

# Exploring Image, Text and Geographic Evidences in ImageCLEF 2007

João Magalhães<sup>1</sup>, Simon Overell<sup>1</sup>, Stefan Rieger<sup>1,2</sup>

<sup>1</sup>Department of Computing  
Imperial College London  
South Kensington Campus  
London SW7 2AZ, UK

<sup>2</sup>Knowledge Media Institute  
The Open University  
Walton Hall  
Milton Keynes MK7 6AA, UK

([j.magalhaes@imperial.ac.uk](mailto:j.magalhaes@imperial.ac.uk), [simon.overell01@imperial.ac.uk](mailto:simon.overell01@imperial.ac.uk), [s.rueger@open.ac.uk](mailto:s.rueger@open.ac.uk))

## Abstract

This year, ImageCLEF2007 data provided multiple evidences that can be explored in many different ways. In this paper we describe an information retrieval framework that combines image, text and geographic data. Text analysis implements the vector space model based on non-geographic terms. Geographic analysis implements a placename disambiguation method and placenames are indexed by their Getty TGN Unique Id. Image analysis implements a query by semantic example model. The paper concludes with an analysis of our results. Finally we identify the weaknesses in our approach and ways in which the system could be optimised and improved.

## Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries; H.2.3 [Database Management]: Languages—Query Languages

## General Terms

Measurement, Performance, Experimentation

## Keywords

Semantic Image Retrieval, Geographic Image Retrieval

## 1 Introduction

This paper presents a system that integrates visual, text and geographic data. Such systems are in great demand as the richness of metadata information increase together with the size of multimedia collections. We evaluated the system on the ImageCLEF Photo IAPR TC-12 photographic collection to assess some of the evidence combination strategies and other design aspects of our system. As the system implemented a set of preliminary algorithms we were able to clearly see the different impacts of each component of our system.

This paper is organized as follows: section 2 discusses the dataset characteristics; sections 3 to 6 present our system: the text data processing algorithms, the geographic data processing algorithms, the image data processing algorithms, and the combination strategies. Results are presented in section 7 and finally section 8 presents the conclusions.

## 2 ImageCLEF Photo Data

ImageCLEF Photo dataset is ideal to test our system: it includes metadata information (both textual descriptions and geographic information), and visual information. The dataset has 20,000 images with the corresponding metadata. The photos vary in quality, levels of noise, and illustrate several concepts, actions or events. Metadata enriches the images by adding information such as the fact that a street is in some location or the profession of one of the persons in the photos. A more thorough description of the dataset can be found in [4].

The goal of this dataset is to simulate a scenario where collections have heterogeneous sources of data and users submit textual queries together with visual examples. This is similar to TRECVID search task with a slight

difference concerning geographic data and actions that are only possible to detect on videos (e.g. walking/running).

	DOCNO	annotations/00/60.eng
	TITLE	Palma
	NOTES	The main shopping street in Paraguay
	LOCATION	Asunción, Paraguay
	DATE	March 2002
	IMAGE	images/00/60.jpg
	THUMBNAIL	thumbnails/00/60.jpg

**Table 1 – Example of metadata information available on the collection.**

### 3 System

The implemented system has two separate indexes for images related data and another one for metadata related data. Next we will describe how information is analysed and stored on both indexes.

#### 3.1 Metadata Indexes

The indexing stage of Forostar begins by extracting named entities from text using ANNIE, the Information Extraction engine bundled with GATE. GATE is Sheffield University's General Architecture for Text Engineering. Of the series of tasks ANNIE is able to perform, the only one we use is named entity recognition. We consider ANNIE a “black box” where text goes in, and categorised named entities are returned; because of this, we will not discuss the workings of ANNIE further here but rather refer you to the GATE manual [2].

##### 3.1.1 Named Entity Fields

We index all the named entities categorised by GATE in a “Named Entity” field in Lucene (e.g. “Police,” “City Council,” or “President Clinton”). The named entities tagged as Locations by ANNIE we index as “Named Entity – Location” (e.g. “Los Angeles,” “Scotland” or “California”) and as a Geographic Location (described in Section 3.1.3). The body of the GeoCLEF articles and the article titles are indexed as text fields. This process is described in the next section.

##### 3.1.2 Text Fields

Text fields are pre-processed by a customised analyser similar to Lucene’s default analyser [1]. Text is split at white space into tokens, the tokens are then converted to lower case, stop words discarded and stemmed with the “Snowball Stemmer”. The processed tokens are held in Lucene’s inverted index.

