

# Gazetiki: Automatic Creation of a Geographical Gazetteer

Adrian Popescu  
CEA LIST

18 Route du Panorama  
92260 Fontenay aux Roses, France  
+33146548013

adrian.popescu@cea.fr

Gregory Grefenstette  
CEA LIST

18 Route du Panorama  
92260 Fontenay aux Roses, France  
+33146549617

gregory.grefenstette@cea.fr

Pierre-Alain Moëllic  
CEA LIST

18 Route du Panorama  
92260 Fontenay aux Roses, France  
+33146549617

pierre-alain.moellic@cea.fr

## ABSTRACT

Geolocalized databases are becoming necessary in a wide variety of application domains. Thus far, the creation of such databases has been a costly, manual process. This drawback has stimulated interest in automating their construction, for example, by mining geographical information from the Web. Here we present and evaluate a new automated technique for creating and enriching a geographical gazetteer, called *Gazetiki*. Our technique merges disparate information from Wikipedia, Panoramio, and web search engines in order to identify geographical names, categorize these names, find their geographical coordinates and rank them. The information produced in *Gazetiki* enhances and complements the Geonames database, using a similar domain model. We show that our method provides a richer structure and an improved coverage compared to another known attempt at automatically building a geographic database and, where possible, we compare our *Gazetiki* to Geonames.

## Categories and Subject Descriptors

### H.3.1 Content Analysis and Indexing

### General Terms

Algorithms, Experimentation.

### Keywords

Geographic gazetteer, thesaurus, information extraction, data mining, Wikipedia, Panoramio.

## 1. INTRODUCTION

The construction of large scale geographic databases such as Alexandria [6] or Geonames [5] has been a long, arduous process and the automation of this process is of great interest. Hill [5] defined the three minimal elements for each location in such a geographical gazetteer: the name of the location, its coordinates and its parent category (for example, *mountain*). Recently [10] described a way of automatically building databases containing

two of these elements: geographic names and their associated coordinates, by extracting geographic information from a large set of tags introduced by Flickr users. Automating the creation of more complete geographic resources, with all three elements, remains a challenging research problem.

In this paper, we describe a methodology to automatically create a geographical thesaurus, called *Gazetiki*, which mines multiple sources of information to create a more complete gazetteer. Our approach bootstraps new geolocated items from the contents of Wikipedia pages describing geo-referenced entities and from Panoramio [9], a platform dedicated to the sharing of georeferenced images; probes Wikipedia content and Alltheweb snippets in order to classify the geographical names found; finally, ranks each discovered geographic name using a relevance measure based on its popularity in Panoramio and in Alltheweb. We present one application using our method, and compare it to an existing geo-referencing search system.

The remainder of this paper is structured as follows: section 2 reviews related work; section 3, details our method for automatically creating a geographical thesaurus; section 4 discusses the relation between our *Gazetiki* and Geonames; the final section 5 presents experiments for validating our approach.

## 2. RELATED WORK

Classical approaches directed towards the automatic organization of conceptual hierarchies using free text include [12] and [4]. Their results suffer from two important drawbacks: the number of automatically mined concepts is small and the quality of the extracted relations is inferior to manually structured resources.

[10] was one of the first attempts to discover geographic names from large scale unstructured multimedia data, exploring multiscale burst analysis to separate geographical locations from others tags found in georeferenced Flickr pictures. Their reported precision of 82% (with 50% recall) using a completely automatic analysis with no linguistic filtering of the resulting data. [1] exploited the results in [10] and associated a relevance value to each discovered entity. The resulting structure was used in an application for geographic image retrieval, with representative tags overlaid on a scalable map. Clicking on a tag displayed associated georeferenced images for a visual exploration. The tag displaying system is also open to linkage with other external geographic databases.

A number of recent works exploit Wikipedia information in different application domains. [11] presented a way for automatically structuring conceptual hierarchies based on the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

JCDL '08, June 16–20, 2008, Pittsburgh, Pennsylvania, USA.

Copyright 2008 ACM 978-1-59593-998-2/08/06...\$5.00.

analysis of linguistic patterns. Hyponymic and paronymic relations were extracted, with 60% - 70% of accurate relations.

[13] introduced a method for building and maintaining person names gazetteers exploiting Wikipedia information.

Since articles often begin with a definition, [7] analysed the first sentence in Wikipedia articles in order to categorize the analyzed concept into a hypernymic category. This leads to the correct extraction of the parent concept more than 90% of the times.

[2] proposed a MySQL representation of Wikipedia content, extracting structured information from the tables included in the encyclopaedic pages, without processing the free text in the articles.

Geonames [5] and Alexandria [6] are two manually constituted databases, whose elements are organized using two types of linguistic relations: *isA* (or *conceptual heritage*) and *partOf* (or *spatial inclusion*). The *isA* relation connects terms from general to particular; *partOf* specifies the spatial coverage of a geographic name and its inclusion relation with respect to other names in the gazetteer. The Geonames database [5] includes 8 main categories (such as *inhabited-locality*, *physical landmark*, *body of water*, *transportation axis*, etc.), 645 intermediary ones (e.g.: *mountain*, *lake* or *museum*) and more than 6 million specific named entries (e.g. *Mont Blanc*, *Baikal*, *Louvre*). Spatial inclusion (*partOf*) allows one to find that the *Louvre* is situated in *Paris*, which, is a part of *France*, part of *Europe*. In addition to these two types of relations, Geonames also contains precise localisation coordinates, alternative names of the object, etc. For a majority of the database entries, Geonames does not include any information which would allow sorting such as the population of a *populated place*, or the height of *mountains*, or length of *rivers*, etc. One can see that such information would be useful in presenting retrieval results.

