve the retrieval effectiveness has not still been sufficiently investigated. Only few methods – e.g., [131, 189], incorporate some temporal and spatial correlation measures to investigate term-to-term relatedness. Specifically, Radinsky et al. [131] propose a method to improve the semantic relatedness measure of two terms by capturing the correlation between the temporal profiles of tag and concepts associated to the two terms. Zhang et al. [189], on the other hand, analyze the tag relatedness by using different correlation measures, based on spatial and temporal co-occurrence. In summary, although these approaches are indeed related ours, to our best knowledge, 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 images retrieval in general [163], thus further supporting the necessity of our approaches.

\section{Preliminary}

In this section we first give an overview of the data our approach is based on and define the problem we address, and then introduce some of the techniques we fond

\textsuperscript{2}http://www.delicious.com/
our approach on.

### 8.3.1 Data and Problem Definition

In social media-sharing applications, resources are usually tagged with terms – i.e., tags, that describe the content of the resource. Sometime such resources also come with a pair of real value – i.e., latitude and longitude, specifying their geographical locations – also called geotags. We refer to such a resource as a geotagged resource.

So with this in mind, let \( \mathcal{D} = \{P_1, \ldots, P_N\} \) be a set of geotagged resources. We assume that each resource \( P_i = \{g, t, T\}, i = 1, \ldots, N \), can be annotated with a set of tag \( T \), a temporal timestamp \( t \) and a geotag \( g = \{lat, lon\} \). Without loss of generality, let our resources be geotagged pictures gathered from Flickr. They may not necessarily contain all of the above information at the same time.

Now, let \( \mathcal{E} = \{E_1, \ldots, E_M\} \) be a set of picture clusters \( E_i = \{P_{j_1}, \ldots, P_{j_{N_i}}\}, \) \( i = 1, \ldots, M \), from our dataset, each of which including images related to the same event. Next, let \( P_{j_q} = \{T_{j_q}\} \) denote a query picture related to an event \( E_{i_q} \in \mathcal{E} \), without geotags and temporal timestamps. This means that to make our approach as general as possible, we assume that a query picture does not contain any information about where and when it was taken, but only a set of textual tag terms. Hence, our problem concerns how we can effectively retrieve event-related pictures with a query picture \( T_q \), using only the tags. To solve this problem, we first investigate how current state-the-art information retrieval methods perform when applied to our dataset. Thus, they serve as the baseline for our experimental evaluation. Thereafter, 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. Finally, we compare our method with baseline methods.

Using spatial statistics for the tags, we can build a spatial profile for each tag. To do this, we employ a large dataset of \( L \) geotagged pictures \( \hat{\mathcal{D}} = \{\hat{P}_{g_1}, \ldots, \hat{P}_{g_L}\} \). Here, we let \( \mathcal{V} = \{w_1, \ldots, w_W\} \) be the vocabulary of the dataset \( \hat{\mathcal{D}} \) – i.e., the set of social tags used to annotate \( \hat{\mathcal{D}} \). Then, as described in Section 8.4, by analyzing the spatial profile characteristics, we can extract some spatial features from each tag \( w_i \in \mathcal{W} \). The main principle behind our approach is to use statistical exploratory analysis to extract these spatial features. So, to be able to use and understand the ideas of exploratory analysis translated into our domain, we define the following concepts.

First, as mentioned before we consider the Flickr pictures as our geotagged Web resources. Thus, we model the spatial distribution of pictures taken in a specific
geographical area as picture point processes, as defined in Definition 7.2.1. We also base our method on the Definition 7.2.2 of Tag Point Pattern.

### 8.3.2 Exploring Interaction between Spatial Patterns

With the above formal definitions of our concepts of Picture Point Process and Tag Point Pattern, we can derive our spatial features by borrowing some concepts from the exploratory analysis of spatial point pattern [42]. We derive our geo-spatial features using multivariate Ripley $K$-function. Thus, to help understanding our approach, we now give a brief overview of the multivariate Ripley $K$-function, that we use to study the interaction between two or more spatial point patterns.

The multivariate Ripley’s $K_{ij}(h)$ function is a generalization of the Ripley’s $K(h)$ function, used to analyze the characteristics of an isotropic spatial point process [134]. It contains information about clustering and dispersion of point patterns at different scales $h$. The multivariate form aims at answering questions regarding the interaction between two (bivariate) or more (multivariate) point patterns, and it is specified as follows [134]: Let $\lambda_i$ and $\lambda_j$ be the intensity of the spatial point patterns $i$ and $j$, and assume $\lambda_i$ and $\lambda_j$ being constant throughout $\mathbb{R}^2$. Then,

$$K_{ij}(h) = \lambda_j^{-1} E(\#\text{distance points } i \text{ within distance } h \text{ from an arbitrary point } j)$$

(8.1)

Reducing to the case of two point patterns, there are 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 most used estimation of the functions $K_{ij}(h)$ was proposed by Ripley [134] as follows:

$$\hat{K}_{ij}(h) = \sum_k \sum_l I_h(d_{ikjl})$$

(8.2)

where $d_{ikjl}$ is the distance between $k$-th point of type $i$ and $l$-th observed point of type $j$, $I_h(d_{ikjl})$ is an indicator which value is 1 if $d_{ikjl} \leq h$ and 0 otherwise. $\hat{\lambda}_i = n_i/a(R)$ and $\hat{\lambda}_j = n_j/a(R)$ 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 $\hat{K}_{ij}$ functions are used in exploratory analyses to study the relationship between two spatial point patterns. For example, in the independence approach proposed by Lotwick and Silverman [96], 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, $K_{12}(h)$ is given by $\hat{K}_{12}(h) = \pi h^2$. Then, the empirical cross-$K$ function $\hat{K}_{ij}(h)$ calculated on the spatial point pattern can be compared with the null model to
8.3. Preliminary

identify attraction when \( \hat{K}_{ij}(h) > \pi h^2 \), spatial independence when \( \hat{K}_{ij}(h) = \pi h^2 \), or repulsion when \( \hat{K}_{ij}(h) < \pi h^2 \), between the two sets of spatial point patterns.

The graph of the \( \hat{K}_{ij}(h) \) function has normally a parabolic curve, which makes it less straightforward to interpret. For this reason, we usually apply a so-called L-function, defined as follows:

\[
L_{ij}(h) = \sqrt{\frac{K_{ij}(h)}{\pi}} \tag{8.3}
\]

Using the same assumption of independence of spatial point patterns as before, \( L_{ij}(h) = h \). In addition, as with the K-function, \( L_{ij}(h) > h \) indicates attraction between the point patterns, \( \hat{L}_{ij}(h) = h \) shows spatial independence, whereas \( \hat{L}_{ij}(h) < h \) means repulsion. We further normalize the L-cross function by considering a D-function for two tag point patterns as \( \hat{D}_{ij}(h) = \hat{L}_{ij}(h) - h \). Again \( \hat{D}_{ij}(h) > 0 \) indicates attraction between the two spatial point patterns \( i \) and \( j \), \( \hat{D}_{ij}(h) = 0 \) independence and \( \hat{D}_{ij}(h) < 0 \) repulsion. In the rest of the chapter we refer this function as D-function.