##### 3.1.3 Geographic Fields

The locations tagged by the named entity recogniser are passed to the disambiguation system. We have implemented a simple disambiguation method based on heuristic rules. For each placename being classified we build a list of candidate locations, if the placename being classified is followed by a referent location this can often cut down the candidate locations enough to make the placename unambiguous. If the placename is not followed by a referent location or is still ambiguous we disambiguate it as the most commonly occurring location with that name.

Topological relationships between locations are looked up in the Getty Thesaurus of Geographical Names (TGN) [5]. Statistics on how commonly different placenames refer to different locations and a set of synonyms for each location are harvested from our Geographic Co-occurrence model, which in turn is built by crawling Wikipedia [13].

Once placenames have been mapped to unique locations in the TGN, they need to be converted into Geographic fields to be stored in Lucene. We store locations in two fields:

- **Coordinates.** The coordinate field is simply the latitude and longitude as read from the TGN.
- **Unique strings.** The unique string is the unique id of this location, preceded with the unique id of all the parent locations, separated with slashes. Thus the unique string for the location “London, UK” is the unique id for London (7011781), preceded by its parent, Greater London (7008136), preceded by

its parent, Britain (7002445). . . until the root location, the World (1000000) is reached. Giving the unique string for London as 1000000\1000003\7008591\7002445\7008136\7011781.

Note the text, named entity and geographic fields are not orthogonal. This has the effect of multiplying the impact of terms occurring in multiple fields. For example if the term “London” appears in text, the token “london” will be indexed in the text field. “London” will be recognised by ANNIE as a Named Entity and tagged as a location (and indexed as Location Entity, “London”). The Location Entity will then be disambiguated as location “7011781” and corresponding geographic fields will be added.

Previous experiments conducted on the GeoCLEF data set in [11] showed improved results from having overlapping fields. We concluded from these experiments that the increased weighting given to locations caused these improvements.

## 3.2 Images Indexes

The image indexing part of our system creates high-level semantic indexing units that allow the user to access the visual content with query-by-keyword or query-by-semantic-example. In our ImageCLEF experiments we only used query by semantic example.

Following the approach proposed in [9], each keyword corresponds to a statistical model that represents that keyword in terms of the visual features of the images. These keyword models are then used to index images with the probability of observing the keyword on each particular image. Next we will describe the different steps of the visual analysis algorithm, see [9] for details.

### 3.2.1 Visual Features

Three different low-level features are used in our implementation: marginal HSV distribution moments, a 12 dimensional colour feature that captures the histogram of 4 central moments of each colour component distribution; Gabor texture, a 16 dimensional texture feature that captures the frequency response (mean and variance) of a bank of filters at different scales and orientations; and Tamura texture, a 3 dimensional texture feature composed by measures of image’s coarseness, contrast and directionality. We tiled the images in 3 by 3 parts before extracting the low-level features. This has two advantages: it adds some locality information and it greatly increases the amount of data used.

### 3.2.2 Feature Data Representation

We create a visual vocabulary where each term corresponds to a set of homogenous visual characteristics (colour and texture features). Since we are going to use a feature space to represent all images, we need a set of visual terms that is able to represent them. Thus, we need to check which visual characteristics are more common in the dataset. For example, if there are a lot of images with a wide range of blue tones we require a larger number of visual terms representing the different blue tones. This draws on the idea that to learn a good high-dimensional visual vocabulary we would benefit from examining the entire dataset to look for the most common set of colour and texture features.

We build the high-dimensional visual vocabulary by clustering the entire dataset and representing each term as a cluster. We follow the approach presented in [8], where the entire dataset is clustered with a hierarchical EM algorithm using a Gaussian mixture model. This approach generates a hierarchy of cluster models that corresponds to a hierarchy of vocabularies with a different number of terms. The ideal number of clusters is selected via the MDL criterion.

### 3.2.3 Maximum Entropy Model

Maximum entropy (or logistic regression) is a statistical tool that has been applied to a great variety of fields, e.g. natural language processing, text classification, image annotation. Thus, each keyword  $w_i$  is represented by a maximum entropy model,

$$p(w_i | V) = \text{MaxEnt}(\beta^{w_i} F(V)),$$

where  $F(V)$  is the feature data representation defined on the previous section of visual feature vector  $V$ , and  $\beta^{w_i}$  is the vector of the regression coefficients for keyword  $w_i$ .