In [8], the authors described NameSet, a system meant to assign names to geographic coordinates. Their techniques however are only applicable to general terms, i.e. large city names like *San Francisco*.

There is also a community based effort to build a geographical database, Wikimapia [14], now exceeding 6 millions entries. The approach taken in Wikimapia has two important drawbacks: many items are not proper to the geographic domain and the location names are only very partially categorized into larger geographic types. In addition, there is no easy access offered to the database for downloading.

Our own work on the automatic construction of geographic databases, presented here, is related to [10], but based on a linguistic analysis of Web documents rather than statistical approach. Another noticeable difference arises from the fact that we structure geographical information using Wikipedia and Panoramio and a text search engine, whereas [10] exploits Flickr information. Similarly to [7], we mine parts of the encyclopaedic pages in order to categorize discovered items, but we incorporate other Web sources beyond Wikipedia which they do not. [2] also mines information from the collaborative encyclopaedia, but since we limit our approach to a particular domain, we perform more specific and complex text analysis in order to discover specific geographical knowledge.

## 3. AUTOMATIC CONSTRUCTION OF GAZETIKI

### 3.1 Problem Statement

How can we merge information from various unstructured or semi-structured sources to obtain valid pieces of knowledge, respecting the minimal conditions for an entry in a geographic gazetteer [6], that is, each location is described by at least three elements (*Entity-Name*, *Entity-Coordinates*, *Entity-Type*)?

This problem poses several challenges: building large scale databases without human intervention; balancing accuracy and coverage; finding freely available information; merging and ranking data from different sources, different formats. We attempt a series of answers to all these problems in the following subsections.

### 3.2 Data Sources

Other than a few large, manually built, but incomplete, gazetteers like Alexandria or Geonames, no other Web sources provide our minimum explicit tuples (*Entity-Name*, *Entity-Coordinates*, *Entity-Type*) for geographic names. The individual pieces can nonetheless be found in different sources.

- **Wikipedia** contains much gazetteer-type information. There are around 100000 georeferenced articles in the English version of Wikipedia [5], a number which still remains significantly smaller than the 6 millions entries already present in Geonames. As Wikipedia is continually expanding, we can exploit the HTML page structure in new georeferenced articles, finding the *Entity-Name* (in the page title), alternate variants of the entity name (in the header, and in foreign language page pointers), as well as explicit geographic *Entity-Coordinates* (though there remain problems of different formats: in decimal or in degree, minutes, seconds, as well as ambiguity about the hemisphere). *Entity-Type* can sometimes be extracted from the first sentence of the article [7]. There is no guarantee, however, that all three components of a minimal gazetteer structure will be found for each item. There are many cases of missing *Entity-Coordinates* (e.g. *The Orthodox Cathedral of Timisoara*, in Romania) or *Entity-Type* in articles of geographically located items.
- **Panoramio**, a website for sharing georeferenced pictures, is another source of exploitable information. The validation of photo locations by other users constitutes a useful particularity of this platform compared to other sites, such as Flickr. Data introduced by the users is available via an API which returns: the title and localization of the images, the names of the user who uploaded it and a link towards the image itself. We will be able to extract new geographic *Entity-Name*, from these titles, with the *Entity-Coordinates* of the entity, which may or may not appear in Wikipedia. We can also exploit frequency information to extract a fourth element, *Entity-Rank*, which we will add to Gazetiki as an indication of the popularity or importance of the geographic name (see section 4.5). We exploit linguistic patterns related to the geographical domain to isolate geographic names but a straightforward use of the explicit category in the names is a naïve way to classify entities. An example of error is that of the *Cathedral of Learning*, which is a *skyscraper* and not a

*cathedral*. The categorization of the elements discovered using Panoramio needs to be double checked.

- **Search engines snippets** will supply pieces missing from the above sources, double-checking *Entity-Type* information when missing in Wikipedia, or in Panoramio. We will also use search engine statistics to complete *Entity-Rank* values.

### 3.3 Construction Algorithm

After describing our reuse of the Geonames domain model, we show here how geographic names are extracted from unstructured or semi-structured documents, how these names are categorized for entity-type, how to find entity coordinates, and finally how to rank the popularity of entities.

#### 3.3.1 Reusing the Geonames Domain Model

We use as the basis of our geographical domain model, the one implemented in Geonames which possesses over 600 intermediate geographic concepts for tagging an *Entity-type*. A preliminary study of Wikipedia articles convinced us, however, that this was not sufficient. We thus added about 20 new concepts for the geographic domain (e.g. *borough*, *neighborhood*), as well as some concepts frequently having a strong spatial component, such *club*, *team* or *laboratory* which are often given with geographic coordinates in Wikipedia.

#### 3.3.2 Entity-Name Extraction

We populate Gazetiki by extracting everything we can consider as a localizable entity from Wikipedia and from Panoramio.

