Deriving Spelling Variants from User Queries to Improve Geocoding Accuracy

Konstantin Clemens
Technische Universität Berlin, Service-centric Networking, Germany

Keywords: Geocoding, Postal Address Search, Spelling Variant, Spelling Error, Document Search.

Abstract: In previous research, to mimic user queries with typos and abbreviations, a statistical model was used. It was trained to generate spelling variants of address terms that a human would use. A geocoding system enhanced with these spelling variants proved to yield results with higher precision and recall. To train the statistical model, thus far, user queries and their expected results were required to be linked with each other. Such training data is very costly to obtain. In this paper, a novel approach to derive such spelling variants from user queries alone is proposed. A linkage between collected user queries and result addresses is no longer required. The experiment conducted proves that this approach is a reasonable way to observe, derive, and index spelling variants too, allowing to measurably improve the precision and recall metrics of a geocoder.

1 INTRODUCTION

Digital maps and digital processing of location information are nowadays widely used. Besides various applications for automated processing of location data, as described in (Can et al., 2005), (Sengar et al., 2007), (Borkar et al., 2000), or (Srihari, 1993), human users rely on computers to navigate, store, retrieve, and display location information. Nevertheless, internally, computers reference locations through a coordinate system such as WGS84 latitude and longitude coordinates (National Imagery and Mapping Agency, 2004). Humans, on the other hand, refer to locations by more comprehensible addresses or common names. The process of mapping such names or addresses to their location on a coordinate system is called *geocoding*.

There are two phases to this error-prone process (Fitzke and Atkinson, 2006), (Ge et al., 2005), (Goldberg et al., 2007), (Drummond, 1995): First, the geocoding system has to parse the user query and to derive the query intent, i.e., the system needs to understand which address entity the query refers to. Only then, the system can look up the coordinates of the entity and return that result. The latter step is a mere lookup and only complex from the perspective of collecting correct data and keeping it up-to-date. The first step, on the other hand, is a non-trivial task algorithmically, especially when considering the human factor: Parts of the address might be misspelled or abbreviated by users in a non-standard way. Also, while postal addresses seem structured and as they adhere to a well-defined format, (Clemens, 2013) shows that each format is only valid within a specific region. Considering addresses from all over the world, address formats often contradict to each other, e.g., expecting a house number before or after the street name. No pattern exists that all queries would fit in. In addition to that, human users may not adhere to a format, leaving names of address elements out or specifying them in an unexpected order. Similarly, humans might specify spelling variants of terms by misspelling or abbreviating them. Such incomplete and shuffled queries with spelling variants are often ambiguous, as similar or same names are reused for different addresses or different address parts. Various algorithms can be employed to mitigate these issues. Even with the best algorithms at hand, however, a geocoding service can only be as good as the data it builds upon, as understanding the query intent is not leading to a good geocoding result if, e.g., there is no data to return.

Many online geocoding services as those offered by Google (Google, 2017), Yandex (Yandex, 2017), Yahoo! (Yahoo!, 2017), HERE (HERE, 2017), or OpenStreetMap (OpenStreetMap Foundation, 2017b) are easily accessible by the end user. Because most of these systems are proprietary solutions, they neither reveal data nor algorithms used. This makes it hard to compare distinct aspects of such services. An ex-

Clemens, K.
Deriving Spelling Variants from User Queries to Improve Geocoding Accuracy.
DOI: 10.5220/0007685700530059
In *Proceedings of the 5th International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2019)*, pages 53-59
ISBN: 978-989-758-371-1

ception to that is OpenStreetMap: The crowd-sourced data is publicly available for everyone. Open-source projects like Nominatim (OpenStreetMap Foundation, 2017a) provide geocoding services on top of that.

In this paper, data from OpenStreetMap is indexed in the document search engine Elasticsearch (Elastic, 2017) - a method that has proven to work well as a geocoding system in (Clemens, 2015a) and (Clemens, 2015b). This system is evaluated for its accuracy as a baseline. Next, an enhancement step is executed that is expected to increase the precision and the recall of the geocoding system: Spelling variants – typos, abbreviations, etc. – are extracted from queries that produce a result with a good-enough score; these spelling variants are then indexed in the system as part of the respective address. The enhancement is repeated multiple times, after each execution, the system is evaluated again for precision and recall.

