Geospatial Web Mining for Emergency Management

Christian Fritz[1], Christian Kirschner[1], Daniel Reker[1], Andre Wisplinghoff[1],
Heiko Paulheim[2], Florian Probst[2]
[1]Technische Universitaet Darmstadt, Karolinenplatz 5, 64289, Darmstadt, Germany
Email: {c_fritz,c_k,d_reker,andre_w}@rbg.informatik.tu-darmstadt.de
[2]SAP Research CEC Darmstadt, Bleichstrasse 8, 64283 Darmstadt, Germany
Email: {heiko.paulheim,f.probst}@sap.com

1. Introduction

Emergency management is a domain where information has to be gathered, aggregated, and visualized dynamically and quickly. By providing the right information at the right time, the chaos phase between the occurrence of a disaster and the start of well-organized relief measures can be significantly shortened (Paulheim et al. 2009).

The information needed in an emergency scenario can be quite diverse. For example, a person planning an evacuation may need to know about companies that can transport people, and places that can serve as emergency shelters. For the first, bus and taxi companies, logistics companies as well as rental car providers may be taken into account. The latter may include hotels and schools as well as sports arenas and concert venues.

Although all this information is available on the web, it cannot be easily accessed. Since such non-trivial categories such as *buildings that can serve as emergency shelter* are not sharply defined, one cannot simply enter *emergency shelter* into Google and retrieve a list of emergency shelters. Instead, lots of subsequent manual searches have to be performed, and the results have to be aggregated by hand. Visual exploration is even more difficult.

While several emergency management tools exist (cf. (Paulheim et al. 2009) for a survey), this concern has not been addressed in this context yet. In this paper, we introduce a prototype which allows for a-priori crawling the web for relevant information on objects belonging to non-trivial categories and provide the aggregated results as an OGC compliant web feature service for visual exploration.

2. Approach

Our approach for gathering information on objects from non-trivial categories from the web consists of five steps (see Figure 1). In a preparation step, a list of sources is manually defined, whereas sources are web catalogues containing a distinct page for each listed object. We identify relevant information by crawling the pages of this predefined list (step 1). The information is extracted (step 2) and processed so that it can be stored in a database (step 3). Finally the information is provided in a standardized format (step 4). We will explain these steps in detail in the following.

We implemented a prototype in Java that mines specific websites and extracts information. As an example, we have evaluated it by searching for buildings in Germany which can be potentially used as emergency shelters.

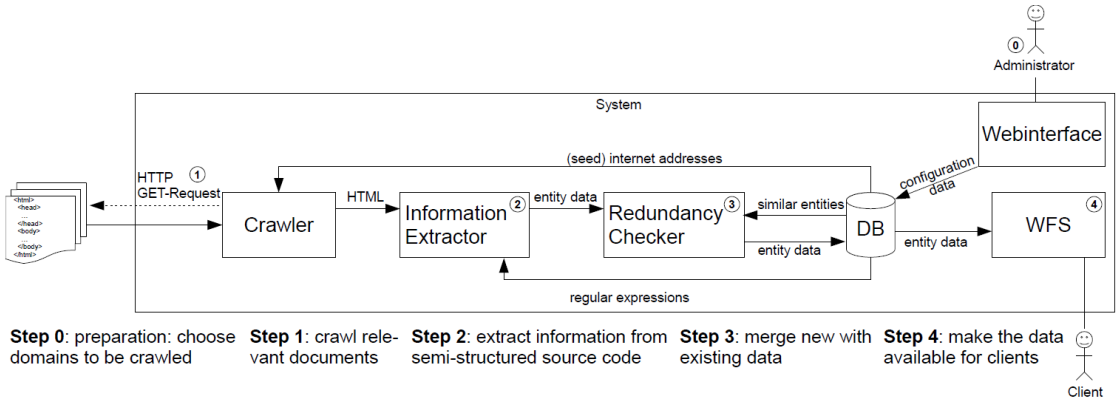

Figure 1. System overview.

