Henrich, Andreas; Lüdecke, Volker; Blank, Daniel

Approaches for determining the geographic footprint of arbitrary terms for retrieval and visualization

Date of secondary publication: 24.02.2025

Accepted Manuscript (Postprint), Conferenceobject

Persistent identifier: urn:nbn:de:bvb:473-irb-1066029

Primary publication

Henrich, Andreas; Lüdecke, Volker; Blank, Daniel (2008): Approaches for determining the geographic footprint of arbitrary terms for retrieval and visualization, in: Walid G. Aref, Mohamed F. Mokbel, und Markus Schneider (Ed.), GIS '08 : Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems, New York: ACM, doi: 10.1145/1463434.1463488.

Approaches for Determining the Geographic Footprint of Arbitrary Terms for Retrieval and Visualization

Andreas Henrich
University of Bamberg
Germany
andreas.henrich@uni-bamberg.de

Volker Lüdecke
University of Bamberg
Germany
volker.luedecke@uni-bamberg.de

Daniel Blank
University of Bamberg
Germany
daniel.blank@uni-bamberg.de

ABSTRACT

Determining the bounds of geographic regions is an important task for geographic search engines which use concept@location-type of queries. The location a user specifies is often not contained in the underlying gazetteer or geographic database, which might be due to vernacular descriptions of regions or because the location is not a geographic region in the narrow sense, which is the case in queries like *campground near theme park*. In the present paper we describe different ways for automatically determining a geographic footprint for those locations so that a geographic search engine is able to deal with all kinds of location-descriptions. The same approaches can be used to visualize the geographic correlation of arbitrary terms, like the visualization of the spread of certain colloquialisms.

The basic idea is to mine locations found in the top documents resulting from a query consisting of the terms the user has chosen to specify the location. We describe how this can be done using kernel density estimation, clustering and a combination thereof.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Indexing methods, Thesauruses; H.3.3 [**Information Search and Retrieval**]: Query formulation, Search process

Keywords

Vague geographic regions, Geographic search engines

1. INTRODUCTION

Textual user queries of a concept@location-type – like *jobs in southern Germany* – are quite common in Geographic Information Retrieval. The location-part of such a query usually refers to a certain place or a region and constitutes a geographic constraint. To take that into account, a geographic search engine processing a geographic query therefore has to be able to know the position or the boundary of the given location. That is usually done by looking it up in a gazetteer-like database [5], which usually contains information about cities, locations or regions with distinct boundaries, like administrative regions. Users might also specify locations that are not covered by the gazetteer, though, either because they use vernacular expressions or locations that are not geographic regions at all, but are perfectly in line with the concept@location-idea. This is the case whenever (non-geographic) terms (concepts) are used in a geographical sense and do not refer to the content of documents, like *campground near golf course*. The search engine should be able to deal with these queries and therefore has to find out appropriate corresponding geographic extensions of these concepts.

In this paper we present and compare approaches to automatically determine boundaries for geographic regions as well as a geographic footprint for arbitrary (non-geographic) concepts. These boundaries are then to be used to create a geographical ranking of documents. The approaches in this paper utilise knowledge from documents of the world wide web and use different ways to derive the resulting footprint, namely density estimation, clustering and a combination thereof. We evaluate the applicability of these methods and the quality of the resulting footprints.

The paper is structured as follows: Our approaches are presented in section 2, while section 4 deals with the automation aspects of the approaches. An evaluation of the quality of the results and the performance of the methods is done in section 3. Section 5 covers related work. Finally, section 6 concludes the paper and discusses future work.

