

Geographic Focus Detection using Multiple Location Taggers

Philipp Berger*, Patrick Hennig*, Dustin Glaeser†, Hauke Klement† and Christoph Meinel‡
Hasso-Plattner-Institute
University of Potsdam, Germany
*{philipp.berger, patrick.hennig}@hpi.uni-potsdam.de
†{dustin.glaeser, hauke.klement}@student.hpi.uni-potsdam.de
‡office-meinel@hpi.uni-potsdam.de

Abstract—Being able to identify locations associated to a Web resource is essential for providing location-based Web applications. However, geographical information in Web documents is rarely supplied in a machine-readable way and therefore not easily discoverable. As a consequence, it is necessary to extract geographical keywords from Web documents and to associate locations with them. This method is called *location tagging*.

In this paper we present a location tagging approach for unstructured documents which utilizes multiple external location providers. Detected locations are ranked according to their relevance for the document, in order to identify a document’s geographical *focus*, which is its most representative location. We present an exemplary implementation of our proposed approach using two location providers and evaluate our method’s applicability.

I. INTRODUCTION

Extracting geographic and location-related information from online documents is one of the many aspects of the ongoing attempt to make the World Wide Web analyzable and understandable to machines - mainly in order to ease the processing of huge amounts of unstructured documents for human users.

Online documents contain a huge amount of geographic information. For some content on the Web relevant locations are easily available, e.g. for photos with embedded GPS coordinates or messages in social networks that automatically get the author’s location assigned. For the majority of unstructured texts, however, the author’s location is neither the only relevant place, nor is it delivered in a formalized, structured way. Often geographically relevant information is not easily discoverable. Therefore, it is necessary to analyze each text and search for hints of relevant locations, the most obvious being place names. As an example, a news article that mentions the cities Potsdam, Cologne, Munich, and Helsinki can be seen as related to these places, as well as to Germany, since multiple mentioned locations belong to this country. In this paper we use the term *location tagging* to describe the process of extracting relevant locations and assigning them to a document (see Figure 1).

An important tool to identify place names in texts is a *gazetteer* which holds a comprehensive list of place names. Nevertheless, assigning the correct location to a term is not as simple as a look-up in the gazetteer, since ambiguity

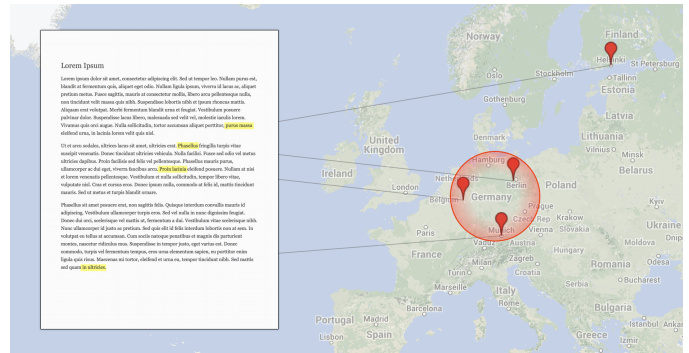


Figure 1: Location Tagging and Focus Detection

of potential place names is a common problem¹. Regarding place names there are two general categories of ambiguity: *geo/geo* and *geo/non-geo*. *Geo/geo* ambiguity means that two distinct locations have the same name, e.g. Berlin (Germany) and Berlin (Wisconsin, USA), while *geo/non-geo* ambiguity refers to the problem of place names that are also commonly used words. Examples for *geo/non-geo* ambiguity are the place names Mobile (Alabama, USA), Reading (England), and To (Myanmar) [2]. To disambiguate place names like these correctly, a gazetteer look-up can be combined with suitable heuristics or techniques like named-entity recognition or machine learning.

Eventually, the odds are that many different locations get assigned to a single document, differing in frequency and relevance. In order to maintain an overview and detect the most significant locations easily, it is desirable to have information regarding all mentioned locations’ relevance and relations at hand.