From a downloaded copy of Wikipedia, we begin by extracting all articles which contain geographic coordinates. Following [7], we retain this article if its first sentence contains a geographic concept (after the verb ‘to be’), for example, “...is a large residential *neighborhood*”. For these names, we have the three minimal elements described by [6]: the *Entity-Name*, the *Entity-Type*, and the *Entity-Coordinates*.

We use these articles to build a second list of other candidate geographic names from Wikipedia. From these articles, we extract all new linked proper names (identified by uppercase Wikipedia links) and access their articles. We retain these names as candidates if the article’s first sentence contains a geographic concept, as described above. These candidates have an *Entity-Name*, and an *Entity-Type*, but no *Entity-Coordinates*, yet.

To find additional geographical names, we exploit Panoramio in this way. Starting at any point, we download the photos there and analyze their titles. We extract candidate entity names from these photo titles by including all capitalized terms to the left and to the right of a geographical concept (e.g., *Museum*). The process stops when an article (*the*, *a*, *an*), a punctuation sign or a non capitalized term is encountered (except for the words *of*, *and* or *for* after the concept, which are retained when followed by a capitalized term, e.g., *of Science*). In this fashion, we extract *Carnegie Museum of Science and Natural History* from a photo title such as: “View of the Carnegie Museum of Science and Natural History from the top of the Cathedral of Learning”. *Cathedral of Learning* is equally extracted from the same title. These candidates have an *Entity-Name*, a potential *Entity-Type*, and *Entity-Coordinates*.

The following steps in the algorithm will fill in missing pieces for all the location and candidate entities, and possibly change their parent concepts.

#### 3.3.3 Entity-Type Categorization

Once we have a candidate geographical entity, we want to find its entity type and coordinates, if missing.

A naïve way to categorize geographic entities is to use the category appearing in the name itself. This method, however, wrongly categorizes instances like *Cathedral of Learning* in Pittsburgh, *Madison Square Garden* in New York or *Palace of Fine Arts* in San Francisco, which are not *cathedrals*, nor *gardens*, nor *palaces*, but rather a *skyscraper*, a *venue* and a *museum*.

As shown above, for candidates from Wikipedia, we use the first sentence in the article to determine *Entity-Type*. We will later call this categorization procedure WIKICAT and its exit values are either the name of a concept in the geographic vocabulary or *null* if no geographic concept was found in the first sentence.

For Panoramio candidates, we employ the text of the snippets extracted from a search engine answers, similar to approaches used in question answering techniques [3]. This technique, later called SNIPPETS, is described figure 1.

```

Given candidate
tempCat = explicitCat(candidate)
Launch Alltheweb query with candidate
Collect first 50 snippets
explicitFreq=SnipFreq(tempCat)
Find the concept in GeoVocabulary
with the maximum maxFreq=SnipFreq(concept)
If (maxFreq greaterThan explicitFreq)
  Find def1 = pageCount("candidate IS A concept")
  Find def2 = pageCount("candidate IS A
tempCat")
If (def1 greaterThan def2)
  finalCat(candidate) = concept
Else
  finalCat(candidate) = tempCat

```

**Figure 1. Pseudo-code representation of the snippets based categorization. *explicitCat* – the term in the geographic vocabulary appearing in the candidate name; *SnipFreq* – function returning the frequency of each element in the geographic vocabulary based on the information in the snippets; *maxFreq* – the maximum value of *SnipFreq* over the geographic vocabulary; *pageCount* – the number of answers of a search engine for definitional queries like “*Y is a (an) Y*”; *def1*, *def2* – temporary variables for storing the values of *pageCount*.**

The default parent concept of a candidate geographic name is the one appearing in its name. If this candidate is associated more frequently with another geographic concept in the snippets associated to the candidate name, we launch definitional queries (“*candidate IS A concept*”) on the Web for both the explicit category and the most frequent geographic concept appearing in the snippets. WE retain as the final *Entity-Type* of the candidate name the one appearing more often in the definitional query.

### 3.3.4 Entity-Coordinates Discovery

For Wikipedia candidate names not having coordinates, we could use the coordinates of the georeferenced articles they came from, but these can be too imprecise. Instead, we search the names in Panoramio and take the average coordinates for photos bearing that name. The function described in figure 2 will be later called LOCALIZATION.

```

coordinates(candidate) = null
If candidate has images in Panoramio
  For each image of candidate
    Put latitude(image) in LatArray
    Put longitude(image) in LongArray
lat = average(LatArray)
long = average (LongArray)
coordinates(candidate) = (lat,long)

```

**Figure 2. Pseudo-code representation of the Panoramio based localization of items.** *image* – Panoramio image returned using *candidate* as a query; *latitude(image)* – latitude associated with *image*; *longitude(image)* – longitude associated with *image*; *LatArray* – array containing all *latitude(image)*; *LongArray* – array containing all *longitude(image)*; *lat* – final latitude of candidate; *long* – final longitude of *coordinates(candidate)* – final coordinates of candidate.

We also use algorithm described in figure 2 for candidate names drawn from Panoramio titles. This simple method produces an approximate localization of the geographic name that, as shown below in the Evaluation section, is satisfying.