2. DETERMINING GEOGRAPHIC FOOTPRINTS OF ARBITRARY CONCEPTS

2.1 Aims and Assumptions

In this section we present three different approaches for automatically computing a geographic footprint for geographic regions as well as for arbitrary concepts or terms. The idea behind all approaches is to use them in the context of a geographic search engine, which leads to several assumptions, aims and restrictions: (1) Since we want to determine the boundaries of the location at query time, the approaches have to work as fast as possible. (2) The approaches should work for geographic regions and arbitrary concepts alike. (3) Our prototype system is restricted to text in German language and to locations in Germany, but the methods are applicable to any language and location data. (4) The accuracy of the results has to be adequate. Since there is a lot of vagueness involved in both the approaches themselves and the IR system using their results, we are satisfied with results that are well applicable to ranking in the retrieval process. (5) The whole process must work completely automatically without user intervention.

Our approaches are designed to work with a geographic search engine that is also capable of answering standard textual queries. User queries comprise textual descriptions of the concept aiming at the content of documents and of a geographic constraint (a location). Whether this is done by two separate textfields or by a more sophisticated analysis of the queries does not matter for the purposes of this paper.

2.2 Basic Procedure

The situation we are concerned with can be described as follows: We are given a set of terms T_{loc} describing the location part of a concept@location-query[1]. We want to derive a geographical footprint of query region R_q (or more precisely $R_q(T_{loc})$) representing that location.

The basic idea is that the knowledge about the geographic representation R_q for T_{loc} can be found in the world wide web. If we have several documents dealing with T_{loc} and these documents contain location descriptors (toponyms), we can derive locations relevant to determine R_q.

For our approaches, this knowledge is represented by web pages $D_{retrieved}$ relevant to the location description T_{loc}. We used the Google API to get the first 500 results (html-pages only) for each considered location description T_{loc} and archived them locally. We used the first k of these documents as $D_{retrieved}$, while k is a design parameter, which was set to $k = 20$ for the experiments in this paper. That way we can better evaluate the total performance of our approaches, since a working geographic search engine could simply use its own (local) index and data.

The next task was to detect and disambiguate toponyms in $D_{retrieved}$. For the approaches in this paper we used the same mechanisms as described in [3], which comprise simple disambiguation steps. The locations of these toponyms are then looked up in a gazetteer. Figure 1 (left) shows an example of the locations found for the geographic region *Eifel* as well as a correct representation for that region, which was determined by human experts. The locations are in addition weighted by term and document frequencies.

In the following we describe how we use density estimation, clustering and a combination thereof to derive the geographic footprints R_q based on that location data.

2.3 Density Estimation

The first approach uses kernel density estimation, as described in [6] with the modifications presented by us in [3], which we will call *incremental density estimation* (IDE) in this paper. The whole target area —Germany in our case— is divided into tiles with a tile size of about one square kilometer, which is reasonable in the context of a Germany-wide search engine.

For each location (toponym) found in $D_{retrieved}$, we calculate the corresponding fraction of the total density function for only a number of tiles, as long as the contributing value is above a certain level and add it to the result matrix.

In order to get a satisfying result region R_q it is necessary

[1]With the query *campground near golf course* we get T_{loc} = *golf course* and with the query *hiking in Franconia* we get T_{loc} = *Franconia*.

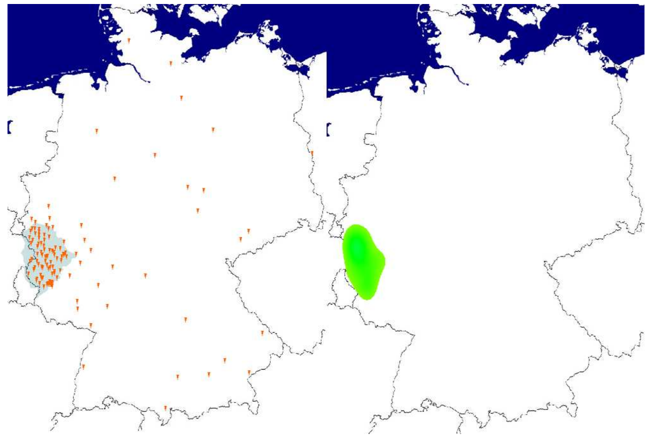
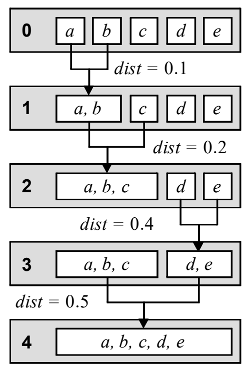