In this paper we present a location tagging solution for unstructured documents which aggregates locations retrieved from multiple external services. By combining multiple heterogeneous location providers we can exploit the strengths of single providers, such as superior coverage for a specific area. Aside from that, locations detected by multiple providers are more likely to be correct. In order to provide a synoptic view, we rank all identified locations according to their relevance

¹Smith et al. [1] report that 92% of the names in their corpus of historical texts are ambiguous.

for the document. Using the ranked locations we determine one or multiple geographic foci of the document, in this way underlining the most representative locations. This work also presents an implementation of our approach, which identifies locations mentioned in blog posts using two location providers: *SAP HANA*'s built-in text analysis engine, as well as Yahoo's *PlaceSpotter* Web service. Based on this implementation, we investigate the two location providers' qualification for our use case, as well as our method's applicability.

II. RELATED WORK

Wang et al. [3] argue that the key challenge for location-aware Web applications is correct and efficient location detection. They state that there are at least three different types of locations which may co-exist in a single document. Various hints and sources are available to derive these locations. However, the selection of an appropriate method depends on the concrete location type. The distinct location types are the following:

- 1) A Web resource's *provider location* refers to the physical location of its owner. It can be determined by extracting embedded addresses and estimating whether they refer to the owner.
- 2) The *servicing location* of a Web page refers to the geographical scope it reaches. The servicing location can be investigated by analyzing geographic properties of outgoing hyperlinks or access logs, if available.
- 3) The term *content location* refers to the location a Web resource's content is about. This type of location can be spotted by identifying geographic entities in the text, collecting embedded meta data, and estimating the dominant location.

We implicitly make use of the proposed geographic clues for a document's content location by aggregating the results of different location taggers.

Amitay et al. [2] also describe a system that associates Web pages with geographic entities, which they call *Web-a-Where*. In addition to finding place names and their associated locations in a text, the system assigns a geographical *focus* to the examined document. This focus is meant to be a geographic entity that is discussed in major parts of the text. It can either be a place that is directly mentioned, such as Berlin, or a region that can be inferred from the mentions of several places, like Germany can be derived from the occurrences of Berlin, Frankfurt, and Munich. The focus algorithm we use is based on this approach, as described in Section IV-B. Amitay et al. report a precision of 80% for the recognition of individual place name occurrences and an accuracy of 91% for focus determination.

In general, there are several different approaches to tackle the problem of place name disambiguation and location tagging. Some are mainly based on gazetteer look-up [4], [5], others make use of named-entity recognition [6] or machine learning techniques. With the help of the latter, some researchers even try to find relevant locations for texts which do not contain any obvious place names at all [7]. However, our proposed solution is unique in terms of exploiting the combination of multiple location providers.

III. PROJECT SCOPE

Blog Intelligence is a web mining application tailor-made for blog mining with the objective to map, and ultimately reveal, content-oriented network-related structures of the *blogosphere* by employing an intelligent *blog crawler*. As described in [8], BlogIntelligence is able to harvest the pool of millions of interconnected blogs, called *blogosphere*.

To create relevant results out of this huge amount of data it is necessary to analyze it based on different techniques and algorithms. For example the knowledge about atmospheric pictures or trends for a specific topic can be very important for a lot of organizations or a special group of people like politicians. There are three important parts, the crawler, an analysis framework and finally a visualization tool to make the results comfortable to read for humans. To complete this overall approach we started a project called *Blog-Intelligence*. The current implementation of the project is already working as a prototype for the German *blogosphere*².

IV. GEOGRAPHIC FOCUS DETECTION

This Section presents our contribution, which is a method for detecting and ranking locations in unstructured documents, such as Web pages, based on multiple location providers. It also presents an exemplary implementation.

A. Location Detection

1) *Workflow*: The general location detection workflow comprises the following steps:

- 1) **Input Selection**: Initially, we select qualified documents from the input data. Qualified documents have to contain textual content from which geographical information can be extracted.
- 2) **Location Tagging**: We query different location providers for locations mentioned in the text. Every provider may use a proprietary format for the presentation of its response.
- 3) **Location Unification**: To be able to compare and relate all locations, we unify them. In this sense, we map the locations retrieved from different services to unique entities in a common format and in the same language.
- 4) **Location Enrichment**: We fetch detailed information for every location, including latitude, longitude, and type. We also obtain every location's superordinate regions.
- 5) **Result Output**: Finally, the identified locations are returned.