### 3.3.5 Entity Ranking

In addition to the minimal information defined by [6], the tuple (*Entity-Name*, *Entity-Coordinated*, *Entity-Type*), we decided that each entry should also include a notion of *Entity-Rank* in order increase the usefulness of the geographic gazetteer in information retrieval. We produce a ranking by aggregating information found in two complementary data sources, Panoramio and a search engine. While the first is well adapted to the geographic domain, its coverage is reduced when compared to that of a search engine such as Alltheweb, which provides better coverage but lower accuracy.

When using Panoramio for ranking items, we take all the pictures around the candidate coordinates (within a distance of 30 kilometers), and search those photos described using the candidate *Entity-Name*. We then combine the total number of retrieved Panoramio images (a classical term frequency measure) with the number of different users (a community-based relevance assessment) having uploaded those images, as shown in the next section. We can reasonably suppose that, if a geographic object was photographed by several people, it is more representative than an object photographed by one person only, and should thus be ranked higher.

The Panoramio based relevance measure is generally accurate but can return the same rank for different items. This happens because the Panoramio corpus is not large enough to produce unique rankings for all discovered geographic names. This is especially true for candidates which are drawn from a small number of photo titles. To differentiate elements having the same Panoramio rank, we use Alltheweb counts for the candidate geographic name.

The implementation of the ranking method (later referred as RANKING) is detailed in figure 3:

```

Given candidate and coordinates
Get panoTF(candidate, coordinates) from Panoramio
Get diffUsers(candidate, coordinates) from Panoramio
panoRank = panoTF(candidate) x diffUsers(candidate)
webRank = pageCount(candidate)
finalRank(candidate) = (panoRank, webRank)

```

**Figure 3. Pseudo-code representation of the ranking function.** We noted: *panoTF* – the candidate frequency in Panoramio; *diffUsers* – the number of different users having uploaded photo of the candidate; *panoRank* – the relevance of the candidate computed using Panoramio; *webRank* – the relevance of the candidate computed using the Web; *finalRank* – the final relevance measure, combining *panoRank* and *webRank*.

The exit value of the function is a pair composed of both *panoRank* and *webRank*, but the last value is only used by Gazetiki when two compared geographic names have equal Panoramio ranks.

## 3.4 Pseudo-code Representation of the Algorithm

### 3.4.1 Aggregated view of the Algorithm

Hereafter, we suppose that we extracted all candidate names from both Wikipedia and Panoramio and we must now decide if they will be included in Gazetiki. In figure 4, we present the pseudo-code of the algorithm used for automatically building a geographic gazetteer.

```

For each candidate in Wikipedia
  coord(candidate) = null
  finalCat(candidate) = WIKICAT(candidate)
  If Wikicoord in article
    coord(candidate) = Wikicoord;
  Else If (finalCat(candidate) != null)
    coord(candidate) = LOCALIZATION(candidate);
  If(coord(candidate) != null and finalCat(candidate) != null
  and pageCount(candidate) greaterThan 15)
    RANKING(candidate);
  Put (candidate, coord(candidate), finalCat(candidate),
    RANKING(candidate) in GAZETIKI;

Foreach candidate in Panoramio
  coord(candidate) = LOCALIZATION(candidate);
  finalCat(candidate) = SNIPPETS(candidate);
  If(coord(candidate) != null and finalCat(candidate) != null)
    RANKING(candidate);
  If( pageCount(candidate) greaterThan 15)
    Put (candidate, coord(candidate), finalCat(candidate),
      RANKING(candidate) in GAZETIKI

```

**Figure 4. Pseudo-code representation of the method designed for the automatic construction of a geographic gazetteer.**

The algorithm takes as entry a candidate name either from Wikipedia or from Panoramio and, if both *Entity-Coordinates* and *Entity-Type* are determined, the Entity-Ranking is equally

calculated. Finally, the (*Entity-Name*, *Entity-Coordinates*, *Entity-Type*, *Entity-Rank*) tuple is included in the geographical thesaurus. Elements whose *Entity-Name* appears in less than 15 pages using Alltheweb as a search engine were not included in Gazetiki, since these were often typographical errors.

#### 4. RESULTS AND EVALUATION

We decided to evaluate Gazetiki against an automatically created database (TagMaps) and a manually built thesaurus (Geonames). To do this, we first manually selected 15 cities from different countries (see Table 1) for which we extracted geographic names using the methodology described in Section 3. The countries were selected to provide a variable quality of the representation in TagMaps and Geonames. The comparison of Gazetiki to TagMaps focuses on the obtained results and not on the exploited data sources. For these cities, we ran our geographical candidate selection algorithms, and generated 6000 entities for Gazetiki, of which the 20% whose *Entity-Names* appeared 15 times or less on the Web were eliminated because of their low frequency of appearance on the Web.

We then evaluated the following:

- the percentage of correctly extracted geographic names;
- given the city regions on a map, we compared the coverage offered by TagMaps and Gazetiki;
- for elements found in both Gazetiki and Geonames, we calculated the precision of our classification procedure using the categorization of geographic names in Geonames as the gold standard;
- for elements common to Gazetiki and Geonames we measured the distance between the coordinates in both databases;

The first test was performed manually because we thought this was the best way to evaluate the precision of the candidate extraction process. The other evaluations were performed automatically. We also illustrate the ranking results obtained in section 3.3.5.