2 RELATED WORK

This paper builds on work done in (Clemens, 2018). There, the geocoding system was enhanced with spelling variants too; a statistical model was trained from real user queries and their clicks on geocoding results to generate spelling variants similar to those that a user would do. The statistical model assumed that some spelling variants are used by humans repeatedly while others are never appearing in user queries. Hence, computing and indexing specific spelling variants was adding the necessary flexibility to support common spelling variants without allowing spelling variants that are made rarely or not at all. Compared to allowing edit distances, the ambiguity of queries was reduced this way. Therefore, the accuracy metrics of a geocoding system enhanced with spelling variants were superior to those of a geocoding system that allowed edit distances between misspelled query tokens and address terms. In this paper the spelling variants are not generated prior to indexing using a statistical model; instead, they are computed from the queries sent to the geocoder. The experiment conducted evaluates how this approach allows deriving spelling variants for indexing. It also evaluates the impact on accuracy metrics of indexing spelling variants derived this way. Unlike in (Clemens, 2018), the approach proposed here does not need user queries to be linked to actual addresses through clicks or manual annotation of queries.

Work on comparing geocoding services has been undertaken in, e.g., (Yang et al., 2004), (Davis and Fonseca, 2007), (Roongpiboonsopit and Karimi, 2010), or (Duncan et al., 2011). Mostly, these papers focus on the recall aspect of a geocoder: Only how often a system can find the right result is taken into consideration. Also, other evaluations of geocoding systems treat every system as a black box. Thus, a system can be algorithmically strong but perform poorly in a measurements because it lacks data. Vice versa, a system can perform better than others just because of great data coverage and accuracy, despite being algorithmically simple. In this paper, the algorithmic aspect is evaluated in isolation, as all systems are set up with the same data. Also, a better way of measuring the geocoders performance is employed in this paper: The statistical model introduced in (Clemens, 2018) is based on real user queries and is used to generate erroneous, user-like queries for any given address. This allows measuring a system on a much greater number of addresses, generating user-queries with spelling mistakes and incompletely specified or shuffled address parts.

An orthogonal approach to the geocoding problem is to find an address schema that is easy to use for both: humans and computers, and is standardized in a non-contradicting way. While current schemata of postal addresses are maintained by the UPU (Universal Postal Union, 2017), approaches like (what3words, 2017), (Coetzee et al., 2008), (Mayrhofer and Spanring, 2010), (Fang et al., 2010), or (geo poet, 2017) are suggesting standardized or entirely alternative address schemata. (Clemens, 2016) showed that some alternative address schemata are simpler to remember for humans, though these are far from being adopted into everyday use.

3 IMPROVING METRICS

The *precision* of a system is the ratio of correctly retrieved results to all results retrieved. For this paper, the stricter variant *precision@1* is used. For that, only the top result of each response is considered; correct and incorrect results in second place and below are not considered for precision calculation. Similarly, while the *recall* is defined by the ratio of correct results retrieved to all the correct results available, in this paper *recall@1* is used taking only the top result of each response into consideration. Thus, the metrics denoting the performance of a geocoding system used are:

$$precision = \frac{\#responses\ with\ correct\ top\ result}{\#queries\ that\ returned\ results}$$

$$recall = \frac{\#responses\ with\ correct\ top\ result}{\#queries}$$

These stricter metrics fit well to the geocoding use case: Each query for an address has exactly one correct result; users want to rely on the top result instead of reading through the entire result list to select the best one.