An overview of our workflow, based on the two location providers utilized in our example implementation (see Section IV-C), is shown in Figure 2.

2) *Location Tagging*: To identify locations mentioned in the input document, we use stand-alone services. We query a number of location providers to exploit the potential of multiple location sources.

²<http://www.blog-intelligence.com>

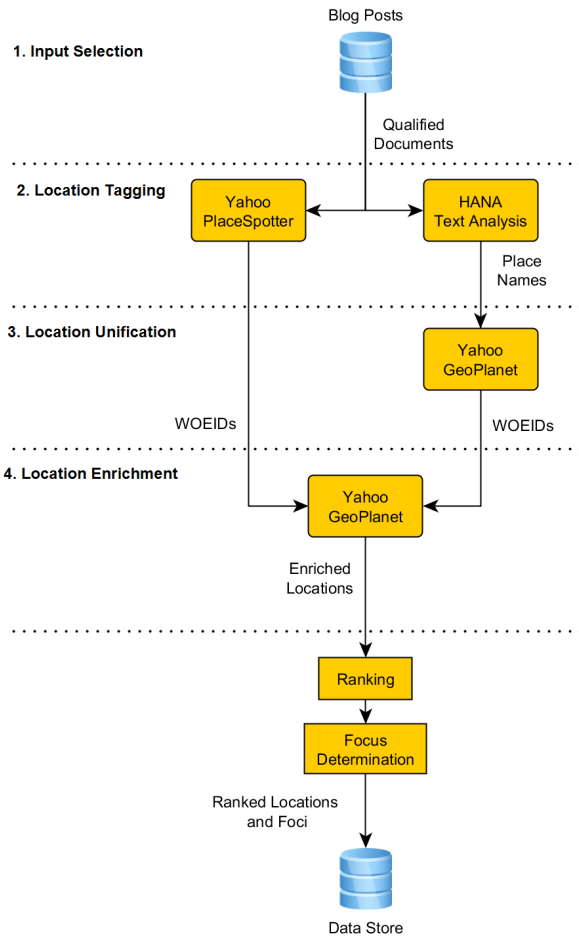


Figure 2: Location Tagging Workflow, Based on PlaceSpotter and HANA Text Analysis

Adding a new location provider requires a provider-specific adapter component, which accepts a full-text document and returns the following information for every detected location:

- A location descriptor
- The identified text token
- The number of occurrences

3) *Location Unification*: For the unification of the places identified by different location taggers we use *Yahoo GeoPlanet*³, a service providing a large subset of all permanently named places on earth. The service covers about six million places, ranging from continents and countries to postal codes and points of interest. GeoPlanet and other Yahoo services use so-called *Where On Earth Identifiers*⁴ (WOEIDs) to reference entities in a unique fashion. WOEIDs support a hierarchy model: Every place may belong to a number of superordinate geographic entities and may contain a number of subordinate geographic entities.

4) *Location Enrichment*: To simplify data analysis and visualization we obtain detailed information for all places, including geographic coordinates.

³<http://developer.yahoo.com/geo/geoplanet/>

⁴<http://developer.yahoo.com/geo/geoplanet/guide/concepts.html#woeids>

The detailed information we retrieve contains the following attributes⁵: name, type, WOEID, latitude, longitude and subordinate regions.

B. Location Ranking

To enable location-based search and analytics we want to determine the significance of all mentioned locations for a document. The most representative location or locations of a document present its geographic focus or foci. Having focus information at hand allows us, for example, to find the most important locations for a blog post or the most significant blog posts regarding a certain place.

Our ranking algorithm is based on the proposal made by Amitay et al. [2]. Every identified location is assigned a score, multiplied by the number of its occurrences. In addition to it, a corresponding score is assigned to every superordinate region. The more hierarchy levels are traversed, the lower is the score. In this way, a smaller and therefore more accurate location achieves a higher score than its superordinate regions. However, a common superordinate region of many mentioned places obtains a higher total score than its individual children and can become a focus, even if not explicitly mentioned. The location with the highest score embodies our page focus.