**Example: usage of cross-D function**  To explain our ideas, let Old Royal Naval College and University of Greenwich be two tags, both referring to areas in London. Then, consider a cross-L function \( L_{12} \), between two tag point patterns (as specified in Definition 7.2.2) related to these two tags, respectively. We can observe that although these tags are different, they are connected. Specifically, the University of Greenwich is located within the area of the Old Naval College. Thus, they refer to the same geographical entity. As a result, pictures tagged with Old Royal Naval College are spatially attracted to pictures tagged with University of Greenwich (See Figure 8.1a and 8.1b). To further illustrate this relatedness, consider the corresponding cross-D function \( D_{12}(h) \) with \( h \in [0, 2] \) km in Figure 8.2. With a statistical test we can check the validity of our observation about the attraction among the involved spatial point patterns. That, is as can be seen in Figure 8.2, 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 \( h \). Hence, based on our rules above we have attraction between the two point patterns.

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.

---

4See http://en.wikipedia.org/wiki/Old_Royal_Naval_College
5Here, the envelope was computed simulating the random labeling with null model, and with 99 simulations.
Figure 8.1: Spatial distribution of the Tag Point Patterns related to the tag Old Naval College and the tag University of Greenwich at two different zoom (a) and (b).

Figure 8.2: 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). 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.

8.4 Spatial Distribution of Tags

We apply a collection of features derived from the bivariate Ripley L-cross function presented in Equation 8.3 and the Ripley L-function for single tag point pattern. We divide the spatial characteristics in two main classes: (1) single-term spatial features and (2) term-to-term spatial similarity. The former consist of features, each of which determining the aggregation tendency of a single tag spatial point pattern. The latter is a set of features related to the geographical similarity between the spatial profiles of two considered tags $w_i$ and $w_j$. 
8.4.1 Single and Term-to-Term spatial features

Let $S$ be a scale interval in kilometers. We divide the interval set in $K$ discrete and equidistant points $h_k$ with $k = 1, \ldots, K$. This means that the D-function is calculated over this interval. As already mentioned, using this function we can capture the clustering tendency of a tag point pattern. This is done by extracting both the positive area within the intersection between the D-function and the curve representing the null hypothesis, and the maximum distance between them. With our discrete D-function they are given by:

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

and

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

respectively. Here, $w_i$ is a specific tag with its related tag point pattern $i$. Note that a high value of $\hat{I}_{SUM}(w_i)$ means we have a tag point pattern that shows strongly aggregation among pictures that are covered by the related underlying picture distribution. Similarly, for bivariate point patterns, to estimate the attraction tendency of two tag point patterns, we use the following:

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

and

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

where $w_i$ and $w_j$ are two specific tags with their tag point pattern $i$ and $j$.

We can further extend these two features by considering their behavior at different scales. For this we define the relative discrete positive area (RDPA) and relative discrete maximum distance (RDMD) as a generalized form of the estimators $\hat{I}_{SUM}(w_i), \hat{I}_{SUM}(w_i, w_j), \hat{I}_{MAX}(w_i)$ and $\hat{I}_{MAX}(w_i, w_j)$, given by

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

and

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

where $f < g$ are two indexes related to two point $h_f$ and $h_g$ of the scale interval $S$. Hence, $\hat{g}_{RDPA}$ represents the positive area between the D-function and the null hypothesis in a specific scale interval, while $\hat{g}_{RDMD}$ is the maximum distance within the same considered interval. So if we set $f = 1$ and $g = K$, then
\[ \hat{g}_{RDPA}(w_i, [h_f, h_g]) = \hat{I}_{SUM}(w_i) \quad \text{and} \quad \hat{g}_{RDMD}(w_i, [h_f, h_g]) = \hat{I}_{MAX}(w_i). \]

This generalization captures more features dividing and summarizing the spatial characteristics over more sub-intervals of the original scale interval. A similar approach can be done for the bivariate case. This means that we can compute \( \hat{g}_{RDPA} \) and \( \hat{g}_{RDMD} \) for the \( D_{ij} \) function.

### 8.4.2 \( n \)-order Spatial Features

The features in Equation 8.8 and 8.9 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. In addition, 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 some short scale sub-interval. To also capture this characteristic, we propose a set of features, called first order spatial features that can capture the information on the shape of the curve of the D-function (still over the scale interval considered and all its sub-intervals), by computing a derivative \( f'(x) \) of a source function \( f(x) \). Geometrically this represents the slope coefficient of the tangent of the source curve at point \( x \). Our idea is to analyze 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 employ the forward finite difference \(^6\) that is a discrete equivalent of the derivative function as follows:

\[
\hat{D}'_i(h) = \Delta_{l,m} \hat{D}_i(h) = \hat{D}_i(h_l) - \hat{D}_i(h_m), \quad \forall h_l < h_m \tag{8.10}
\]

where \( h_l \) and \( h_m \) are two specific scale points. Note that the value of \( \hat{D}'_i(h) \) is positive in each scale point where the D-function increases, and negative for all scale points where D-function decreases. Moreover, the higher the positive value of the \( \hat{D}'_i(h) \) is, the higher the increasing intensity of the function becomes. Using similar a equation, can compute the derivative of \( \hat{D}_{ij}(h) \) as \( \hat{D}'_{ij}(h) \) for the bivariate form of the the D-function.

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 more spatial features. We call such features second order spatial features, which we get by further deriving the function \( \hat{D}_i(h) \). So 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 estimators RDPA and RDMD in Equations 8.8 and 8.9.

\(^6\)http://en.wikipedia.org/wiki/Finite_difference
8.4.3 Query Expansion Framework

Although query expansion (QE) techniques have been one of the most studied approaches within the information retrieval field since the work by Maron and Kuhns [107], new application areas have made QE still needed to improve the retrieval effectiveness [25]. Nevertheless, QE is not the main focus of this work. Rather, we use it as a framework to evaluate the effectiveness of our proposed method on event-related image retrieval. In this section, we elaborate on how we use our proposed spatial features within a QE framework. In addition we explain how these spatial features can be combined with temporal features within the same framework.