Figure 1: left: Locations found for *Eifel* and its manually determined polygonal representation; right: Density Estimation for *Eifel*



	a	b	c	d	e
a	0	0.1	0.3	0.5	0.65
b	-	0	0.2	0.5	0.5
c	-	-	0	0.55	0.5
d	-	-	-	0	0.4
e	-	-	-	-	0

Exemplary distances

Figure 2: Dendrogram from example data

to determine a threshold value T_{min} for the density surface, with all tiles with a smaller value not being part of R_q. The resulting 2D-area may be stored in a quadtree, a polygonal representation or any other representation the search engine uses for storing geographic footprints of documents or query footprints. Figure 1 (right) shows the region *Eifel* as automatically determined via incremental density estimation.

2.4 Clustering

Another approach is to use the location data as input for hierarchical clustering. That way, one cluster is supposed to contain the locations related to T_{loc}, while the other clusters may be considered noise.

We use single-link clustering for building a dendrogram of possible clusters, because its tendency to form one big cluster and many small ones is desirable in our application. That way we have a single parameter —namely, the maximum distance d_{max} for merging two clusters— for changing the size of the relevant cluster. Figure 2 sketches the dendrogram resulting from the exemplary data given there. Step by step the two clusters with the smallest distance to each other are merged. Because of the single-link clustering the distance between two clusters C_i and C_j is defined as $\min_{a \in C_i, b \in C_j} dist(a, b)$.

Each layer of the dendrogram marks a merging step. De-

approach	max	median	Ø	min
IDE	0.83	0.43	0.45	0.0
Clustering, IDE	0.78	0.39	0.40	0.0
convex hull	0.81	0.41	0.42	0.0
contour polygon	0.84	0.42	0.41	0.0

Table 1: Comparison of the result quality of all approaches by sim_{prec} for 75 test regions

pending on d_{max} this results in a different set of clusters. For $d_{max} = 0.3$ we get three clusters, for example (situation at merging step 2). In our experiments we suppose the largest cluster to contain the relevant locations (which would be $\{a, b, c\}$ in the example).

Since we are looking for representations of areas, we determine the convex hull and the contour polygon for the locations of the result cluster.

2.5 Combined Clustering and IDE

Another idea is to use clustering as a filter for relevant locations and then apply another method for determining a geographic representation. We use density estimation surfaces as alternative to a polygonal representation, the input being only those locations that belong to the main cluster. That way, we do not have to consider the threshold T_{min} to remove false positives, but instead use it to determine the spread of the approximated region R_q. Especially more complex shapes cannot be well represented by a convex hull or a contour polygon, which both simplify a U-shape to an O-like shape, for example. In such situations the combined approach has its advantages.

3. EVALUATION

In this section we evaluate the quality of the results of each approach presented above as well as their performance.

3.1 Result Quality by Approach

There are simple measures for measuring the similarity between two polygons. Overlap-percent of the areas or percent-inside versus percent-outside are typically used for that purpose. One example is the measure mentioned in [4]. $A(R)$ is the surface area of a region R, R_q is the approximation of the query region, and R_c is the correct region:

$$sim_{prec} = \frac{2 * A\left(R_q \cap R_c\right)}{A\left(R_q\right) + A\left(R_c\right)} \tag{1}$$

For an evaluation of the quality of the computed geographic footprints, we used 75 manually created "correct" footprints R_c. To this end, two persons diligently researched various sources of information to find out the commonly accepted borders for these regions of various sizes in Germany. Since the borders of most of the regions are vague, the resulting polygons must be seen as an approximation, but are well suited for our evaluation purposes, because all regions were created by using the same criteria and judgements.

Table 1 shows the quality of our approximations R_q in comparison to the manually determined correct geographic regions R_c, calculated by sim_{prec}.