In contrast to Amitay et al., in our use case the algorithm has to work with unified aggregated data. To fit our needs we had to make the following adaptations:

- 1) We omit a confidence score for detected locations, as location disambiguation is not in our project's scope and because we cannot presume that every utilized location provider supplies this piece of information. However, as we combine multiple location sources and count every detected location individually, locations found by multiple providers achieve a higher ranking. This effect partially compensates for the lack of a confidence score.
- 2) The algorithm uses a decay factor to assign decreasing scores to all superordinate regions of a directly mentioned place. We iteratively adjusted this factor to fit the usual hierarchy depth in our location data. We achieved satisfying results with a decay factor of 0.8. However, since the number of hierarchy levels differs from region to region, our results also differ depending on the local administrative hierarchies.

1) *Example*: Let us take a look at an example to illustrate our location ranking approach. Assume we analyzed a blog post on the subject of the United States' relations to Middle Eastern countries and the following locations have been identified by the location providers:

- Israel, 10 occurrences
- Palestine, 8 occurrences
- Gaza, 3 occurrences
- Doha, 1 occurrence
- United States, 1 occurrence
- Washington, 1 occurrence

⁵For the sake of clarity we omit the superordinate regions' attributes.

These six locations are directly mentioned in the text. Therefore, their scores are raised by 1.0 for each occurrence, which leads to ten points for Israel, six for Palestine, and so on. Since we want to be able to determine an overall focus for the document, the superordinate regions of each location are also taken into account. However, those indirectly mentioned locations do not get the same score assigned for each occurrence. Instead, the particular score is multiplied with a factor of 0.8 for each traversed hierarchy level. Therefore, we use a decay factor of 0.8^n for the n -th superordinate region of a location.

The enclosing regions of Gaza are Palestine and Asia. Palestine receives a score of 0.8 points and Asia a score of 0.8^2 for every occurrence of Gaza in the text. For six occurrences this sums up to 2.4 points for Palestine and about 1.5 points for Asia. The resulting scores for all locations and their superordinate regions are shown in Table 1⁶.

In the next step, we aggregate the points for each location and sort them by their final score. We also normalize the scores which makes it possible to compare the relative importance of locations between documents. The final ranking for our example is shown in Table 2.

To determine the foci of the document we step through the ranked locations from top to bottom. We consider every location a focus that is not already covered by or covering an existing focus. Therefore, the highest ranked location is always considered the main focus of a document, in this case Asia. Please note that Asia is determined as the main focus of the document, although it is not mentioned directly. Israel, Palestine, and Gaza are all covered by Asia, which prevents them from becoming foci themselves. The next location that is not covered by Asia is the United States, which becomes the second focus in our example. North America covers the United States, but ranks lower. The remaining locations are covered either by Asia or the United States.

C. Example Implementation

We implemented our proposed solution for the *BlogIntelligence*⁷ project which collects blog posts from the Blogosphere and enables real-time analysis on the harvested data. We built a Java-based component to be integrated into the project's *MapReduce*⁸ content retrieval workflow [8].

Our objective was to implement a location tagging solution which is able to handle a large bulk of data and which supports multiple languages, with emphasis on English and German, as these are the most prominent languages in the corresponding data set.

1) *SAP HANA Text Analysis*: Since the *BlogIntelligence* project uses SAP's in-memory database HANA [9] as database and application platform we decided to use HANA's built-in text analysis⁹ feature as our first source for location informa-

⁶Washington's administrative hierarchy includes both a county and a district named "District of Columbia". These are listed as "D.C.". Furthermore, North America is listed as "N.A."

⁷<http://blog-intelligence.com/>

⁸<http://research.google.com/archive/mapreduce.html>

⁹<http://www.saphana.com/community/about-hana/advanced-analytics/text-analysis>

tion and, by doing so, investigating its qualification for location tagging.

2) *Yahoo PlaceSpotter*: Yahoo PlaceSpotter¹⁰ is the second location provider we have chosen for our implementation. This service enables location-aware applications by providing geo-enrichment for unstructured content. PlaceSpotter identifies mentioned locations by evaluating place names, meta-data, and microformats. It disambiguates detected places automatically and returns WOEIDs for the identified locations. Thus, PlaceSpotter's result format is compatible with other services in Yahoo's ecosystem, especially GeoPlanet, the service we use for unification (see Section IV-A).