We implemented the binomial model, where one class is always modelled relatively to all other classes, and not a multinomial distribution, which would impose a model that does not reflect the reality of the problem: the

multinomial model implies that events are exclusive, whereas in our problem keywords are not exclusive. For this reason, the binomial model is a better choice as documents can have more than one keyword assigned.

### 3.2.4 Images Indexing by Keyword

ImageCLEF data is not annotated with keywords, thus we used a different dataset. This dataset was compiled by Duygulu et al. [3] from a set of COREL Stock Photo CDs. The dataset has some visually similar keywords (jet, plane, Boeing), and some keywords have a limited number of examples (10 or less). Each image is annotated with 1-5 keywords from a vocabulary of 371 keywords of which we modelled 179 keywords to annotate ImageCLEF images.

## 4 Query Processing

The previous sections described how the dataset information is processed and stored. This section will describe how the user query is processed and matched to the indexed documents. Similarly to the documents processing the user’s query is divided into its text and image elements. Figure 1 illustrates the query processing and how multiple evidences are combined.

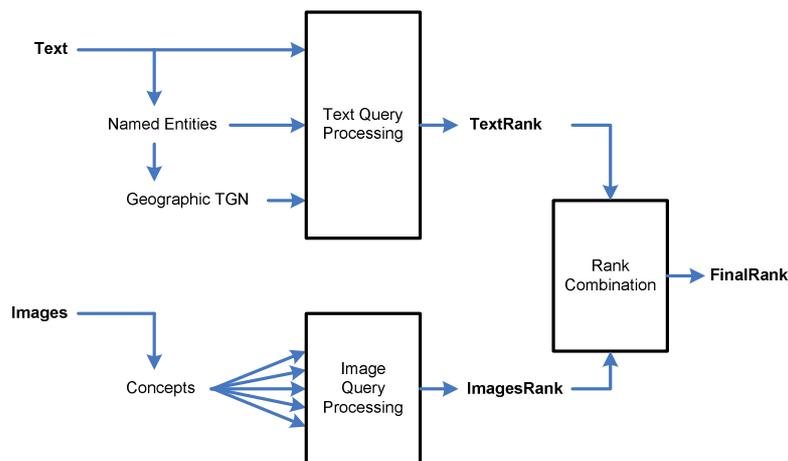


Figure 1 – Query processing and evidence combination.

The textual part is processed in the same way as at indexing time and a combination strategy fuses the results from the different textual part processing (text terms, named entities and nameplaces). Each image is also processed in the same way as previously described and a combination strategy fuses the results from the images. Thus, only the documents-query similarity and the evidence combination is new in the query processing part.

### 4.1 Text Query Processing

The querying stage is a two step process: (1) manually constructed queries are expanded and converted into Lucene’s bespoke querying language; (2) then we query the Lucene index with these expanded queries.

#### 4.1.1 Manually Constructed Query

The queries are manually constructed in a similar structure to the Lucene index. Queries have the following parts: a text field, a Named Entity field and a location field. The text field contains the query with no alteration. The named entity field contains a list of named entities referred to in the query (manually extracted). The location field contains a list of location – relationship pairs. These are the locations contained in the query and their to the location being searched for. A location can be specified either with a placename (optionally disambiguated with a referent placename), a bounding box, a bounding circle (centre and radius), or a geographic feature type (such as “lake” or “city”). A relationship can either be “exact match,” “contained in (vertical topology),” “contained in (geographic area),” or “same parent (vertical topology)”. The negation of relationships can also be expressed i.e. “excluding,” “outside,” etc.

We believe such a manually constructed query could be automated with relative ease in a similar fashion to the processing that documents go through when indexed. This was not implemented due to time constraints.

### 4.1.2 Expanding the Geographic Query

The geographic queries are expanded in a pipe-line. The location – relation pairs are expanded in turn. The relation governs at which stage the location enters the pipeline. At each stage in the pipeline the geographic query is added to. At the first stage an exact match for this location’s unique string is added: for “London” this would be 1000000\1000003\7008591\7002445\7008136\7011781. Then places within the location are added, this is done using Lucene’s wild-card character notation: for locations in “London” this becomes 1000000\1000003\7008591\7002445\7008136\7011781\*. Then places sharing the same parent location are added, again using Lucene’s wild-card character notation. For “London” this becomes all places within “Greater London,” 1000000\1000003 \7008591\7002445\7008136\*. Finally the coordinates of all the locations falling close to this location are added. A closeness value can manually be set in the location field, however default values are based on feature type (default values were chosen by the authors). The feature of “London” is “Administrative Capital,” the default value of closeness for this feature is 100km. See [12] for further discussion on the handling of geographic queries.