2.1 Step 0: Preparation

In our approach we do not gather data from the whole web, but limit our search to a set of specific internet domains. The assumption is that such strict subsets of the data can provide almost all relevant information for the non-trivial category of interest with less interference. We are aware of the impact of this limitation, as a manual selection of data sources is needed, which carries the risk to leave out relevant information.

Since we direct our attention on the correctness of mined data, the precision of the data is more important than an outstanding recall. Regarding our prototype, we needed to decide on a set of websites that should be crawled. Emergency shelters are not clearly defined, so multiple concepts, each with its own sources, were combined.

2.2 Step 1: Crawling relevant web pages

Following (Najork and Wiener 2001), we use breadth-first search for crawling the sources. Multiple domains can be crawled in parallel to speed up the process.

2.3 Step 2: Extracting Information

We assume that within one heterogeneous part of a source, entities that contain the same type of information are stored in the same, semi-structured way. Provided that there are textual characteristics which identify the required information, we have chosen regular expressions (Friedl 2006) as our means for information extraction.

For each data attribute and each source, one regular expression is needed. We use additional services to complete the data, mainly if we can deduce new information from already found data. When no geographic coordinate is found, the Google Geocoding API is used to compute the missing information using the address. Missing phone numbers are added as well, using a German online telephone directory service (http://dasoertliche.de).

2.4 Step 3: Merging retrieved data

In our approach we extract information from different sources. Consequently, we have to ensure that no duplicates are inserted into our database, and that different incomplete information on the same entity are merged to a more complete one. Therefore, a strategy is needed to detect whether two entities are equal. If we find an object and identify it as equal to another object found before, we add the missing attributes and replace attribute values by those of the newer object.

Besides increasing the completeness of the data, we can also include more attributes per object by merging the information of different sources.

In our implementation, we developed heuristics to find equal objects/entries using certain key attributes and normalization algorithms.

2.5 Step 4: Publishing the data

The retrieved and revised data can be published in a standardized format. There are several possibilities to output the data, for example as RDF or with an OGC compliant web feature service. In our prototype, we implemented a basic web feature service (WFS), so the mined data can be directly included in GIS software like Udig or Gaia.

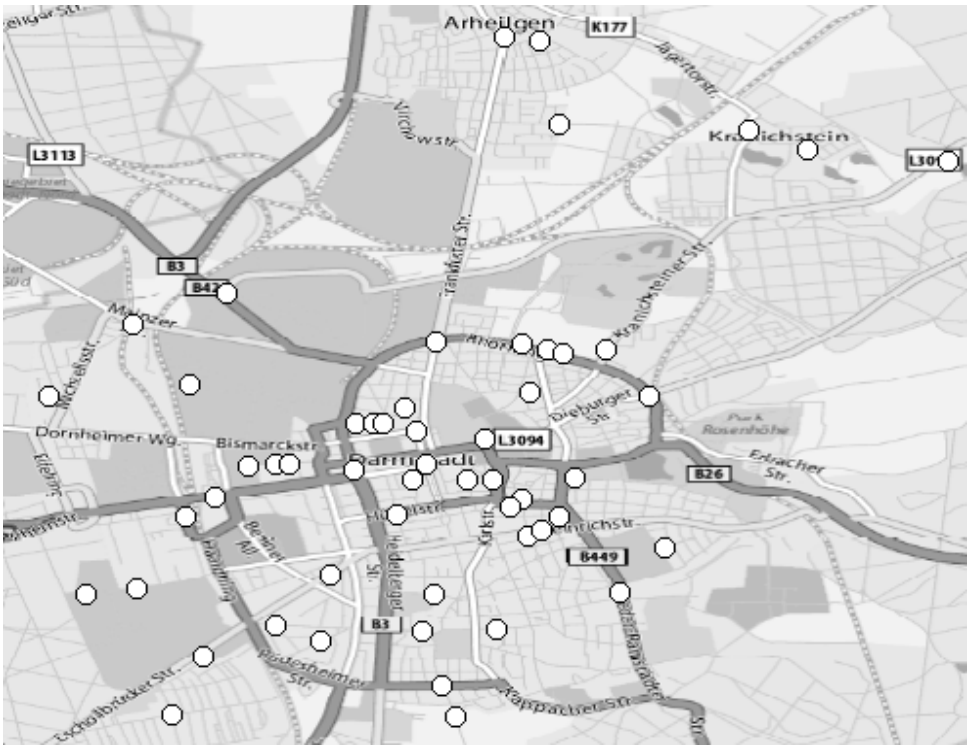

Figure 2. Map Section of Darmstadt with results from our prototype visualized by Gaia (white points are buildings that might serve as emergency shelters).