D. Location Explorer

To provide a tool to explore location-tagged blog data, we created the *Location Explorer*, a prototypical Web application. The tool's purpose is to visualize locations and foci detected in blog posts.

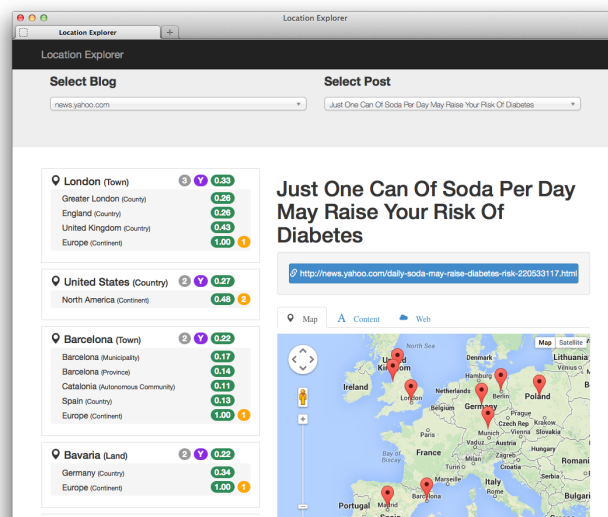


Figure 3: Location Explorer

Figure 3 shows the application while being used to explore the locations mentioned in a news article. The select widgets at the top of the user interface are used to select a blog from the database, as well as a post from this blog. The list on the left contains all places which are mentioned in the blog post. The same places are also visualized on a map on the right. Using the tab navigation, one can also examine the Web page from which the places have been extracted or the full text (highlighting identified tokens) only. This view allows to comprehend the mapping between analyzed text and detected locations.

The locations list also includes the more in-depth information like superordinate regions per mentioned place and amount of each place's occurrences in the text (gray numbers). It also contains the location providers which identified a specific place. In the example depicted in Figure 3, all visible

¹⁰<http://developer.yahoo.com/boss/geo/>

Table 1: Individual Location Scores

| Count | Scores | | | | |
|-------|----------------|-----------------|--------------|--------------|--------------|
| 10 | Israel (10) | Asia (8) | | | |
| 6 | Palestine (6) | Asia (4.8) | | | |
| 3 | Gaza (3) | Palestine (2.4) | Asia (1.536) | | |
| 1 | Doha (1) | Qatar (0.8) | | | |
| 1 | US (1) | N.A. (0.8) | | | |
| 1 | Washington (1) | D.C. (0.8) | D.C. (0.512) | U.S. (0.410) | N.A. (0.328) |

Table 2: Aggregated Location Scores, Sorted Descending

| Score | Normalized Score | Location |
|-------|------------------|---------------------------------|
| 14.85 | 1.00 | Asia |
| 10.00 | 0.67 | Israel |
| 8.40 | 0.57 | Palestine |
| 3.00 | 0.20 | Gaza |
| 1.41 | 0.09 | United States |
| 1.21 | 0.08 | North America |
| 1.00 | 0.07 | Doha |
| 1.00 | 0.07 | Washington |
| 0.80 | 0.05 | Qatar |
| 0.80 | 0.05 | District of Columbia (County) |
| 0.51 | 0.03 | District of Columbia (District) |

places have been provided by Yahoo PlaceSpotter and are therefore tagged with the letter Y. Further it shows each place’s normalized score (green numbers) according to the ranking algorithm (see Section IV-B) and the ranks of all detected geographic foci (yellow numbers). In the depicted example, these are Europe (1) and North America (2).

V. EVALUATION

This Section focuses on the evaluation of our work. Section V-2 evaluates the precision of our location tagging method, whereas Section V-3 evaluates the focus determination approach.

1) *Setting*: We performed our evaluation for a subset of Web pages from the BlogIntelligence database that our algorithms have fully processed. In order to acquire a representative sample of the database, the Web pages have been partitioned by their capture time. We used the two location providers discussed in Section IV-C.

We applied a crowd-based evaluation method. For that purpose, we provided two simple Web applications including a participants leaderboard as evaluation platform. Following this approach, we reached about 50 participants and collected circa 24,000 data points.