Baseline KL Expansion Model

A general query expansion model is a post-processing step, in the retrieval system, expanding and re-weighting an original query $q$ with terms from top-$k$ retrieved documents that are assumed to be pseudo-relevant. Such top-$k$ retrieved documents are also called feedback documents. For our purpose, we consider the Kullback-Leibler (KL) divergence based approach, an example of a query expansion approach that has been proven to be effective with respect to retrieval performance [82]. With KL, the main idea is to analyze 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 [26]. The terms chosen for the query expansion are those with the highest KL-scores, and thus contributing to the highest divergence [26]. To compute the KL-score for a specific term $t$ in the feedback documents, we use the following equation [26]:

$$KL = P_{Rel}(t) \log \left( \frac{P_{Rel}(t)}{P_{Coll}(t)} \right), \quad (8.11)$$

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.

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 [142] using the Rocchio’s Beta equation, given by [123]:

$$\hat{w}(t_q) = \frac{tf_{q,t}}{\max tf_q} + \beta \frac{w(t_q)}{\max w}, \quad (8.12)$$

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 $tf_{q,t}$ denotes the frequency of the term in the query.
Proposed Framework
As already mentioned, our approach is based on query expansion. Our main idea deals with how to choose good expansion terms to maximize the retrieval performance. Figure 8.3 gives an overview of the principle behind our framework, which we explain below.

Let \( q = \{q_1, \ldots, q_n\} \) be a query of \( n \) terms, and \( \mathcal{E} = \{e_1, \ldots, e_m\} \) denote the set of candidate terms for the query expansion process. Inspired by the work by Cao et al. [24], we define a good candidate expansion term \( e_i \) being a term that improves the retrieval performance of the original query \( q \). More specifically, we compute the average precisions (AP) gained from running the original query \( q \) and the query we get from expanding \( q \) with a specific candidate term expansion \( e_i \), and calculate the difference between them as follows:

\[
AP_{\text{diff}}(q, e_i) = \frac{AP(q + e_i) - AP(q)}{AP(q)} \tag{8.13}
\]

This means that we consider a candidate expansion term to be a good term if \( AP_{\text{diff}}(q, e_i) \) is positive, and bad otherwise. In practice, we set a threshold \( \lambda \) to control the difference value such that \( AP_{\text{diff}}(q, e_i) < \lambda \) would indicate a bad term, whereas \( AP_{\text{diff}}(q, e_i) > \lambda \) would mean we have a good term. To do the actual term selection, we define this task as a binary classification problem. The main idea is to learn a classifier to discriminate the good expansion terms from the bad ones. Here we use Equation 8.13 as a basis for the learning process, and to define the positive examples for the classifier.
In brief, we first represent each expansion term $e \in \mathcal{E}$ as a vector of features (see Section 8.4.3). Thereafter, as part of the retrieval process, we use a classifier to compute a confidence value as function of $e$ (see Section 8.4.3). Finally, we combine this score with the baseline KL score of the same candidate term to re-rank the set of candidate expansion terms $\mathcal{E}$. Algorithm 3 summarizes the query expansion (QE) process.

Algorithm 3 Query expansion procedure incorporating the geo-temporal dimensions.

1: Query $q$ by using 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: $X \leftarrow \text{ExtractTermFeats}(e_j, Q)$  ▷ Sec. 8.4.3
6: $Y \leftarrow \text{ExtractTemporalFeats}(e_j, Q)$  ▷ Sec. 8.4.3
7: $Z \leftarrow \text{ExtractGeoFeats}(e_j, Q)$  ▷ Sec. 8.4.3, Sec. 8.4.3
8: Calculate confidence value $\text{Conf}$  ▷ Sec. 8.4.3, Fig. 8.4
9: Combine KL score and confidence value $\text{Conf}$ in a single score $\to KL_{Final}(q, e_j)$  ▷ Sec. 8.4.3
10: end for
11: Rank $e_j \in \mathcal{E}$ terms according to $KL_{Final}(e_j) \to \mathcal{E}_{Rank}$
12: Re-build $q$ with the top-k terms from $\mathcal{E}_{Rank} \to \hat{q}$
13: Query $\hat{q}$ by using ranking model $r$

Feature Set
To learn a classifier, we define a vector of features for each candidate expansion term $e_i$ from the top-K retrieved items, given a query $Q = \{q_i\}$. This vector of features is divided into three families of features that are briefly discussed in the following.

Term Features $\mathcal{X}(e_i, Q)$. The first set of features is a family document-characterizing features. They were chosen based on the assumption that terms that contribute to improve the retrieval effectiveness are those that are most frequent and distinctive [24]. As proposed by Cao et al. [24] and Lin et al. [89], we can use features related to the distribution of the candidate term $e$ in feedback documents and the whole collection, and those capturing the co-occurrence of $e$ with the terms in the original query $q$. In our experiments, they constitute the features in the baseline approach, which basically consist of traditional statistical features such as Document Frequency (DF). Note that since a tag generally appears only once for each picture – i.e., $tf = 1$, we do not consider Term Frequency (TF) as a relevant features. Finally all the proposed features are related for the whole collection and for the feedback documents for a total of twelve features. In Table 8.1 we summarize the features (for
space limitation only the ones related to the candidate expansion term \( e \) where \( N \) indicates the total number documents, \( C(q_i, e) \) indicates the number of documents in which \( q_i \) and \( e \) co-occurs and finally \( \Omega \) is the possible set of query term pairs.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( DF_0(e) )</td>
<td>Raw document frequency</td>
</tr>
<tr>
<td>( DF_1(e) )</td>
<td>( \log(\frac{N}{DF_0}) )</td>
</tr>
<tr>
<td>( DF_2(e) )</td>
<td>( \log(1 + \frac{N}{DF_0}) )</td>
</tr>
<tr>
<td>( DF_3(e) )</td>
<td>( \log(\frac{N}{DF_0}) )</td>
</tr>
<tr>
<td>( CoOccSingle(e) )</td>
<td>( \log\left(\frac{\sum_{q_i \in Q} C(q_i, e)}{</td>
</tr>
<tr>
<td>( CoOccPair(e) )</td>
<td>( \log\left(\frac{\sum_{(q_i, q_j) \in \Omega} C(q_i, q_j, e)}{</td>
</tr>
</tbody>
</table>

Table 8.1: A Summary of the Term Features

**Temporal Features** \( \mathcal{Y}(e_i, Q) \). 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 different characteristics of the temporal distribution of a term, we adopt the kurtosis defined as \( \frac{\mu_4}{\mu_2^2} \), where \( \mu \) is the mean and \( \mu_j \) is the \( j \)-th central moment. Kurtosis were originally proposed by Jones and Diaz [73] to capture the dynamics of such a time series. It can be used to quantify the probability distribution concentrated in peaks of a time series – i.e., the “peakedness” of the time series. Here, we measure this peakedness for both a single candidate expansion term \( e \) (\( KURT_1 \)), and the combination of a candidate expansion term \( e \) with a term \( q_i \) from the original query (\( KURT_{12} \)). In addition to this, we are interested in knowing about the randomness of terms over time. According to Jones and Diaz [73], bursty events in a time series normally contribute to a high autocorrelation value. To capture this, we compute the first order autocorrelation of a time series for both a single candidate expansion term \( e \) (\( AC_1 \)), and the combination of a candidate expansion term \( e \) with a term \( q_i \) from the original query (\( AC_{12} \)). Finally to measure the temporal similarity between the time series of two terms \( q_i \) and \( e \), we can apply the cross-correlation measure (\( CC \)), mainly used in Statistics to measure the relationship between two random variables [32, 131]. To summarize, the five relevant features are listed in Table 8.2.