The goal of this paper is to find a suitable approach to improve these metrics, specifically for queries that contain typos and abbreviations. Commonly, these problems are tackled through (i) *normalization* and (ii) supporting *edit distances* between query and result terms. For normalization (i), address terms are modified before indexing in the same way as query terms are during runtime. Normalization would, e.g., replace all occurrences of "*avenue*" with "*ave*" in every address indexed and every query received. This way, it becomes irrelevant which of the two versions of the term are part of a query: Internally the lookup will always use the normalized version that also was indexed. This mechanism of normalization, however, can only handle the most common abbreviations. Non-standard abbreviations that are not known up front cannot be normalized away. Edit distances (ii) can tackle some of such cases together with some spelling variants or accidental typos. The edit distance or the *Levenshtein distance* (Levenshtein, 1966) between two strings is the number of edits one has to apply to one of them to get the other. The terms "*av*" and "*ave*", e.g., have an edit distance of one, as one single edit – appending an "*e*" – converts the first term into the second one. Thus, if addresses are indexed so that looking them up with terms with an edit distance of one is possible, a query with the non-standard abbreviation "*av*" will retrieve addresses that contain the term "*avenue*" too, given that the normalized version "*ave*" was indexed. Additionally, typos like "*Nw Yorg*" still can retrieve addresses in "*New York*", as both terms are only one edit away from their correctly spelled counterparts.

Elasticsearch (Elastic, 2017) is a generic document search engine that supports retrieving documents by terms that are up to two edits apart from the indexed terms. It also relies on importance weighting of the query terms using TF/IDF (Salton et al., 1975) or BM25f (Robertson et al., 2004) formula to score documents. Therefore, it performs well with query formats that contain incomplete addresses and shuffled address parts as has been shown in (Clemens, 2015a).

In (Clemens, 2018), an alternative to edit distances and normalization has been evaluated and proven to be superior. There, a log of real user queries and their clicks on search results were used to create statistical models capable of two things: Given a term, such a model can compute spelling variants users might use for that term, together with the probability of each such variant. Also, given a fully qualified address, it can generate queries in a format a user would state, i.e., it generates queries with such address parts and in such order like a human would compose the query. Again, generated query formats come with their probability of being used by a human user. In that paper, one such model was used to generate spelling variants for address terms and index them along with the actual address. That approach yielded better results than allowing edit distances.

To train the statistical models, user queries and real address data were linked by user clicks, assuming that users would select by a click the result that they queried for. This paper tries to avoid the need for user queries being linked to addresses, by entirely removing the dependency on a trained statistical model. Instead, spelling variants are derived from queries sent to the system and results returned by it. In other words, the geocoding service, of which we want to improve the metrics, functions as a model that defines which spelling variants are to be indexed for which documents. A model from (Clemens, 2018) is only used to generate training and test queries at scale.

While it is understood that geocoding itself is error-prone, the hypothesis is that it is possible to reduce the problem of false results well enough to be neglectable. It is expected that indexing spelling variants derived from queries will produce an observable increase in precision and recall metrics.

4 EXPERIMENT

To conduct the experiment, addresses needed to be indexed in Elasticsearch. For that, a Nominatim (OpenStreetMap Foundation, 2017a) geocoder has been set up. Nominatim is an open source geocoder that builds on top of a PostGIS (PostGIS, 2017) enabled PostgreSQL (PostgreSQL, 2017) database. It comes with a long-running process to compile addresses out of OpenStreetMap data into its PostgreSQL data base. From there, all address entities can be extracted with their IDs, latitude, longitude and address text. This way two geocoders – Elasticsearch and Nominatim – containing the exact same data are made available. Nominatim will, therefore, function as a baseline too, next to an Elasticsearch index that has not been enhanced with spelling variants.

It is worth noting that, like all crowd-sourced data, OpenStreetMap data is not fully correct and consistent. Some addresses have false address parts in them, others are plain wrong. Some addresses are present multiple times in the data. Also, OpenStreetMap data