3. Evaluation

For our evaluation, we chose six websites which contain information about possible emergency shelters.

These are for example sites about schools, venues, hotels, etc. We will evaluate single sources first and then compare them to the state-of-the-art.

3.1 Evaluation of single sources

We wanted to verify the need for more than one resource and compared the results of a crawler run making use of all sources with other crawler runs limited to single sources. Therefore we created several independent databases for the crawler runs. We evaluated them based on a manually created gold standard for the counties “Darmstadt-Dieburg” and ”Darmstadt”.

The difficulty in gaining information about emergency shelters is the diversity of the eligible buildings. Since there is no source which contains all entities, we combine different sources to a new type, which can be understood as ad hoc category. As shown in Table 1, a satisfying recall is only possible by the aggregation of sources.

Table 1. Comparison of evacuation shelter sources (address), based on all evacuation shelters of the gold standard.

Source	Type	Recall	Precision
schulweb.de	schools	0.152	0.866
gezielt.net	venues	0.071	0.714
places.falk.de	venues, schools, hotels, museums	0.263	1.000
hotel.de	hotels	0.152	0.867
hotelguide.de	hotels	0.061	1.000
schulradar.de	schools	0.384	0.947
our prototype	all	0.687	0.955

3.2 Evaluation against state-of-the-art

Since, as discussed above, Google and OSM do not allow searching for emergency shelters without a disproportional effort, we chose the easier and clearly defined domain of hospitals. In our prototype we used four web pages as sources, which complement each other. We evaluated the results of our prototype and data taken from the Open Street Maps project and Google Maps by calculating its correctness and coverage with the help of our gold standard (Table 2).

Table 2. Comparison of different approaches.

Approach	Recall	Precision
LinkedGeoData - hospitals	0.714	1.000
Google Maps - hospitals	0.800	0.857
Our prototype - hospitals	0.857	1.000

4. Conclusion & Future Work

We presented a concept for fetching and aggregating specific spatial information from semi-organized sources like selected websites. Important parts of a system capable of executing this task are a web crawler, an information extraction module, a persistent store and an output subsystem for spatial exploration.

We have shown a sample implementation which is able to find hospitals and emergency shelters. We evaluated our prototype against state-of-the-art approaches and discovered that we outperform them concerning recall and precision by merging information of different sources.

Because of the encouraging results, further research on this topic is worthwhile. To improve data quality, a ranking algorithm could be implemented instead of majority voting to deal with the fact that websites have varying correctness and up-to-date-ness. Data retrieved from higher-rated sources would be preferred.

We have explained that in our system an expert is required to identify relevant websites and to describe the structure of these sites using regular expressions. We plan to try different advanced techniques, namely automatic source discovery and automated pattern learning to reduce the amount of needed user interaction and to make the setup of our service possible even for untrained users. As we use very simple regular expressions, a knowledge-lean approach could be used to learn these patterns (Muslea 1999).

References

Friedl J 2006, Mastering Regular Expressions. O'Reilly Media, Inc.

Muslea I 1999, Extraction patterns for information extraction tasks: A survey, In The AAAI-99 Workshop on Machine Learning for Information Extraction.

Najork M and Wiener JL 2001, Breadth-first search crawling yields high-quality pages.

Paulheim H, Doeweling S, Tso-Sutter K, Probst F and Ziegert T 2009, Improving Usability of Integrated Emergency Response Systems: The SoKNOS Approach, *Proceedings "39. Jahrestagung der Gesellschaft für Informatik e.V. (GI) - Informatik 2009"*, volume 154 of LNI, 1435–1449.