### 4.1.3 Combining using the VSM

A Lucene query is built using the text fields, named entity fields and expanded geographic fields. The text field is processed by the same analyzer as at query time and compared to both the notes and title fields in the Lucene index. We define a separate boost factor for each field. These boost values were set by the authors during initial iterative tests (they are comparable to similar weighting in past GeoCLEF papers [10] and [14]). The title had a boost of 10, the notes a boost of 7, named entities a boost of 5, geographic unique string a boost of 5 and geographic co-ordinates a boost of 3. The geographic, text and named entity relevance are then combined using Lucene’s Vector Space Model.

## 4.2 Image Query Processing

In automatic retrieval systems, processing time is a pressing feature that directly impacts the usability of the system. We envisage a responsive system that processes a query and retrieves results within 1 second per user, meaning that to support multiple users it must be much less than 1 second. Figure 2 presents the architecture of the system. We implemented a query by semantic example algorithm [7] that is divided into three parts:

- **Semantic Multimedia Analyser:** The semantic multimedia analyser infers the keywords probabilities and is designed to work in less than 100ms. Another important issue is that it should also support a large number of keywords so that the semantic space can accommodate the semantic understanding that the user gives to the query. Section 3.2 presented the semantic multimedia analyser used in this paper, see [9] for details.

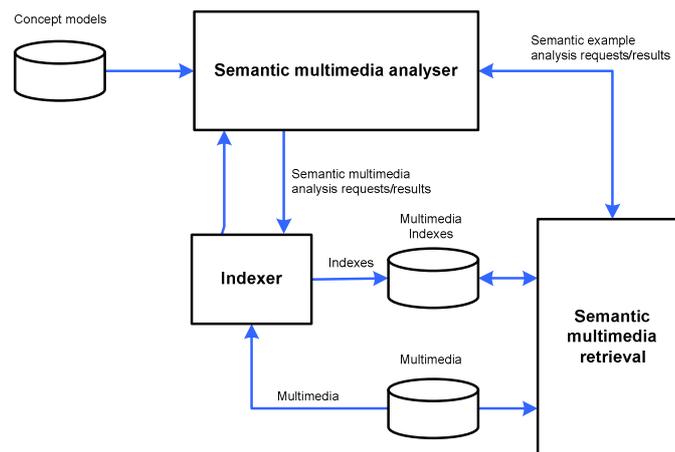


Figure 2 – Query by semantic example system.

- **Indexer:** Indexer uses a simple storage mechanism capable of storing and providing easy access to each keyword of a given multimedia document. It is not optimised for time complexity. The same indexing mechanisms used for content based image retrieval can be used to index content by semantics.

- **Semantic Multimedia Retrieval:** The final part of the system is in charge of retrieving the documents that are semantically close to the given query. First it must run the semantic multimedia analyser on the example to obtain the keyword vector of the query. Then it searches the database for the relevant documents according to a semantic similarity metric on the semantic space of keywords. In this part of the system we are only concerned with studying functions that mirror human understanding of semantic similarity. See [7] for details.

### 4.2.1 Semantic Space

In the semantic space multimedia documents are represented as a feature vector of the probabilities of the  $T$  keywords (179 in our case),

$$\vec{d} = [d_{w_1}, \dots, d_{w_T}],$$

where each dimension is the probability of keyword  $w_i$  being present on that document. Note that the vector of keywords is normalised if the similarity metric needs so (normalisation is dependent on the metric). These keywords are extracted by the semantic-multimedia-analyser algorithm described in Section 3.2.

It is important that the semantic space accommodates as many keywords as possible to be sure that the user idea is represented in that space without losing any concepts. Thus, systems that extract a limited number of keywords are less appropriate. This design requirement pushes us to the research area of metrics on high-dimensional spaces.