**Spatial Features** \( \mathcal{Z}(e_i, Q) \). As explained in Section 8.1, the concept of event is also strongly related to the spatial dimension. Recall that an event can be defined by
8.4. Spatial Distribution of Tags

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$KURT(e)$</td>
<td>The kurtosis value of the time series for the documents annotated with an expansion term $e$</td>
</tr>
<tr>
<td>$KURT(Q+e)$</td>
<td>The maximum kurtosis value of the time series for the documents annotated with both a given expansion term $e$ and a query term $q_i \in Q$</td>
</tr>
<tr>
<td>$AC(e)$</td>
<td>The autocorrelation value of the time series for the documents annotated with an expansion term $e$</td>
</tr>
<tr>
<td>$AC(Q+e)$</td>
<td>The maximum autocorrelation value of the time series for the documents annotated with an expansion term $e$ and a query term $q_i \in Q$</td>
</tr>
<tr>
<td>$CC(Q,e)$</td>
<td>The maximum cross-correlation between the time series for the documents annotated with an expansion term $e$ and the time series for the documents annotated with query term $q_i \in Q$</td>
</tr>
</tbody>
</table>

Table 8.2: A Summary of the Temporal Features

“something happening in a time slice and at a certain geographical location”. Having this in mind, our hypothesis is that a good term expansion is spatially correlated with at least one of the query terms. Similar to what we have done with the temporal features, we also study the clustering tendency of the spatial distribution for the pictures annotated with the candidate expansion terms. To compute the spatial features, we use the approach presented in Section 8.4.1. As explained before, for each pair of terms $e$ and $q_i$, we first extract the set of geographical world tiles $T_{q_i,e}$ 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 $T_{q_i,e}$. The first three feature vectors are vectors computed using the relative discrete maximum distance (RDMD) function of the specified tag point pattern (see Equation 8.9), consisting of $\overrightarrow{D}_{RDMD}(e)$, $\overrightarrow{D}_{RDMD}(e + Q)$, $\overrightarrow{D}_{RDMD}(e, Q)$. The second three feature vectors are based on the relative discrete positive area (RDPA) function (see Equation 8.8), consisting of $\overrightarrow{D}_{RDPA}(e)$, $\overrightarrow{D}_{RDPA}(e + Q)$, $\overrightarrow{D}_{RDPA}(e, Q)$. For all the extracted features, we compute the values of these functions, varying the distance values from 0 to 1 km, with a step of 0.1 km. The same operations are done for both the resulting first and second derivative of the $\hat{g}_{RDPA}$ and $\hat{g}_{RDMD}$ functions (see Section 8.4.2). As a result, since each feature vector contains twelve elements, we have a total number of 72 spatial features. To summarize, our spatial features are listed in Table 8.3.
Table 8.3: A Summary of the Spatial Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\vec{D}_{RDMD}(e)$</td>
<td>The vector of the values of $RDMD$ function related to the $D$ function of the spatial point patterns for the pictures annotated with a candidate expansion term $e$</td>
</tr>
<tr>
<td>$\vec{D}_{RDMD}(e+Q)$</td>
<td>The vector of the values of $RDMD$ function related to the $D$ function of the spatial point pattern for the pictures annotated with both $e$ and $q_i \in Q$</td>
</tr>
<tr>
<td>$\vec{D}_{RDMD}(e,Q)$</td>
<td>The vector of the values of $RDMD$ function related to the cross-$D$ function between tag point pattern associated to $e$ and the tag point pattern of $q_i \in Q$</td>
</tr>
<tr>
<td>$\vec{D}_{RDPA}(e)$</td>
<td>The vector of the values of $RDPA$ function related to the $D$ function of the spatial point pattern for the pictures annotated with a candidate expansion term $e$</td>
</tr>
<tr>
<td>$\vec{D}_{RDPA}(e+Q)$</td>
<td>The vector of the values of $RDPA$ function related to the $D$ function of the spatial point pattern for the pictures annotated with both the terms $e$ and $q_i \in Q$</td>
</tr>
<tr>
<td>$\vec{D}_{RDPA}(e,Q)$</td>
<td>The vector of the values of $RDPA$ function related to the cross-$D$ function between tag point pattern associated to $e$ and the tag point pattern of $q_i \in Q$</td>
</tr>
</tbody>
</table>

**Combining the Spatial Features over the World Dataset**

Our dataset was build from a collection of Flickr pictures covering the whole world map. So the spatial distribution of the pictures is not uniform. To cope with this, we divided 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, thus avoiding the Arctic and Antarctic areas. These areas have normally poor photographic activity. Further, the size of each tile related to 1 degree of latitude is always around 111 km, while following the latitude value, the size varies from around 0 at the poles to around 111 km at the equator. In our approach, we consider a total set, $Q = \{Q_1, ..., Q_L\}$, of tiles composed by $L = 50400$ elements. In addition, we will only consider significant tiles containing more than a $\gamma$ (threshold) number of pictures. We refer this set to as $\hat{Q}$. 

To extract the spatial feature vectors for a pair of tag $w_i$ and $w_j$, we must first extract the list of tiles $\{Q_{i1}, ..., Q_{iN}\}$ and $\{Q_{j1}, ..., Q_{jM}\}$ containing pictures tagged with $w_i$ and $w_j$, respectively. We then merge the two lists to find the set of tiles $\hat{Q}_{ij}^\gamma \in \hat{Q}^\gamma$ containing pictures tagged with $w_i$ and those tagged with $w_j$ in the same
tile. For each tile, the tag point pattern for each of the two tags must be extracted to calculate the bivariate D-function and the corresponding estimators, as presented in the previous section.