Looking at all approaches, the approximations gained by incremental density estimation (IDE) have the highest quality, although the differences to the clustering results are small, overall. Although all approaches still have a lot of tuning potential, the density estimation is the most mature of all approaches, which might be one reason for that.

approach	max	0.90 quantile	median	Ø	min
IDE	6941	1915	653	963	99
Clustering, IDE	7847	1943	635	977	59
convex hull	893	51	5	26	1
contour polygon	892	45	6	27	1

Table 2: Comparison of the run-time performance of all approaches (values in ms) for 115 test regions

3.2 Performance Evaluation

If these approaches are to be used in a geographic search engine, in order to determine geographic footprints for those user-specified *locations* that are unknown to the gazetteer of the search engine, performance is a very important issue. The total time needed for determining the footprint is the sum of the efforts for the processing steps (1) retrieval of $D_{retrieved}$, (2) toponym resolution, and (3) the actual approaches described in this paper. (1) is a standard top-k-query for textual search engines, which can be done very fast with state-of-the-art search engine technology. For (2) we introduced a specialized toponym index in [3], which is built at the time of indexing, so that the cost for this is negligible. In addition, for a comparison of the approaches, the costs for (1) and (2) are the same, so we will evaluate the runtime performance of the approaches for determining the geographic footprints – i.e. for step (3).

On the other hand, if the aim is to use these approaches as a standalone application to visualize arbitrary concepts, performance is less important. This would allow for a more sophisticated tuning of the approaches, which is not the focus of this paper, though.

For an evaluation of the performance of each approach, we computed 115 region representations and measured the time effort for each region and each approach, using k=20 web documents $D_{retrieved}$ per region, respectively. For the incremental density estimation we used a tile resolution of about one square kilometer per tile, which we found to be sufficient for the context of not too small regions for a Germany-wide search engine. We used a desktop PC (Pentium4, 3GHz, 2GB) for all measurements, so the actual values are less important than the relative differences between the approaches. For the density estimation we used an implementation of our own and for single-link clustering, i.e. building the dendrogram, we used an off-the-shelf implementation (Alias-i LingPipe[2]).

Table 2 gives an overview of our performance evaluation. While computation time of the density estimations (*standalone* and *after a clustering step*) mostly seems to be fast enough, the clustering approaches are significantly faster.

Overall, the performance of our system seems to make it possible to use our approach at query time. Clustering increases the performance by a factor of roughly 40 versus density estimations while maintaining a similar result quality.

[2] http://alias-i.com/lingpipe/

4. AUTOMATION ASPECTS

The presented approaches can work fully automatically without user interaction. The critical aspect for the quality of the automatic process is the determination of threshold values, which is more or less the decision what belongs to a certain region and what does not.

In case of the density estimation representation, we have shown in [3] that the threshold value correlates quite well with the maximum value V_{max} found in the data tiles. That way, the individual threshold value $T_{predicted}$ for a given query region can easily be determined by the formula: $T_{predicted} = max(0.2, 0.91 * V_{max} - 0.79)$

While the clustering itself (building the dendrogram) already is a fully automatic process, the relevant decision is where to set the maximum distance d_{max}, which determines the number of clusters and therefore also the elements in the main cluster.

We took a training set of 40 user regions (disjoint to the 75 user regions R_c used for the evaluation of the result quality) and computed the average sim_{prec} for each minimum distance between 0 and 1, step size 0.01. Based on this data, we chose a distance d_{max} of 0.2 for the experiments in this paper, which leads to robust and good average results of the training set.

The combined approach described in section 2.5 does a clustering first, for which we could use the same distance parameter as we did for the standalone clustering, of course. For the subsequent density estimation we also used the same formula as for the standalone density estimation approach at first. But since we already use clustering as a filter for eliminating noise, the threshold value has a different role here (determining the spread of the region). We therefore calculated a fixed threshold value that was on average optimal for the 40 regions of the training set. Using this value lead to a slightly better result quality (about 10% for the average quality).