We use the tf-idf vector space model. Each document is represented as a vector  $\vec{d}$ , where each dimension corresponds to the frequency of a given term (keyword)  $w_i$  from a vocabulary of  $T$  terms (keywords). The only difference between our formulation and the traditional vector space model is that we use  $P(w_i | d)$  instead of the classic term frequency  $TF(w_i | d)$ . This is equivalent because all documents are represented by a high-dimensional vocabulary of length  $T$  and

$$P(w_i | d) \cdot T \approx TF(w_i | d).$$

Thus, to implement a vector space model we set each dimension  $i$  of a document vector as

$$d_i = P(w_i | d) \cdot IDF(w_i).$$

The inverse document frequency is defined as the logarithm of the inverse of the probability of a keyword over the entire collection  $D$ ,

$$IDF(w_i) = -\log(P(w_i | D)).$$

### 4.2.2 Semantic Similarity Metric

Documents  $\vec{d}$  and queries  $\vec{q}$  are represented by vectors of keyword probabilities that are computed as was explained in the previous section. Several distance metrics exist in the tf-idf representation that compute the similarity between a document  $\vec{d}_j$  vector and a query vector  $\vec{q}$ . We rank documents by their similarity to the query image according to the cosine-distance metric. The cosine similarity metric expression is:

$$\text{sim}(\vec{q}, \vec{d}) = 1 - \frac{\sum_{i=1}^T q_i d_i}{\sqrt{\sum_{i=1}^T (q_i)^2} \cdot \sqrt{\sum_{i=1}^T (d_i)^2}}.$$

### 4.2.3 Multiple Images Query Combination

The semantic similarity metric gives us the distance between a single image query and the documents in the database. There are two major strategies of combining multiple examples of a query: (1) merging the examples into a single query input and produce a single rank; (2) submit several queries and combine the ranks. Obviously each of these two types of combinations uses different algorithms. For ImageCLEF2007 we implemented a simple and straight forward combination strategy: we submit one query for each example and combine the similarity values from all individual queries:

$$\text{sim}(\vec{q}_1, \vec{q}_2, \vec{q}_3, D) = \left\{ \text{sim}(\vec{q}_1, D), \text{sim}(\vec{q}_2, D), \text{sim}(\vec{q}_3, D) \right\}.$$

This is an OR operation while an AND operation would be achieved with a query vector that is the product of all individual query image vectors. We return the top 1000 results.

### 4.3 Rank Combination

The text query and the image query are processed independently and are later combined by a simple linear combination. Previous work on this area has found a set of good weights to combine text and image ranks. Applying these weights we reach the expression that combines ranks:

$$CombinedRank_i = 0.375 \cdot (1000 - ImageRankPos_i) + 0.675 \cdot (1000 - TextGeoRankPos_i).$$

The different metric spaces hold different similarity functions, thus producing incompatible numerical measures. Thus, we used the document rank position (e.g.  $ImageRankPos_i$ ) to compute the final rank. The produced metric gives the importance of each document  $i$  for the given query. This linear combination only considers the top 1000 documents of each rank. Documents beyond this position are not considered.

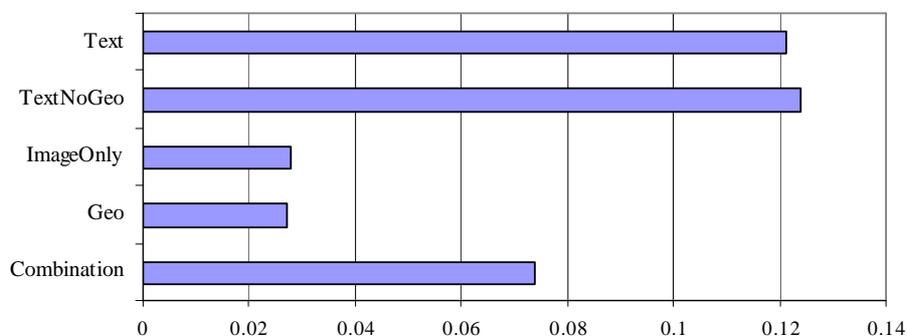
## 5 Results and Discussion

We ran 5 experiments:

- **Text:** The Text part of the query only;
- **TextNoGeo:** The Text part of the query with geographic elements removed;
- **ImageOnly:** The Image part of the query only;
- **Geo:** The geographic part of the image only; and
- **Combination:** A combination of the Text, ImageOnly and Geo runs.

Note the TextNoGeo, ImageOnly and Geo runs are orthogonal. Results are presented on Figure 3.

We can see that TextNoGeo results achieved the best results. Next was the Text results, however, there is not a significant difference (using the Wilcoxon Signed Rank Test [6]). This was a bit surprising as one would expect to improve results when you add location information to the query and to the documents. We believe that the decrease in performance was due to the fact that some queries use the geographic part as inclusive or exclusive.