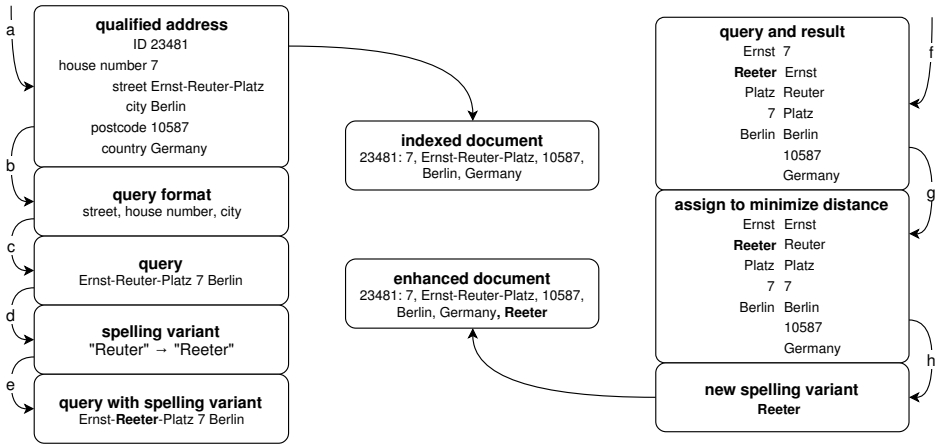

Figure 1: Example flow of generating user-like queries with spelling variants (left), address document before and after enhancing (center), and how spelling variants are derived from the queries and results to enhance indexed addresses (right). (a) qualified address picked from the data (b) query format chosen (c) query without spelling variants generated (d) spelling variant chosen for random query term (e) query with spelling variant (f) same query and correct result tokenized (g) terms aligned so that the sum of the edit distances is minimal (h) spelling variant derived.

contains points of interest that sometimes share one same address. While, for this experiment, exact address copies have been deduplicated, all their IDs have been aggregated. This way, results for duplicate addresses contained the IDs of all the duplicates. Yet, all further inconsistencies were left in the data. In sum, all OpenStreetMap data for Europe has been indexed in Nominatim. All compiled addresses with house number have been extracted from Nominatim and indexed in Elasticsearch as plain documents.

To measure and enhance the performance of geocoding systems with user-like queries, a statistical model from (Clemens, 2018) has been used. First, 50,000 addresses indexed in the system were selected uniformly at random. Next, the model's capability to generate queries in user-like formats along with their probability has been used to create queries for the selected addresses. Queries were chosen so that their formats were distributed in accordance with the formats probabilities: Queries with likely to be generated formats were picked more often than those with unlikely ones. In the last step, the same model was used to replace some of the query terms with a spelling variant. Like with query formats, the probability a spelling variant was selected was exactly the probability of that spelling variant. Thus, rare spelling variants were chosen rarely, while common spelling variants were picked often. For each address, four queries, each with zero, one, two, or three spelling variants, were generated. In total, 200,000 queries were generated from the set of 50,000 randomly selected addresses. The query formats and the spelling variants in these queries thereby varied in the same way as they varied in real user queries the model was trained on. However, because the queries were generated from known addresses, the correct response for each of them was known. The query set was shuffled randomly and used as an *enhancement set* – the set of queries that were used to enhance the system. A subset of the enhancement set, 40,000 randomly chosen queries, was selected as the *test set*. The test set was large enough to contain a representative distribution of the training set with regards to spelling variants present, their number, and query formats. It was used to measure the precision and recall metrics of geocoding systems under test, and how they evolve after indexing derived spelling variants. The test set is required to be part of the enhancement set, as only addresses from results for queries in the enhancement set are enhanced. At the same time, the enhancement set needs to be much larger to assert that enhancing additional addresses does not degrade the metrics of the system.

The experiment was conducted as follows: The test set was used to assess the precision and the recall of the geocoding system based on Elasticsearch with plain address documents. Because the expected result for each query was known upfront, and because IDs of duplicate addresses were aggregated already, this was a simple task. Afterward, the training set was used to enhance the index with spelling variants. To mitigate the risk of incorrect results, two simple thresholds were established: For one thing, the top result needed to match to the query well enough. The result score, as computed by Elasticsearch, counts the number of query terms matching address terms, weighting each match with the TF/IDF weight of the respective term. It therefore weights important terms that ap-

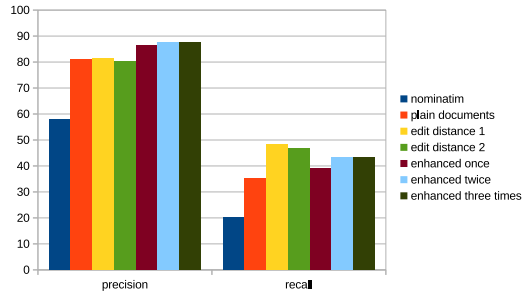

Figure 2: Precision and recall of the geocoders Nominatim, Elasticsearch with plain address documents, Elasticsearch with allowed edit distances of one and two, as well as of Elasticsearch enhanced with derived spelling variants after one, two, and three iterations.

