Geocoding with OpenStreetMap Data

Konstantin Clemens

Service-centric Networking
Telekom Innovation Laboratories
Technische Universität Berlin, Germany
Email: kclemens@tu-berlin.de

***Abstract*—*OpenStreetMap* (OSM) is a platform where users contribute geographic data. To serve multiple use cases, these data are held in a very generic format. This makes processing and indexing OSM data a challenge. *Nominatim* is an open source search and geocoding engine that consumes OSM data. While Nominatim does process OSM data well, it does not use *term frequency – inversed document frequency* (TF/IDF) based ranking of search results. *Lucene* is a framework offering TF/IDF for ranking of indexed documents. In this paper Nominatim's processing of OSM data is utilized to assemble full addresses with their geocoordinates. These addresses are then indexed in *Elasticsearch*, a web service on top of Lucene. The resulting TF/IDF based geocoding system is benchmarked in comparison with plain Nominatim. The analysis shows: TF/IDF based ranking yields more accurate results, especially for queries with unordered address elements or only partially specified addresses.**

***Keywords*–Geocoding, Address Search, OSM, Nominatim, Elasticsearch**

I. INTRODUCTION

OSM data consist of the three entity types *node, way*, and *relation*. For referencing, each entity type has an ID. There are also additional attributes specifying a contributor and a version.

Nodes have values for *longitude* and *latitude*, thus they model points on the globe. Ways compose lines between points by specifying ordered lists of node references. Relations, in turn, may reference both ways and nodes. Therefore relations can model complex geographic features as polygons with holes as well as specify, e.g., a center point for displaying pins on the map. Relations may also reference other relations assembling abstract entities that span several 'real things' such as universities with multiple, wide-spread buildings or groups of islands. Finally, nodes, ways, and relationships can hold an arbitrary number of tags. These key-value pairs specify names, categories, address elements, house number ranges, data sources, speed limits, and other attributes of real-world features that the entities model.

Because of the structure of data, address elements are often spread across different entities. For example, a node might only be tagged with a house number, while the way that references this node only holds the street name information. The way, may be covered by a relation that represents the postal code area. However, the relation not necessarily references the way. Therefore, to offer a geocoding service, addresses need to be assembled out of these OSM entities first.

Nominatim is an open source system that builds on top of OSM data to provide a *geocoding service*. That means, Nominatim resolves named locations (full addresses, or named areas) into latitudes and longitudes of their whereabouts. Nominatim is implemented in multiple programming languages and builds on top of a *PostGIS* enabled *PostgreSQL* data base. It offers a pipeline that parses, assembles, and indexes OSM data. Unlike document stores, Nominatim also precomputes result ranks at indexing time, independent of queries.

In contrast to that, *Lucene* is a generic open source document indexing framework. Lucene supports various ranking schemes for ordering results, including TF/IDF [1][2]. TF/IDF is a formula to rank documents based on query terms and their distributions. Particularly, a document is ranked higher, if it contains query terms that are rarely used in other documents. In the same way, a document is ranked lower, if it only contains query terms that are very common. *Elasticsearch* is a document store that uses Lucene internally to index, find, and rank documents.

In general, geocoding services utilize several algorithms [3][4][5][6]. They have to parse addresses, recognize address elements, derive their meaning, and rank and filter discovered candidates. The index of address data thereby has to be both fuzzy and robust against errors in queries and source data as well as precise enough to rule out ambiguously named address parts that a query did not refer to. TF/IDF based ranking strives to fulfill these requirements with unstructured documents. Because of that, Nominatim is compared to the generic document store Elasticsearch in this paper.

II. EXPERIMENT

To compare Nominatim and Elasticsearch, first Nominatim has been set up with OSM data for Europe. Next, the function Nominatim uses to assemble search results has been used to extract all available addresses. Finally, an Elasticsearch instance has been set up next to Nominatim and populated with the extracted addresses. This way two geocoders with the same addresses indexed were running next to each other: Nominatim with pre-ranked results as well as TF/IDF based Elasticsearch. Note that while Nominatim has loaded and indexed all of OSM data, only assembled addresses have been indexed in Elasticsearch. Thus, extended features of OSM and Nominatim, e.g., translated addresses, were not available in Elasticsearch.

For the first experiment, 2000 randomly selected addresses were extracted from the Nominatim database. In accordance to the set up, the addresses were from various European countries and were all indexed in both systems. From these addresses queries with exact addresses have been generated first. The

 ISBN: 978-1-61208-383-4

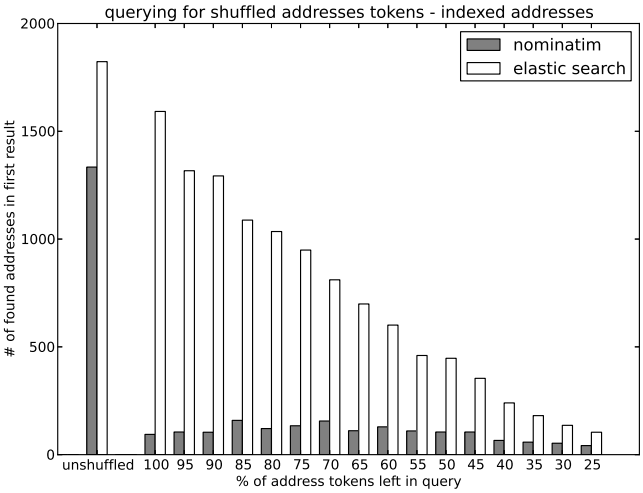

Figure 1. Nominatim and Elasticsearch on full and partial addresses.

next set of queries contained shuffled address tokens. This simulated queries that do not adhere to an address format. Finally in steps of 5% random address tokens have been removed from the queries, simulating incomplete requests. A request was regarded as successfully served, if the first result of the response was the address the query has been generated from. The number of successful requests of the respective systems is displayed in Figure 1.