2) Location Detection:

a) *Precision*: To evaluate the precision of our utilized location providers, we built a Web application presenting a location, as well as an excerpt of the document’s text where the location has been detected. Participants had to decide whether the detected location is correct or not. Alternatively, participants could skip a location.

Tables 3 and 4 present the results of this evaluation. The overall precision of our location tagging approach is 70.0% if we count skipped locations or 82.5% otherwise. In either instance, the results show that HANA’s text analysis engine performs more precise location detection than Yahoo’s PlaceSpotter.

As expected, the number of skipped locations does not differ much between the two location providers. Our participants

Table 3: Location Detection Correctness (Including Skipped Locations)

| Source | Correct | Incorrect | Skipped |
|--------------|---------|-----------|---------|
| HANA | 84.2% | 4.7% | 11.1% |
| PlaceSpotter | 67.4% | 16.7% | 15.9% |
| Overall | 70.0% | 14.8% | 15.2% |

Table 4: Location Detection Correctness (Excluding Skipped Locations)

| Source | Correct | Incorrect |
|--------------|---------|-----------|
| HANA | 94.7% | 5.5% |
| PlaceSpotter | 80.1% | 19.9% |
| Overall | 82.5% | 17.5% |

often skipped locations due to insufficient textual context or missing command of the document’s language. These issues might also give location providers a hard time.

The false positives our participants have uncovered during evaluation show where our utilized location providers have weak spots. Often, false positives result from the wrong detection of ambiguous nouns and surnames. Also, homonymous verbs or units of measurement account for falsely identified places. Even virtually unmistakable terms and word groups, such as “I believe” and “I like”, are among the identified locations, since same-titled points of interest are listed by Yahoo PlaceSpotter.

b) *Recall*: We did not employ our crowd-based evaluation method to detect false negatives. Even so, we want to discuss some numbers.

Among the analyzed 149,000 Web pages there are about 91,000 Web pages without any identified location, whereas about 58,000 Web pages contain geographic information. All in all, 229,000 locations have been identified in our regarded data set which implies that there are almost four places mentioned in an average located Web page.

The majority of identified locations (197,000) has been found by Yahoo’s PlaceSpotter, whereas only 32,000 locations have been identified by HANA’s text analysis engine. However, we do not know for sure whether this gap is due to a wide difference in both location providers’ recall or if HANA simply had not processed the entire data set under consideration. Indeed, if we only take those Web pages into account that have certainly been processed by HANA’s text analysis engine, only 41,000 locations identified by PlaceSpotter remain. Moreover, another issue that shifts the detection ratio in favor of Yahoo’s PlaceSpotter is that locations detected by HANA may get lost during the unification step. However, this is not the case for PlaceSpotter as its location identifiers already match the ones used for unification.

As HANA’s text analysis engine delivers fewer but more precise results we assume that the employed algorithm places emphasis on precision rather than recall.

Table 5: Location Ranking Correctness (Including Skipped Web Pages)

| Correct | Incorrect | Skipped | Corrupt Locations |
|---------|-----------|---------|-------------------|
| 68.1% | 4.7% | 18.1% | 9.1% |

3) *Location Ranking*: We evaluated our location ranking algorithm by providing an adapted, slightly simplified version of the Location Explorer application (see Section IV-D). Participants had to decide whether the geographical foci we determined are representative or not. Again, participants could skip a document in case they were unsure. As we wanted to evaluate our location ranking approach independently of the quality of its input, participants should reject corrupted ranking results based on falsely detected locations.

It turned out that the second evaluation task was less suitable for our crowd-based evaluation method than the first one. Not only did we have fewer participants but also less informative results. This might be due to the fact that the second evaluation task was less precise and straightforward than the first one. Also, the concept of multiple foci has not been communicated well enough.

Table 5 shows the evaluation results including skipped documents. In 68.1% of cases our participants rated our determined geographical foci as accurate. It should be noted that a relatively high proportion of Web pages has been skipped. This was often the case for documents with widely scattered locations, since no clear focus was visible on the map. 93.6% are correct and 6.4% are incorrect, if we ignore skipped Web pages and assume a consistent distribution of correct and incorrect foci.