##### 4.1 Instance Extraction

We evaluated the correctness of the geographic names extraction process for 424 elements generated for Gazetiki. For each of the 15 cities, a maximum of 30 randomly extracted items were tested (some cities like *Toulouse* or *Tunis* had less than 30 elements discovered). These elements come from both Wikipedia and Panoramio. In the evaluation procedure, we considered as correct all exact matches of the extracted instances to the real names (i.e. *University of Pittsburgh* or *Squirrel Hill*) and incomplete matches which are commonly equivalent to their longer forms (i.e. *Louvre* instead of *Louvre Museum*).

**Table 1. Evaluation of the extraction process in Gazetiki. Accuracy of automatically placing geographical instances in each city**

City Name	Correct extractions/Out of total entities tested
Athens (Greece)	28/30
Beijing (China)	26/30
Bucharest (Romania)	28/30
Kiev (Ukraine)	27/30
London (UK)	29/30
Moscow (Russia)	27/30
Paris (France)	26/30
Pittsburgh (US)	28/30
San Francisco (US)	28/30
Singapore	29/30
Sydney (Australia)	30/30
Timisoara (Romania)	29/30
Tokyo (Japan)	28/30
Toulouse (France)	7/10
Tunis (Tunisia)	22/24
<b>Overall</b>	394/424
<b>Precision</b>	92.9%

The results in Table 1 show that the geographic names extraction process we propose in this paper is accurate in over 90% of the cases.

The 92.9% precision is to be compared to the 82% precision reported in [10], the only large scale automatically built geographic database we know of. The reader should keep in mind that the data sources exploited in our approach are different from that in [10] and we only compare the final results. We exploit Wikipedia and Panoramio, while the other database is constructed using Flickr data. The Flickr geo-referenced pictures set contains around 30 millions items, whereas Panoramio only includes 5 millions images. In the [10], the authors performed their evaluation after eliminating 50% of their location candidates, starting with the least frequent. Our thresholding of 15 or more web hits only eliminated 20% of our candidates before evaluation.

The errors in our approach are due to some imperfections of our named entities extraction. For example, some common terms, like *Big House*, were mined when using Panoramio and they are reported as errors in this test. A simple solution to eliminate this type of errors would be to eliminate candidates composed of an adjective and an element of the geographic vocabulary. We chose not to use this procedure because it would equally eliminate a term like *White Pagoda* in Beijing, which is a named entity. We also counted as errors vague terms like *Athens Theater*, considering that this term covers several geographic objects and is not a geographic name.

## 4.2 Coverage

We selected a rectangle of approximately 900 km<sup>2</sup> around each evaluated city and compared the total number of geographic names in Gazetiki and in the automatically built TagMaps. We present the results in table 2.

**Table 2. Coverage TagMaps and Gazetiki**

City Name	TagMaps	Gazetiki
Athens (Greece)	20	<b>214</b>
Beijing (China)	64	<b>489</b>
Bucharest (Romania)	27	<b>129</b>
Kiev (Ukraine)	8	<b>145</b>
London (UK)	580	<b>1313</b>
Moscow (Russia)	24	<b>83</b>
Paris (France)	176	<b>321</b>
Pittsburgh (US)	113	<b>413</b>
San Francisco (US)	472	<b>1006</b>
Singapore	46	<b>827</b>
Sydney (Australia)	186	<b>534</b>
Timisoara (Romania)	1	<b>31</b>
Tokyo (Japan)	173	<b>548</b>
Toulouse (France)	<b>18</b>	10
Tunis (Tunisia)	7	<b>24</b>

[10] does not provide detailed information about the obtained coverage and the only way to get information about the level of detail offered by the other automatically built thesaurus was to interrogate the TagMaps Web-service with queries covering the cities used in this evaluation.

The coverage provided in Gazetiki outperforms that in TagMaps (see Table 2) except for Toulouse area, for which the two methods discover a reduced number of results (18 in TagMaps and 10 in Gazetiki). A remarkably high number of results is obtained for cities that are well represented in Wikipedia and in Panoramio. Examples include London (1313), San Francisco (1006) and Singapore (827) and the reader will note that these are all regions in English speaking countries. Tokyo (548) and Beijing (489) are equally well represented, but this is mainly due to the existence of a large number of Panoramio pictures for these cities. The Wikipedia articles for Tokyo and Beijing are not as detailed as those for London or San Francisco.

Somewhat surprisingly, although a major tourist destination, Paris does not appear among the best represented cities. The use of an English geographic vocabulary to extract candidates from Panoramio constitutes an explanation for this situation (and equally for that of Toulouse). The internationalization of the vocabulary will further enrich Gazetiki. Significant differences between the number of candidates in Gazetiki and TagMaps can be signaled for Kiev (145 location names against 8) or Timisoara (31 against 1).

## 4.3 Instance Categorization

Candidate categorization can be automatically evaluated using the intersection of Gazetiki and Geonames for the selected cities. As shown in Table 3, the total number of common elements is 543, with 217 coming from Wikipedia and 326 from Panoramio. We briefly remind the categorization procedure:

- for Wikipedia, the first sentence of the article is considered a definition of the candidate and, if exists, we extract the first occurrence of an element of the geographic vocabulary appearing after the “to be” verb.
- for Panoramio, we use the snippets based categorization procedure described in Subsection 3.3.3.