5. RELATED WORK

In [1] regions are approximated based on Voronoi diagrams which are constructed by a set of points known to be inside the target region and another one with points outside the region, but it does not deal with the source of the information about the sets. A similar approach is found in [9], where regions are represented by indeterminate boundaries with an upper and a lower approximation, which is also based on given sets with points and regions inside/outside the region.

The following methods for semi-automatically creating representations for vague regions mine the knowledge contained in the www: The authors of [6] use kernel density surfaces for the representation of the imprecise regions. [7] presents two approaches to compute representations of regions, based on evidence of points that are likely to lie inside or outside this region. [2] modifies the algorithm from [1]. The authors describe methods to obtain locations inside the target region and locations outside and use trigger phrases and patterns for the document retrieval to improve precision. [8] also makes use of regular expressions to acquire place names by a web search. It presents a method for automatically determining vague footprints, represented as fuzzy sets.

In comparison to the above mentioned approaches this paper has two main contributions: It proposes novel approaches to derive the region representing the location of a concept@location-query where the location is unknown to the used gazetteer and it presents experimental results showing the result quality and the performance.

In [3] we have presented preliminary work. We showed in this paper that the newly introduced approach based on clustering is advantageous with respect to the processing time.

6. CONCLUSION AND FUTURE WORK

In this paper we introduced an approach using clustering as a tool to determine the geographic footprints for geographic regions and arbitrary concepts. We looked at the results using it as a standalone-tool as well as combining clustering with density estimation.

We compared the effects of each measure as well as the performance of the approaches regarding result quality as well as run-time performance. While the density estimation approach lead to slightly better approximations of regions, the clustering approaches proved to be significantly faster (roughly factor 40).

Future work will be to improve the quality of the footprints created by clustering approaches. The first step will be to use a variable parameter d_{max}, depending on the properties of the locations and the resulting dendrogram.

7. REFERENCES

[1] H. Alani, C. B. Jones, and D. Tudhope. Voronoi-based region approximation for geographical information retrieval with gazetteers. *Intl. Journal of Geographical Information Science*, 15(4):287–306, 2001.

[2] A. Arampatzis, M. J. van Kreveld, I. Reinbacher, C. B. Jones, S. Vaid, P. Clough, H. Joho, and M. Sanderson. Web-based delineation of imprecise regions. *Computers, Environment and Urban Systems*, 30(4):436–459, 2006.

[3] A. Henrich and V. Lüdecke. Determining geographic representations for arbitrary concepts at query time. In *LOCWEB '08: Proc. of the first intl. workshop on Location and the web*, pages 17–24, New York, NY, USA, 2008. ACM.

[4] L. L. Hill. *Access to Geographic Concepts in Online Bibliographic Files: Effectiveness of Current Practices and the Potential of a Graphic Interface*. PhD thesis, University of Pittsburgh, 1990.

[5] L. L. Hill. Core elements of digital gazetteers: Placenames, categories, and footprints. In *ECDL '00*, pages 280–290, London, UK, 2000. Springer-Verlag.

[6] R. Purves, P. Clough, and H. Joho. Identifying imprecise regions for geographic information retrieval using the web. In *Proc. of GIS RESEARCH UK 13th Annual Conf.*, pages 313–318, Glasgow, UK, 2005.

[7] I. Reinbacher, M. Benkert, M. J. van Kreveld, J. S. B. Mitchell, and A. Wolff. Delineating boundaries for imprecise regions. In G. S. Brodal and S. Leonardi, editors, *ESA*, volume 3669 of *LNCS*, pages 143–154. Springer, 2005.

[8] S. Schockaert, M. D. Cock, and E. E. Kerre. Automatic acquisition of fuzzy footprints. In R. M. et al, editor, *OTM Workshops*, volume 3762 of *LNCS*, pages 1077–1086. Springer, 2005.

[9] T. J. Vögele, C. Schlieder, and U. Visser. Intuitive modelling of place name regions for spatial information retrieval. In *COSIT*, pages 239–252, 2003.