To have a structure that efficiently support these operations, we index each tile \( Q_l \) as a document composed by the set \( \mathcal{W}_{Q_l} = \{ w_{l_1}, \ldots, w_{Q_l} \} \) of all the tag used to annotate the pictures in the considered tile. Thus, an inverted index \( \mathcal{I} \) is created for each tag \( w \in \mathcal{W}_{Q_l} \). Each of these tags is linked to an inverted list containing the id of the tile and the term frequency of of the tag in the tile:

\[
\mathcal{I} : \{ w_i \rightarrow \{ Q_{i_1}, t f_{Q_{i_1}}(w_i) >, \ldots \} \}_i
\]  
(8.14)

Selection of the Tile for Spatial Features Extraction. As explained before, we use the geographical tiles to compute and extract the spatial features. We denote such tiles \( \mathcal{T}_{Q,e} \) if they contain documents annotated with both \( Q \) and \( e \), where \( Q \) is a query consisting of terms \( q_i, i = 1, \ldots, n \), and \( e \) is a candidate expansion term \( e \). To make the spatial features suitable for our classifier, we must select only one tile that is most representative for a specific input query \( Q \). We call this the best tile. To do this, we first run \( Q \) on our dataset. Thereafter, we select the first \( K \) geotagged pictures from the resulting ranked list. Finally, we select the tile containing the highest number of occurrences, weighted with the TF-IDF-based ranking score. For simplicity, we treat each tile as a document, and then index and search them using Solr search platform. Thus the resulting ranked list is a list of tiles with Lucene scores\(^7\).

Query Re-weighting Process
With the features being presented, we now explain how we perform the re-weighting process using our sets of features. Figure 8.4 shows a part of this process. As depicted in this figure, the Temporal Classifier is learned with positive and negative examples using only term and temporal features. Further, the Spatiotemporal Classifier is trained with instances using the complete set of features. So, given a query term \( q_i \) and the candidate expansion term \( e \), we first extract the complete set of features. Then, the input instance are classified with both of the classifiers. Finally, a Selector module selects the best confidence value as follows: If some of the elements (features) of the vector of spatial features are missing, then the confidence score from the Temporal Classifier is selected. Otherwise, the Selector chooses the confidence score from the Spatiotemporal Classifier. To incorporate the final confidence score \( Conf(+|e) \) for good candidate expansion terms into the KL score for the query expansion, we mix the values in a linear combination as follows:

\[
KL_{\text{Final}}(e) = \alpha KL(e) + (1 - \alpha)Conf(+|e).
\]  
(8.15)

\(^7\)http://lucene.apache.org/core/3_6_2/scoring.html
The confidence value $\text{Conf}(+|e)$ is calculated from the idea that both the classifiers give their contribution exploring the terms over different dimensions and are not one the extension of the other one but are complementary. For that reason the confidence value is calculated simply by merging the contributions of the two classifiers as follows:

$$
\text{Conf} = \begin{cases} 
0 & \text{Conf}_T < 0.5 \text{ AND } \text{Conf}_ST < 0.5 \\
\text{Conf}_T & \text{Conf}_T > 0.5 \text{ AND } \text{Conf}_ST < 0.5 \\
\text{Conf}_{ST} & \text{Conf}_T < 0.5 \text{ AND } \text{Conf}_ST > 0.5 \\
\frac{\text{Conf}_T + \text{Conf}_{ST}}{2} & \text{Conf}_T > 0.5 \text{ AND } \text{Conf}_ST > 0.5 
\end{cases} 
\quad (8.16)
$$

Here $\text{Conf}_T(+|e)$ is the confidence values from the Temporal Classifier and $\text{Conf}_{ST}(+|e)$ is the confidence value from the Spatiotemporal Classifier.

![Diagram](image)

Figure 8.4: Good expansion term selection process through classification

Note that the vectors of spatial features may be empty or contain elements with null values. This happens if there does not exist geotagged documents that are annotated with both $e$ and $q_i \in Q$. In such a case, the functions $g_{RDMD}(Q + e)$ and $g_{RDPA}(Q + e)$ cannot be estimated and the classification would be performed with a query feature vector containing missing instances. We want to avoid such a situation. This is the reason we implemented a classifier selector, as shown in Figure 8.4.
8.5 Experimental Setup

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

8.5.1 Dataset

To perform our experiments for 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 covering a time period from 01.01.2006 to 31.12.2010 and without spatial restrictions. As a result, our final dataset consists of 88.257.485 pictures, of which 18.861.585 pictures are without any tags and around 23.5% with 1 to 3 tags. For relevance judgement we apply the well-established Upcoming dataset [14] used previously in other relevant works [176]. Thus, this data set serves as our ground truth. Specifically, the Upcoming dataset consists of 270.425 pictures from Flickr, taken between 1st of January 2006 and 31st December 2008, each of which belongs to a specific event from the Upcoming event database. 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. And, 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\(^8\). As part of the dataset preparation, we perform a preprocessing step consisting of tokenization – i.e., UTFTokenizion based on whitespace and punctuation marks, and English stopword removal.

8.5.2 Evaluation Methodology

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

Input Query Set

We randomly selected set of 150 pictures, one for each event cluster in the Upcoming dataset and use the tags annotating them, as queries. We divide this set of queries into two subsets; one consisting of 100 queries that we use train and to evaluate the

\(^8\)See http://www.terrier.org/
performance of the classifiers, and the remaining 50 queries are used as the test set, to evaluate the retrieval effectiveness of the proposed retrieval framework.

**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 existing retrieval models from the state-of-the-art, including the Vector Space Model (VSM), Okapi BM25 (BM25) [140], 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 set 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 (ML) approach with the baseline features as proposed Lin et al. [89] 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 MAP values of KL. We compare the baseline approaches with our proposed methods, first by comparing them with a query expansion framework with a classifier learned with the combination of documents and temporal features (KL\_T); and then a framework with a classifier learned with the combination of documents, temporal, and spatial features (KL\_ST). Note that in addition to the above models, we also experimented with the Mixture Model [188] and the Relevance Model [84], also incorporating the feedback documents in the ranking score computation. However, the results from these experiments were, though comparable, were worse than those from the BM25+KL query expansion models. Thus, for simplicity we did not include the results from these experiments in this chapter.

In stead, to have a fair comparison with similar approaches we implemented the geo-temporal tag relatedness by Zhang et al. [189]. In this work the authors proposed specifying tag similarity by comparing their temporal and geographical distributions. First, they quantize the world into \( m \) tiles of 1 degree and the time in \( n \) bin of two weeks. Then, they represent each tag by a geo-temporal signature – i.e., a \( m \times n \) vector, with normalized counts of unique users tagging a picture within the geo-temporal bin. Finally, they define the geo-temporal similarity \( rel_{geo-temp} \) of two tags results as the euclidean distance between the two corresponding signatures.