Given that Wikipedia articles about geographic entities generally contain reliable information, when an element appeared in both sets, the Wikipedia based result was preferred. In table 3, we present the results of the categorization process for the two data sources and their synthesis.

**Table 3. Evaluation of the classification process in Gazetiki**

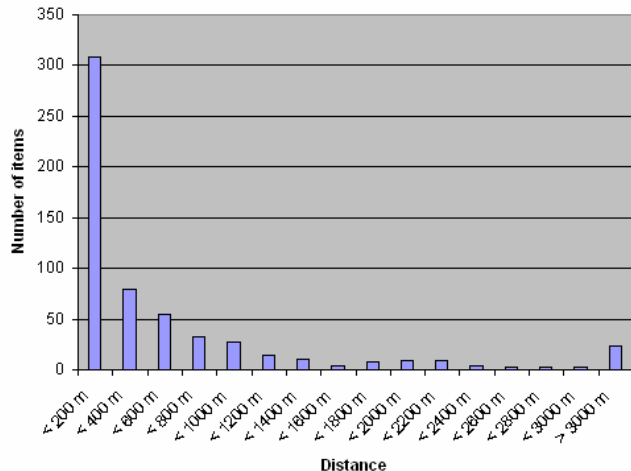
	Wikipedia	Panoramio	Overall
<b>Number of items</b>	217	326	543
<b>Errors</b>	13	32	45
<b>Precision</b>	94%	90%	92%

The results in table 3 indicate a high rate of success of the overall categorization process. The results for the Wikipedia based categorization are consistent with those reported in [7] and the precision is above 90%. The errors that appear are mainly caused by complicated definitions. For example, the verb *to be* is sometimes followed by a reference to the geographic situation rather than a direct reference to the instance type: X is placed in the east of Y and is a Z. Instead of correctly extracting Z, it is possible that an element of the geographic vocabulary appears in Y and the algorithm will wrongly extract this item. In future work, we plan to add a syntactic analysis meant to avoid this type of situation.

In Panoramio, the errors appear when the snippets based categorization fails to find the real parent class of a candidate. The preference given to the Wikipedia based categorization is justified by the results in 3 but the difference between the two classification methods is not considerable.

## 4.4 Instance Localization

The intersection between Geonames and Gazetiki was equally employed so as to assess the distance between common items in the two databases. It is impossible to propose a good binary classification of the distances in acceptable or not and we preferred to present the results using distance ranges. Each bin of Gazetiki results stands for a 200 meters radial sector compared to the coordinates of the item in Geonames. An exception is made for those items where the distance to the baseline is superior to three kilometers. The results are presented in figure 5.



**Figure 5. Distribution of distances between the coordinates of a geographic name in Gazetiki and those in Geonames.**

The results in figure 5 show that the large majority of the coordinates sets for geographic names calculated in Gazetiki are distant of no more than 1 kilometer compared to the corresponding manually supplied coordinates in Geonames. 60% of the distances are inferior to 200 meters; 81% are inferior to 600 meters and 92% are smaller than 1 kilometer. The highest concentration of results is to be found in the first sector of 200 meters around the Geonames coordinates. Object classes whose coordinates are really close to those in Geonames include: *church, school, tower or monument*. The total surface of this type of objects is small and allows a precise spatial localization. In the same time, they often correspond to landmarks and have Wikipedia coordinates or a lot of associated pictures in Panoramio. The mining of coordinates has more chances to be precise when a lot of photos of the objects exist and they depict an object having a small surface.

As for the differences that are superior to 1 kilometer, they usually appear for items having a large surface (*gulf, river, borough or island, university, bay, beach or park*). For these geographic objects, a displacement of the geographic coordinates with a distance of the order of 1 kilometer is comparable to their spatial dimensions and does not greatly affect the quality of the representation. A special case is that of rivers, which are objects with a disproportionate rapport between their length and width. One might correctly place their coordinates at any point along their course.

The differences between the coordinates in the two compared databases are mainly an effect of the fact that the photos of a given geographic object are often taken from a significant distance to the entity (especially for objects that have considerable dimensions). The final coordinates are obtained as an average of individual coordinates of pictures and this averaging works well when the object is photographed from different points of view.

When visualizing tags on a map, the difference between the coordinates of an object in the database and those in reality are partially masked by the fact that the represented text covers a certain area on the map. This text is wider than it is tall, so longitude differences are better masked than those due to latitude imprecision. The differences are more evident when the map is

regarded at high resolution (street or quarter) and become less important for large regions (city, region, country etc.)

## 4.5 Instance Ranking

Our importance ranking of each geographic name was calculated using two reference corpus, Panoramio and Alltheweb, with a preference given to the results obtained using Panoramio. Using these statistics, we present the top 5 “most salient” ranked results as for each city (Table 4).