The second experiment used 1000 addresses of German pharmacies. Only addresses indexed in both systems have been used. To count successful requests, all addresses have been geocoded with Google's geocoder. Because Google data and OSM data differ, different coordinates for same results have been expected. Therefore distances of the first result to Google's geocoordinates have been grouped into buckets. The buckets *within 100m, 100m-1000m, further than 1000m*, and *no result* have been used. For a general purpose geocoder, only the first bucket should be treated as a successful result. Similar to the first experiment, requests with full addresses as well as requests with only 75% of the address tokens have been stated. Table I enumerates the results.

TABLE I.
NOMINATIM AND ELASTICSEARCH ON PHARMACY ADDRESSES.

full addresses	*within 100m*	*100m-1000m*	*further 1000m*	*no result*
Nominatim	946	0	0	54
Elasticsearch	1000	0	0	0
75% address	*within 100m*	*100m-1000m*	*further 1000m*	*no result*
Nominatim	399	13	0	588
Elasticsearch	988	11	1	0

Note that addresses used in the first experiment have been generated by Nominatim. They contained many additional address elements stating administrative areas that are not commonly mentioned in German postal addresses. Pharmacy addresses where of that common type: They only contained the four address elements house number, street, city, and postal code. This is why for the second experiment only addresses with 75% address tokens have been used next to the full addresses: Most of them lacked exactly one token which made up an entire address element. Also some of the chosen pharmacies were already present as POIs in the OSM data set in addition to their addresses. Still, because the pharmacies' addresses have been used to create requests, the presence or absence of a POI in OSM has not influenced the result.

III. RESULTS

Figure 1 shows that Nominatim is incapable to deal with shuffled address tokens: it recognizes only a small fraction of the addresses that Elasticsearch can find. While more missing address tokens lead to less accurate results of Elasticsearch proportionally, the performance of Nominatim at first increases slightly. Because less tokens can be shuffled in less ways, Nominatim reaches it's sweet spot at ca. 70% of address tokens being queried. After that addresses become less distinctive and Nominatim decreases in accuracy in the same way as Elasticsearch. Also, already for full addresses Elasticsearch outperforms Nominatim.

According to Table I, Nominatim fails to geocode 54 full addresses. All full addresses are resolved by Elasticsearch to 100m correctly. Again, there is a big drop in Nominatim's performance for incomplete addresses with shuffled tokens – only 399 addresses are still found, while Elasticsearch manages to geocode 988 addresses into the *within 100m* bucket.

IV. CONCLUSION

The experiments showed that Nominatim is tightly coupled to specific address formats. This is a strong limitation, as there are contradicting address formats world wide [7]. It is also clear that preranking results independent of queries leads to less accurate results. It is worth noting that the experiments only took queries for street addresses with house numbers into account. The actual ratio of queries for named areas should be incorporated into benchmarks of live services.

Elasticsearch has proven to be more robust against shuffled and dropped address tokens. Because of that, for geocoding addresses Elasticsearch populated with addresses assembled by Nominatim yields better results than Nominatim alone. It is also clear that TF/IDF ranking is more suitable for geocoding than precomputing result ranks. Obviously, the actual performance of a geocoding system highly depends on the actual queries. Still, Elasticsearch can be used as a solid base line when developing and comparing geocoding algorithms and indexes.

REFERENCES

[1] G. Salton and C.-S. Yang, "On the specification of term values in automatic indexing," Journal of documentation, vol. 29, no. 4, 1973, pp. 351–372.

[2] G. Salton, C.-S. Yang, and C. T. Yu, "A theory of term importance in automatic text analysis," Journal of the American society for Information Science, vol. 26, no. 1, 1975, pp. 33–44.

[3] J. Fitzke and R. Atkinson, "Ogc best practices document: Gazetteer service-application profile of the web feature service implementation specification-0.9. 3," Open Geospatial Consortium, 2006.

[4] D. Goldberg, J. Wilson, and C. Knoblock, "From text to geographic coordinates: The current state of geocoding," URISA-WASHINGTON DC-, vol. 19, no. 1, 2007, p. 33.

[5] D. Yang, L. Bilaver, O. Hayes, and R. Goerge, "Improving geocoding practices: evaluation of geocoding tools," Journal of Medical Systems, vol. 28, no. 4, 2004, pp. 361–370.

[6] L. Can, Z. Qian, M. Xiaofeng, and L. Wenyin, "Postal address detection from web documents," in Web Information Retrieval and Integration, 2005. WIRI'05. Proceedings. International Workshop on Challenges in. IEEE, 2005, pp. 40–45.

[7] K. Clemens, "Automated processing of postal addresses," in GEOProcessing 2013, The Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services, 2013, pp. 155–160.

 ISBN: 978-1-61208-383-4