Incorrect foci which have not been caused by corrupted input data were often not precise enough, owed to a non-optimal location hierarchy. Repeatedly, continents were considered to be too vague to present a suitable focus. For instance, a human observer might consider the Middle East the geographic focus of some Asian countries mentioned in a text. Based on the data we can obtain from GeoPlanet, this region cannot be assigned, since it is a *supername* and not part of any location’s enclosing hierarchy. The same is true for other geographic umbrella terms, such as Benelux countries, Silicon Valley, Scandinavia, or Northern Germany.

Apart from that issue, individual foci were sometimes considered incorrect due to an arbitrary ordering of foci with the same score.

VI. FUTURE WORK

When inspecting our tagged blog posts we came across several cases where the result of our focus determination algorithm was not adequate enough due to non-optimal place hierarchies. As reviewed in Section V-3, too coarse or inconsistent location hierarchies can cause the effect that out-of-the-ordinary regions, such as the Middle East or Northern Germany, cannot embody a geographic focus. To solve this problem we would have to refine the set of places covered by Yahoo’s WOEIDs.

Moreover, a too fine-grained place hierarchy can equally produce suboptimal results. Locations with very deep hierarchies, including multiple minor administrative regions, are

hardly of any use and provoke an excessive decay of their superordinate regions’ scores, preventing them from becoming foci. To address this problem we might test taking fixed hierarchy levels only (town, state, country, continent) into consideration during place ranking.

VII. CONCLUSION

In this paper we proposed a location tagging approach for unstructured documents which aggregates locations obtained from multiple, heterogeneous location taggers. We also illustrated a focus determination approach which allows to identify the most essential locations associated to a document. Afterwards, we described an exemplary implementation of our approach, utilizing HANA’s built-in text analysis engine and Yahoo’s PlaceSpotter as location providers.

Using our implementation we evaluated our method’s applicability. We think that the results are promising, even for polyglot Web content. PlaceSpotter and HANA are suitable for processing larger sets of documents, such as our blog posts collection. SAP HANA’s general-purpose text analysis feature seems qualified for location detection. It yields fewer but more precise results than PlaceSpotter. Our focus determination algorithm achieved very good results, especially when provided with clean input data.

REFERENCES

- [1] D. A. Smith and G. Crane, “Disambiguating geographic names in a historical digital library,” in *Research and Advanced Technology for Digital Libraries*. Springer, 2001, pp. 127–136.
- [2] E. Amitay, N. Har’El, R. Sivan, and A. Soffer, “Web-a-where: geotagging web content,” in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2004, pp. 273–280.
- [3] C. Wang, X. Xie, L. Wang, Y. Lu, and W.-Y. Ma, “Web resource geographic location classification and detection,” in *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 2005, pp. 1138–1139.
- [4] R. Tobin, C. Grover, K. Byrne, J. Reid, and J. Walsh, “Evaluation of georeferencing,” in *Proceedings of the 6th Workshop on Geographic Information Retrieval*. ACM, 2010, p. 7.
- [5] M. J. Silva, B. Martins, M. Chaves, A. P. Afonso, and N. Cardoso, “Adding geographic scopes to web resources,” *Computers, Environment and Urban Systems*, vol. 30, no. 4, pp. 378–399, 2006.
- [6] J. Ding, L. Gravano, and N. Shivakumar, “Computing geographical scopes of web resources,” 2000.
- [7] B. Adams and K. Janowicz, “On the geo-indicateness of non-georeferenced text,” in *ICWSM*, 2012, pp. 375–378.
- [8] P. Berger, P. Hennig, J. Bross, and C. Meinel, “Mapping the blogosphere—towards a universal and scalable blog-crawler,” in *Privacy, security, risk and trust (passat), 2011 IEEE third international conference on and 2011 IEEE third international conference on social computing (socialcom)*. IEEE, 2011, pp. 672–677.
- [9] F. Färber, S. K. Cha, J. Primsch, C. Bornhövd, S. Sigg, and W. Lehner, “Sap hana database: data management for modern business applications,” *ACM Sigmod Record*, vol. 40, no. 4, pp. 45–51, 2012.