So to incorporate this relatedness in a retrieval framework and compare it with our approach, we define a ranking equation as a linear function between the geo-temporal similarity from [189] and the KL score given by \( (\alpha \times KL + (1 - \alpha) \times rel_{geo-temp}) \). We tune and select the best value of the parameter \( \alpha \) over a set of 50 queries.

**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 [9]. 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 chapter are all statistically significant, unless otherwise specified.

8.6 Results

In this section we perform two different analyses. First, we study the usefulness of using the classifier trained with our temporal and spatial features in a retrieval setting. Second, we investigate the effectiveness of using the temporal and spatial features in a classifier with an optimal feature selection procedure.

8.6.1 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. So to do this, we evaluated several existing classifiers with respect to their classification accuracy, and then selected one having the best accuracy. Specifically, we tested our method using Naive Bayes classifier (NB), Support Vector Machine (SVM), C4.5 decision tree (C4.5) and the Random Forest (RF) ensemble classifier. We used Weka machine learning toolkit to test the classifiers [60]. The training set was composed by a set of 1000 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 we obtained by selecting a positive and a negative class using different values of threshold \( \lambda \) (see also Section 8.4.3). The \( \lambda \) values we selected were \{0.005, 0.001, 0.05, 0.01, 0.5\}.

We summarize the averaged results in Table 8.4. 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 the RF classifier, with an accuracy of around 95\%. The precision values for the classification of the good terms was 93\% and the recall value were as high as 97.56\%.
Table 8.4: Classification performance using different classifiers. Best scores within each column are type-set boldface.

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>+</td>
</tr>
<tr>
<td>NB</td>
<td>59.12</td>
<td>0.6102</td>
</tr>
<tr>
<td>SVM</td>
<td>69.22</td>
<td>0.6802</td>
</tr>
<tr>
<td>C4.5</td>
<td>91.52</td>
<td>0.8870</td>
</tr>
<tr>
<td>RF</td>
<td>94.98</td>
<td>0.9288</td>
</tr>
</tbody>
</table>

We now analyze the behavior of the accuracy value of the four proposed classifier over the different $\lambda$ values. The results is summarized in Figure 8.5. 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 Naive Bays (NB) and the SVM classification, with high margin.

![Figure 8.5: Accuracy of the four different classifiers over the different values of $\lambda$](image)

Overall, our experiments showed that tree-based classification approaches work best for our application, with which RF is the best classifier. For this reason, we use RF as the base classifier for our framework. In the next section, we compare the classification performance using different sets of features. In addition, we perform a features analysis to find out which features set contributes to the best performance in the end.
8.6. Results

8.6.2 Retrieval Effectiveness Comparison

To perform this experiment, as well as executing the standard model (BM25), we also applied the Query Expansion models described in the previous section (KL_T and KL_ST). More specifically, we used the the Rocchio’s framework weighting model, with both the Kullback-Leibler divergence model (KL) to choose the expansion terms. For each query expansion run, we used the default values – i.e., setting $\beta = 0.4$ and choosing the first $n$ terms of the top-$K$ documents for the Rocchio’s Beta weighting model. The values of $K$ – i.e., the number of pseudo relevant documents, were chosen from \{20, 40, 60, 80, 100, 120\}, and $n$ – i.e., the number of selected terms, from \{15, 25, 35, 45, 55\}. Finally we perform the query expansion baseline model based on the geo-temporal tag similarities as proposed in [189]. For simplicity, we refer this approach to as ZKYC[189].

The results are summarized in Table 8.5. We can observe that the baseline query expansion method always outperforms the baseline BM25, with a maximum MAP improvement of 6.2%. We can also see that ranking the feedback tags by using KL and the ZKYC[189] method for selection of query expansion terms does not significantly improve the effectiveness of the simple KL ranking score for query expansion. This is mainly because it captures the feature of a tag distribution on a single temporal and geographical scale – i.e, due to the size of the geo-temporal bin.

In contrast, both of our proposed query expansion methods outperform both the baseline BM25 and the baseline query expansion model, for all the combinations of number of documents and number of terms considered. Further, for KL_T the maximum improvement of the MAP value is 9.7%, while for KL_ST the maximum improvement is 12.4%. From studying these MAP values, we also conclude that the geo-temporal tag relatedness proposed in [189] does not produce better retrieval performance. The main reason for this is that with the method of Zhang et al. [189] – i.e., the ZKYC[189], terms do not seem to have sufficient capability of ”attracting” good expansion terms. This is because the geo-temporal attractiveness of two terms is on a fixed scale, only. In contrast, our method is able to catch this geo-temporal attractiveness property at different geo-temporal scales.

In Figure 8.6, we show the maximum gain of MAP from the BM25 as function of the different numbers ($k$) of top-$k$ documents. As we can observe, all the three 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 then this benefit decreases. In contrast, with both of our two approaches, KL_T and KL_ST, this breakpoint is much higher than with the baseline KL. Specifically, the breakpoints are around 80 and 40 documents for our approaches and the baseline approach, respectively. The reason for this is
related to the fact that with baseline KL, the set of candidate expansion terms are explored by considering only document features, which seems to be too restrictive.
8.6. Results

Our methods, on the other hand, use temporal and/or spatial features to select the best candidate terms.

![Figure 8.6: Maximum gain of MAP from the baseline according to different top-k document used in the QE process](image)

8.6.3 Analysis of the Features

In this section we analyze the effectiveness our temporal and spatial features in learning our classifier to select the expansion terms. The question we want to answer is do our purposed features 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) [181], Gain Ratio (GR) [48] and symmetrical uncertainly (SU) [184]. IG is given by $\text{IG}(C, F) = \mathcal{H}(C) - \mathcal{H}(C|F)$, 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 $F$. GR is the direct extension of IG, and is given as follows: $\text{GR}(C, F) = IG(C, F)/\mathcal{H}(C)$. SU evaluates the goodness of a subset of features $F$ by comparing its symmetrical uncertainty with another subset of features [184]. Let $F_{\text{Sub1}} \subset F$ and $F_{\text{Sub2}} \subset F$ such two subsets. Then,

$$SU(F_{\text{Sub1}}, F_{\text{Sub2}}) = \frac{IG(F_{\text{Sub1}}, F_{\text{Sub2}})}{\mathcal{H}(F_{\text{Sub1}}) + \mathcal{H}(F_{\text{Sub2}})}$$ (8.17)

As before, we use the Weka machine learning toolkit to implement of the feature selection methods.