**Table 4. Term ranking in Gazetiki**

City	Method Using Panoramio & Alltheweb
Athens	Acropolis; Parthenon; Plaka; Olympic Stadium; Temple of Zeus
Beijing	Summer Palace; Temple of Heaven; Tiananmen Square; Lama Temple; Railway Station
Bucharest	Intercontinental; Carol Park; Herestrau Park; Parliament Palace; Stavropoleos
Kiev	South Bridge; Trianon Palace; Rusanivka; Paton's Bridge; Partizan's Victory Park
London	London Eye; Tower Bridge; Trafalgar Square; Buckingham Palace; Hyde Park
Moscow	Red Square; Elk Island; Moscow River; St. Basil's Cathedral; Historical Museum
Paris	Louvre; Eiffel Tower; La Défense; Arc de Triomphe; Montmartre
Pittsburgh	PNC Park; Downtown Pittsburgh; Heinz Field; Cathedral of Learning; Station Square
San Francisco	Golden Gate Bridge; Coit Tower; Oakland; San Francisco Bay; Lombard Street
Singapore	Sentosa; Merlion; Raffles Hotel; Singapore River; Boat Quay
Sydney	Opera House; Harbour Bridge; Darling Harbour; Bondi Beach; Sydney Tower
Timisoara	Bega River; Iulius Mall; Unirii Square; Millenium Church; Timisoara Cathedral;
Tokyo	Tokyo Tower; Rainbow Bridge; Imperial Palace; Kiyosumi Palace; Landmark Tower; Mori Tower
Toulouse	La Grave; Le Canal; Toulouse Cathedral; Toulouse Airport; City Hall
Tunis	St. Louis Cathedral; American Cemetery; Roman Theatre; President Palace; Lookea Beach

The results in Table 4 indicate that the ranking procedure introduced in this paper generally succeeds in ranking best what seems to be the most representative location names for the analyzed cities. If we take the example of Paris, one might wonder why *Notre Dame* does not appear among the first results. The full name of the item in Gazetiki is *Notre Dame de Paris* and this reduces the chances for this item to be found frequently. The same observation stands for *Alcatraz* in San Francisco, which

appears either as *Alcatraz Island* or *Alcatraz Prison*. We are not sure how to correctly identify the preferred short form of these names without introducing ambiguities into our automatic system.

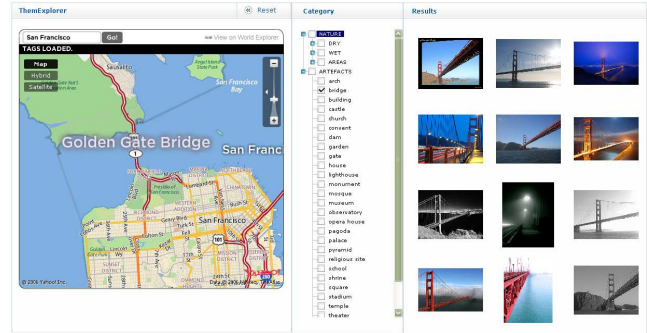
Timisoara is a city for which the number of Panoramio images is significantly smaller than that for Paris or San Francisco. Nevertheless, all top ranked terms correspond to landmarks in the area and this indicates that the ranking method applies well both to well represented regions and to less known areas.

## 5. RELATION TO GEONAMES AND TAGMAPS AND DISCUSSION OF RESULTS

Section 4.2 shows that the richness of the sets of geographic names obtained with our method is not uniform, dependant as it is on the richness of the Wikipedia page dedicated to each particular place and on the volume of Panoramio images for the respective region. Still, the differences between cities in different countries are less pronounced in Gazetiki than in TagMaps and Geonames. For example, Geonames contains only 5 records for Timisoara, while we obtain 30 results in Gazetiki. Geonames is generally richer than our database but a hefty chunk of its content points towards administrative regions (around 50%) and hotels. The intersection between Geonames and Gazetiki, around 15%, shows that the two structures are complementary. Gazetiki is especially useful for regions of the world, like England, Russia, and Romania, in which cases the description of tourist attractions is not yet detailed in existing resources like Geonames. Since our database is built on a similar model to that of Geonames, the two structures can be easily merged.

The comparison of the coverage of the two automatically built gazetteers shows that the Gazetiki is richer than the structure described in [10]. Once again, we remind the reader that we compare the results of our approach and those in [10] and not the data sources or the methodologies. The difference in favor of Gazetiki is due to the fact that the extraction of candidate terms is less selective than in [10]. Our richer content is also accompanied by a higher precision of term extraction, which comes from the approach we adopted: the use of Wikipedia articles and of seed terms from the geographic domain to mine Panoramio titles. The precision of the geographic names extraction process in Gazetiki is superior to that in TagMaps (93%, respectively 82%) while using a stricter definition of what is a correctly determined item than that retained in [10]. In the future, it would be interesting to apply our algorithm to Flickr data in order to obtain a better comparison of the two automatically created geographical databases.

We introduce a simple but efficient entity categorization procedure, a feature that does not exist in [10] or in [1]. The hierarchical organization of Gazetiki allows for our database to be thematically queried. It is possible to propose a more structured exploration of the geographic space than in [1] and allow the users to select interest topics for which they want to see relevant tags and images. For example, one might ask to see which are the bridges or the museums (or both types of objects) in a region of the world. The resulting geo-referenced image retrieval system, which is an improved version of our ThemExplorer prototype (see Fig. 5), combines a spatial and a thematic restriction of the content to be visualized, whereas World Explorer [1] only allows a spatial selection.