**Figure 3 – Retrieval results for the different types of analysis.**

The Geo results were statistically significantly worse than all the other results with a confidence of 99.98%. This is due to minimal information being used (generally a list of placenames). Only 26 of the 60 queries contained geographic references. Across these geographic queries the Geo method achieved an MAP of 0.062 compared to 0.085 for TextNoGeo and 0.025 for ImageOnly. In fact across these 26 queries there is no significant difference between Text, TextNoGeo, Combination and Geo methods. This shows that (for the geographic queries) the geographic component of the query is extremely important.

Image results also achieved very low results, which given the scope of the evaluation might seem a bit surprising. This is related to the uses of the images to illustrate keywords that are not obvious. For example from Figure 4 we can see that in some cases it is very difficult to guess the query or how images should be combined.



Figure 4 – Image examples for query “people in San Francisco”.

Summing up all these problems that we faced on single data-type evidences, together with unbalanced combinations of the different types of evidences, we found that the final rank is almost an average of individual ranks.

## 6 Conclusions and Future Work

Most of our work was done on the documents analysis and indexing part of the evaluation. However, it became evident that our single combination strategy does not cover all possible types of queries. In some cases images should be combined with AND, others with OR operations. The same happened with text and geographic, e.g. locations can be inclusive (“in San Francisco”) or exclusive (“outside Australia”). Moreover, some images only illustrate part of the query (“people in San Francisco”, Figure 4) and it is obviously difficult to identify correct results with only the visual part of the query.

All these lessons show that it is essential to make good use of the different algorithms by combining them properly according to the query text. Moreover, the query analysis must produce an accurate logical combination of the different entities of the query to achieve a good retrieval performance. In our future work we would like to repeat the experiments described in this paper using a combination strategy based on the logical structure of the query.

## 7 References

- [1] "Apache Lucene Project," 2006.
- [2] H. Cunningham, "GATE, a General Architecture for Text Engineering," *Computers and the Humanities*, vol. 36, pp. 223-254, May 2004 2004.
- [3] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *European Conf. on Computer Vision*, Copenhagen, Denmark, 2002, pp. 97-112.
- [4] M. Grubinger, P. Clough, HanburyAllan, and H. Müller, "Overview of the ImageCLEF 2007 Photographic Retrieval Task," in *Working Notes of the 2007 CLEF Workshop*, Budapest, Hungary, 2007.
- [5] P. Harping, *User's Guide to the TGN Data Releases. The Getty Vocabulary Program 2.0 edition*, 200.
- [6] D. Hull, "Using statistical testing in the evaluation of retrieval experiments," in *ACM SIGIR Conference*, Pittsburgh, PA, USA, 1993, pp. 329--338.
- [7] J. Magalhães, S. Overell, and S. Rüger, "A semantic vector space for query by image example," in *ACM SIGIR Conf. on research and development in information retrieval, Multimedia Information Retrieval Workshop*, Amsterdam, The Netherlands, 2007.
- [8] J. Magalhães and S. Rüger, "Logistic regression of generic codebooks for semantic image retrieval," in *Int'l Conf. on Image and Video Retrieval*, Phoenix, AZ, USA, 2006.
- [9] J. Magalhães and S. Rüger, "Information-theoretic semantic multimedia indexing," in *ACM Conference on Image and Video Retrieval* Amsterdam, The Netherlands, 2007.
- [10] B. Martins, N. Cardoso, M. Chaves, L. Andrade, and M. Silva, "The University of Lisbon at GeoCLEF 2006," in *Working Notes for the CLEF Workshop*, Alicante, Spain, 2006.
- [11] S. Overell, J. Magalhães, and S. Rüger, "Forostar: A system for GIR," in *Lecture Notes from the Cross Language Evaluation Forum 2006*, 2006.

- [12] S. Overell, J. Magalhães, and S. Rüger, "GIR experiments with Forostar at GeoCLEF 2007," in *ImageCLEF 2007*, Budapest, Hungary, 2007.
- [13] S. Overell and S. Rüger, "Geographic co-occurrence as a tool for GIR," in *CIKM Workshop on Geographic Information Retrieval*, Lisbon, Portugal, 2007.
- [14] M. E. Ruiz, S. Shapiro, J. Abbas, S. B. Southwick, and D. Mark, "UB at GeoCLEF 2006," in *In Working Notes for the CLEF Workshop*, 2006.