Table 8.6, 8.7 and 8.8 report the IG, GR and SU scores, respectively, for the features we used in our classification of good and bad expansion terms. It shows which
features are the best using the baseline and temporal features compared with applying baseline, temporal and spatial features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>IG Score</th>
<th>Feature</th>
<th>IG Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>coOccSingleWhole</td>
<td>0.104</td>
<td>AC2</td>
<td>0.204</td>
</tr>
<tr>
<td>CC</td>
<td>0.066</td>
<td>RDM12[2]</td>
<td>0.097</td>
</tr>
<tr>
<td>KURT12</td>
<td>0.065</td>
<td>RDM12[3]</td>
<td>0.088</td>
</tr>
<tr>
<td>AC12</td>
<td>0.046</td>
<td>RDM12[3]</td>
<td>0.086</td>
</tr>
<tr>
<td>DF3Feedback</td>
<td>0.035</td>
<td>RDM12[1]</td>
<td>0.086</td>
</tr>
<tr>
<td>coOccSingleFeedback</td>
<td>0.034</td>
<td>RDM12[3]</td>
<td>0.086</td>
</tr>
<tr>
<td>coOccPairFeedback</td>
<td>0.031</td>
<td>RDM12First[3]</td>
<td>0.080</td>
</tr>
<tr>
<td>DF0Feedback</td>
<td>0.029</td>
<td>RDM12[1]</td>
<td>0.074</td>
</tr>
<tr>
<td>DF1Feedback</td>
<td>0.025</td>
<td>DF3Feedback</td>
<td>0.064</td>
</tr>
<tr>
<td>DF2Feedback</td>
<td>0.025</td>
<td>RDM12Second[1]</td>
<td>0.063</td>
</tr>
<tr>
<td>DF2Whole</td>
<td>0.021</td>
<td>RDM12First[1]</td>
<td>0.062</td>
</tr>
<tr>
<td>DF0Whole</td>
<td>0.021</td>
<td>RDM12[2]</td>
<td>0.062</td>
</tr>
<tr>
<td>DF1Whole</td>
<td>0.021</td>
<td>RDM12[4]</td>
<td>0.056</td>
</tr>
<tr>
<td>DF3Whole</td>
<td>0.021</td>
<td>KURT2</td>
<td>0.048</td>
</tr>
<tr>
<td>coOccPairWhole</td>
<td>0.021</td>
<td>coOccSingleWhole</td>
<td>0.048</td>
</tr>
<tr>
<td>KURT1</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC1</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 8.6: Comparison of the feature quality based on Information Gain.

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 seems to not 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 between the time series of the candidate expansion term and the query term – i.e., CC.

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 $\overrightarrow{D}_{RDMD}(Q,e)$, $\overrightarrow{D}_{RDMD}(Q+e)$, $\overrightarrow{D}_{RDM}(Q,e)$ and $\overrightarrow{D}_{RDM}(Q+e)$. This means that the features related to the spatial distributions of the documents annotated with both the candi-
Table 8.7: Comparison of the feature quality based on Gain Ration.

date expansion terms and the query terms, and the spatial correlation between the two tag point patterns have a strong impact on classification. As a conclusion, our analysis 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.

8.7 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 proposed 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 these geo-spatial features. As we have shown in this chapter, 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. Our experiments, comparison against the baseline methods, and extensive analyses have demonstrated the effectiveness of our approach. These have particularly shown the importance of our proposed spatial features and the feasibility of our approach.

<table>
<thead>
<tr>
<th>Baseline+Temporal</th>
<th>Baseline+Spatial+Temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>SU Score</td>
</tr>
<tr>
<td>$KURT12$</td>
<td>0.080</td>
</tr>
<tr>
<td>$coOccSingle_W$</td>
<td>0.073</td>
</tr>
<tr>
<td>$AC12$</td>
<td>0.060</td>
</tr>
<tr>
<td>$CC$</td>
<td>0.059</td>
</tr>
<tr>
<td>$coOccSingle_F$</td>
<td>0.048</td>
</tr>
<tr>
<td>$DF3_F$</td>
<td>0.037</td>
</tr>
<tr>
<td>$DF0_F$</td>
<td>0.034</td>
</tr>
<tr>
<td>$coOccPair_F$</td>
<td>0.031</td>
</tr>
<tr>
<td>$DF1_F$</td>
<td>0.029</td>
</tr>
<tr>
<td>$DF2_F$</td>
<td>0.029</td>
</tr>
<tr>
<td>$DF2_W$</td>
<td>0.023</td>
</tr>
<tr>
<td>$DF0_W$</td>
<td>0.023</td>
</tr>
<tr>
<td>$DF1_W$</td>
<td>0.023</td>
</tr>
<tr>
<td>$DF3_W$</td>
<td>0.023</td>
</tr>
<tr>
<td>$coOccPair_W$</td>
<td>0.023</td>
</tr>
<tr>
<td>$KURT1$</td>
<td>0.000</td>
</tr>
<tr>
<td>$AC1$</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 8.8: Comparison of the feature quality based on Symmetrical Uncertainty.
Part IV

Conclusion and Future Work
Chapter 9

Conclusion

The main focus of this thesis has been on searching and mining events from collections of pictures. In the methods that has been presented, the information surrounding the pictures has been used, such as timestamp, textual tags and geographical information as the base for the searching and mining approach. Methods for extracting groups of images related to events have been proposed. First, a mining algorithm is developed here that leverages on the geographical, temporal and textual data surrounding pictures – i.e., tags. Then, for the same purpose, external knowledge databases such DBPedia, last.fm and Upcoming are used to improve the event identification. Next, using existing information retrieval models as starting points, methods have been proposed to improve the effectiveness of event search systems. This has been done by incorporating temporal and geographical raw data for both free non-timestamped and timestamped textual query. As part of this, a new approach has been developed to represent the geographical data by extending spatial statistics analysis concepts. This has also enabled the analysis of the spatial distribution of tags.

With this in mind, the following summarizes the main contributions of this thesis.

9.1 Contributions of the Thesis

The main contributions of this thesis are specifically related to extraction, search and mining of events from media collections. First, it has developed frameworks for the automatic extraction of groups of pictures representing events. Then, it has proposed methods that improve the effectiveness of searching events from collections of Web-images. Specifically, the main contributions of this work can be summarized as follows.
9.1.1 Detection of Event-Related Resources

A novel algorithm is proposed for the extraction and clustering of event-related images. To be able to develop this method, it was necessary to establish a definition of event in the context of timestamped and geo-annotated images within media-sharing applications. The proposed algorithm has extended the well-known clustering algorithm STC to deal with both the textual annotations, and time and locational information. Previously, the STC algorithm was used for text document clustering only. Further, inspired by the well-known DBSCAN method, a spatiotemporal based merging function is proposed here for refining the set of extracted clusters. The behavior of the algorithm with respect to different time and space granularity is analyzed, and its performance is evaluated with two different Flickr datasets. To prove the validity of the presented algorithm, the effectiveness of the present approach is compared with existing similar methods.