**Figure 5. Interface of a geo-referenced image retrieval system allowing a thematic selection of the information to be displayed.**

The results of ranking process, presented in Subsection 4.5, are generally satisfactory. The top ranked discovered elements are well known geographic names, representative for the respective regions. This association of a pertinence value could enhance the current presentation of existing geographical gazetteers, like Geonames, by allowing a presentation of the most interesting elements in a region first. Currently, geographic objects in gazetteers are presented in an unordered fashion and finding a specific object in a richly represented area of the thesaurus is often fastidious.

## 6. CONCLUSION AND FUTURE WORK

We introduced a method for automatically constructing a geographic gazetteer using heterogeneous sources of information available on the Internet. The main contributions of this paper can be summarized as follows:

- An approach to the automatic acquisition of large-scale and fair quality structured data is described in detail. The method is employed for mining geographic information but it can be adapted to other conceptual domains. Two other applications domains we can think of are the extraction of structured information for celebrities and for events.
- Gazetiki is, to our best knowledge, the second attempt (after the structure presented in [10] and [1]) to automatically structure large-scale geographic information. We showed that our structure fulfils the minimal conditions of existence of a geographical gazetteer because a name, a set of coordinates and a parent class are associated to each extracted element. A supplementary dimension was introduced: a relevance score associated to each geographic name. This last dimension is important in information retrieval because it allows the presentation of the most salient elements of a database in priority (especially in map overlays).
- The salience ranking of elements is obtained using a simple measure, not previously described in the literature. The novelty of the ranking comes from the association of frequency information (from Panoramio or from the web) and of a community based popularity score (number of photographers in Panoramio).
- The categorization of candidate names is performed using a method in part known (adapted from [7]) and in part new. The novelty comes from the use of information contained in the snippets on the answers page of a search engine which is

confronted to an existing vocabulary and further validated by launching definitional queries in a search engine.

- The results obtained with our methodology were compared to the results reported for the other automatically built geographic database (described in [10] and [1]). We showed that our method outperforms that in [10] and [1] on two important dimensions, the coverage of the database and the precision of the extraction process. In addition, an *Entity-Type* dimension appears in Gazetiki, while it is not present in [10]. Finally, the geographic names location procedure was evaluated against the content of Geonames and the results show that a majority of items are situated within 200 meters from the Geonames coordinates (with over 90% within 1 kilometer).

The results presented in this paper are promising. The main aspects of our methodology we would like to improve are the segmentation of candidate names and the elimination of some elements that appear more than once, for example Paris museum names written in French and in English. Trying automatic translation of the name or comparing geographic coordinates may help us detect these doubles. For example, if the database contains Musée d'Orsay and Orsay Museum with roughly the same coordinates, one can reasonably suppose that the two names point toward the same entity.

We will also explore the effects of the internationalization of the geographic vocabulary because we observed that a hefty chunk of the textual information in Panoramio is written in languages other than English. We will attempt to use other frequently used languages, like Spanish and French, in order to increase the coverage of Gazetiki.

A third line of work concerns the integration of Gazetiki and Geonames and their exploitation in an improved version of our geographic image retrieval system - ThemExplorer (video available at <http://moromete.net/themexplorer.avi>).

## 7. REFERENCES

- [1] Ahern, S., Naaman, M., Nair, R. and Yang, J. 2007. World Explorer: Visualizing Aggregate Data from Unstructured Text in Geo-Referenced Collections. In *Proc. of JCDL 2007* (Vancouver, Canada, June 2007).
- [2] Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R. and Ives, Z. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proc. of ISWC 2007* (Busan, Korea, November 2007).
- [3] Brill, E., Lin, J., Banko, M., Dumais, S. and Ng, A. Data-intensive question answering. 2001. In *Proc. of the TREC-10 Conference* (Gaithersburg, Maryland, USA, Nov. 2001), 183—189.
- [4] Cimiano, P., Pivk, A., Schmidt-Thieme, L. And Staab, S. 2004. Learning taxonomic relations from heterogeneous evidence. In *Proc. Of ECAI 2004, OLP Workshop* (Valencia, Spain, 2004).
- [5] Geonames - <http://geonames.org>
- [6] Hill, L. L., Frew, J. and Zheng, Q. 1999. Geographic names – the implementation of a gazetteer in a georeferenced digital library. *CNRI D-Lib Magazine* (January, 1999).
- [7] Kazama J. and Torisawa, K. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In *Proc. of EMNLP 07* (Praque, Czech Republic, June 2007).
- [8] Naaman, M. Song, Y. J., Paepcke, A., Garcia-Molina, H. 2007. Assigning Textual Names to Sets of Geographic Coordinates. *Journal of Computers, Environment, and Urban Systems*, 30(4):418-435 (July 2006).
- [9] Panoramio - <http://panoramio.com>
- [10] Rattenbury, T., Good, N., Naaman, M. 2007. Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In *Proc. of SIGIR 2007* (Amsterdam, The Netherlands, July 2007).
- [11] Ruiz-Casado, M., Alfonseca, E., Castells, P. 2007. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data and Knowledge Engineering*, 61(3) (2007).
- [12] Sanderson, M. and Croft, B. 1999. Deriving concept hierarchies from text. In *Proc of SIGIR '99* (Berkeley, CA, USA, August 1999).
- [13] Toral, A. and Munoz, R. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *Proc. of Workshop on NEW TEXT Wikis and blogs and other dynamic text sources* (Trento, Italy, April 2006).
- [14] Wikimapia – <http://wikimapia.org>