Further, an event detection framework is proposed that is able to extract groups of images related to specific events from a dataset of timestamped and partially geo-annotated pictures. As part of this, the effectiveness of using external knowledge sources is demonstrated for improving the query expansion process. The effectiveness of the framework is proven by submitting the experimental results in a competitive challenge by comparing the present method with other methods with similar purposes.

9.1.2 Spatiotemporal Analysis of Tag Point Pattern

The concept of exploratory analysis and spatial statistics in the domain of media-sharing applications is extended to improve the effectiveness of searching images related to events. In particular, the concept of the tag point pattern is proposed. This has managed to model the spatial distributions of tag terms and this is used to define two estimators for measuring and testing the clustering tendency of the tag spatial point patterns. In addition, it is shown how to employ these estimators for extracting tags related to geographical hotspots or landmarks in a specific geographical area. The results of extensive experiments have shown the effectiveness of the proposed estimators. The comparative analyses have also revealed the ability of the two estimators to capture spatial point patterns with different characteristics. Finally, applying sub-sampling techniques, has proven that the present method is feasible for both small and large datasets.
9.1.3 Tag-based search of event-related pictures

Still focusing on event search systems, an information retrieval (IR) framework is proposed to improve the retrieval performance using a machine learning-based method, assuming the query is a free text, in combination with temporal and geographical raw data surrounding the pictures.

An IR framework has been developed that has improved existing query expansion models for textual and timestamped queries, by incorporating temporal and geographical proximity clues. Further, the scalability of this approach has been proven through extensive experiments. These experiments have also shown the effectiveness of this method with short queries.

Finally, analytic work has compared to compare the effectiveness of different retrieval models first when using only textual metadata. In addition, it has analyzed how different combinations of textual fields affect the retrieval effectiveness, depending on the adopted retrieval model.

9.2 Answers to research questions

The following elaborates on how the research questions presented in Section 1.3 have been answered:

RQ1: Can raw geographical and temporal information surrounding resources improve the clustering and detection of resources geotagged and timestamped dataset?

Considering a dataset of timestamped and geotagged images, annotated by users, Chapter 4 has presented a method for clustering and detecting groups of images representing event. This is performed by extending the clustering methods previously performed on textual documents. The limitation of the applicability of this method might be caused by the lack of precision of the geographical annotations associated with the pictures.

RQ2: Can a raw geographical and temporal term profile improve existing retrieval and query expansion models for the task of tag-based search of event-related resources?

Chapter 8 proposes query expansion framework that incorporates geographical and temporal profile for each candidate expansion term. The extensive experiments have shown an improvement in the retrieval effectiveness compared to state-of-the-art retrieval models.
RQ3: How can raw geographical and temporal data be modeled and cooperate to improve and extend the retrieval performance of a tag-based search of event-related resources?

Referring to the answer of the RQ2, in Chapter 8, has shown, through spatial statistics tools how to model the temporal and geographical profile of terms. As part of this, a method is proposed to measure the similarity between two terms, incorporating a geographical and temporal profile, and a machine learning-based query expansion framework.

RQ4: Can a geographical and temporal tag profile improve tag co-occurrence and tag similarity?

To be able to define the clustering tendency of picture tags, Chapter 8 has proposed a robust set of geographical features that can be used for the analysis of the geographical structure of the spatial distribution of the tags. Thereafter, by using tools from a well-founded exploratory analysis within statistics, a method is derived to explore the correlation between two tags by analyzing the correlation between their geographical distributions. Moreover, as stated in the answer of RQ3, this similarity is combined with existing methods for measuring the correlation between temporal patterns of terms, for searching purpose.

RQ5: How can a geographical and temporal term profile be used to infer semantics to the tags?

Chapter 7 has defined a set of estimators that can be used to indicate the clustering tendency based on a given spatial distribution of terms. It has shown how the clustering tendency can be used to implement a method that is able to extract and detect terms related to geographical point of interest.

9.3 Future Work

This section presents the possible future work related to the methods proposed in the present work.

Extraction of Event Clusters In the proposed approach for extracting groups of images related to the same event, it was intended to analyze the possibility of extracting a short description summarizing each group of events. Different sources of information have been explored such as the description metadata and comments related to each picture in media-sharing applications. Nevertheless, the approach can still be improved. Therefore, as part of the future work, we will investigate the use of the description metadata, as for example in [177, 63], to extract more informative annotations from the extracted event clusters. Moreover we will explore
9.3. Future Work

the possibility to include this metadata in the construction of the suffix tree to analyze the trade-off between size of the tree and clustering performance. Further, we will study how techniques from query expansion such as those described in [154, 113, 89] can be used in retrieving relevant event clusters. Finally, we will further explore the possibility to exploit the incremental characteristic of the STC algorithm so that it can be used in online applications. In the same approach, further work will be done in the direction of employing a variant of the suffix tree – i.e., distributed suffix tree. This will improve the scalability for larger datasets. Existing methods [35] are based on a construction algorithm for the subtrees of a suffix tree in a linear time. The constructed suffix tree would tackle the memory bottleneck problem by constructing the subtrees in parallel.

Searching event-related images  We also plan further work related to the proposed approaches for searching related events. First, searching of events from a textual and timestamped query images, there are still interesting results and aspects of this work that we have omitted, but will be part of the future research. More specifically, we will investigate the effects of including semantic similarities among terms by using and linking to knowledge bases, such as Wikipedia, in term re-weighting. We will also investigate the possibility of integrating features from (Web-based social) user interactions to further improve the retrieval performance. Second, focusing on search of events from textual query, to further explore the usefulness of the proposed spatial features in more general information retrieval settings, we are plan to investigate using our approach on other resources than pictures – e.g., geographical data from Foursquare and geotagged tweets from Twitter. Finally, we will investigate the possibility of extending the proposed set of classifiers with a metadata classifier to combine the heterogeneous features and comparison with a learning to rank approach as proposed in [89]. We will further consider the possibility to apply and extend the same retrieval models, leveraging on temporal and spatial profile of terms in more general retrieval datasets such as the test collections provided from Text Retrieval Conference (TREC) and Cross Language Evaluation Forum (CLEF).
Bibliography


and knowledge management, CIKM ’09, pages 523–532, New York, NY, USA, 2009. ACM.


[75] J. Kamps, C. Monz, M. Rijke, and B. Sigurbjörnsson. Language-dependent and language-independent approaches to cross-lingual text retrieval. In C. Pe-


20th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR ’97, pages 267–276, New York, NY, USA, 1997. ACM.


[161] K. Sparck Jones. Document retrieval systems. chapter A statistical interpreta-