pear in few addresses higher than common terms that appear in many. For the other thing, the relative distance between the scores of the first and the second results was calculated. In few iterations on a small random subset of the enhancement set the two thresholds were chosen manually. The goal was to make sure that as many queries as possible lead to derived spelling variants, while keeping the ratio of spelling variants derived of incorrect results at ca. 10%. The first threshold for the minimal score of the top result was set to 22. For the second threshold the score of the second result was required to be less than 96% of the score of the first result. If only one result was present in the response, the second threshold was not evaluated. In such scenarios spelling variants were derived if only the first threshold was satisfied.

To match queries with spelling variants at all, edit distances of two were allowed during the enhance step. Whenever a query and a result set met the thresholding criteria, the spelling variants were derived: First, the Levenshtein distance between each query term and each address term of the top result were computed. Then, the Hungarian method (Kuhn, 1955) was used to assign query terms to address terms of the top result while minimizing the sum of all Levenshtein distances between assigned terms. Finally, each query term that did not exactly match its assigned address term was simple to collect spelling variant for that address. The original address a query was generated from was known up front so that, again, it was a simple task to assess how many incorrect results were used to derive spelling variants on. These cases have been counted and categorized by the number of spelling variants in the query for statistical reasons; that information, however, has not been used to affect the behavior during the enhancement step: All derived spelling variants were used to update the indexed addresses.

After enhancing the index with spelling variants computed this way, another measurement step was executed. To see if more spelling variants can be derived by repeating this process, the iteration has been repeated three times. Additionally, the test set was used to measure precision and recall of Nominatim and Elasticsearch with allowed edit distances of one and two so that any improvement achieved through spelling variants can be compared to that achieved by allowing edit distances.

Figure 1 illustrates how user-like queries with spelling variants are generated and how such spelling variants are derived to enhance the index for future querying. In the center of the figure, the same address document is presented right after indexing, as well as right after enhancing.

5 RESULTS

The chart in Figure 2 summarizes the precision and recall metrics measured. Nominatim performs much worse than any other system in both metrics. This behavior is in line with measurements made in previous papers. A slight increase in precision from 81.14% to 81.37% when an edit distance of one is allowed is in line with previous measurements too. Also observed in previous work: Allowing an edit distance of two reduces the precision to 80.42% – a value lower than geocoding with Elasticsearch with no edit distances allowed. Doing so allows a large portion of query terms to match to wrong addresses causing more harm than good here. The best recall achieved is 48.32% with address documents indexed in Elasticsearch with an allowed edit distance of one. Again, increasing the allowed edit distance to two only has a negative effect reducing the recall to 46.71%. Clearly visible on the chart, too, is that deriving and indexing spelling variants increases the precision further than allowing edit distances. After the first execution of the enhancement step, precision grows by over 6% from 81.14% with plain Elasticsearch to 86.38%. That is more than allowing an edit distance of one, which achieves a precision of just 81.37%. Though by smaller extent, recall too grows from 35.38% to 39.22% when enhancing addresses with spelling variants. The chart also shows that the second and the third iteration of the enhancement process do not further improve the metrics. In fact, after the third iteration precision starts regressing to 87.51% from 87.54% after the second one.

Figure 3 is a chart presenting the ratio of queries for which spelling variants could be derived, plotted next to the ratio of spelling variants derived from incorrect results. For queries with zero spelling vari-

ants, unsurprisingly, almost no spelling variants can be derived. Those few variants that the system was able to derive, however, are almost exclusively based on incorrect results. With one single spelling variant in the query, the picture turns around: 29.29% of queries had results passing the threshold criteria and leading to derived spelling variants. Only 6.29% of these were based on incorrect results. The more spelling variants are present in a query, the more derivations are made based on incorrect results, and the fewer results are passing the thresholds and allow deriving spelling variants.

6 CONCLUSION

The experiment proves: With two simple thresholds and an off-line enhancement process, the accuracy of a geocoding system can be enhanced measurably. Recorded queries just need to be replayed against the existing index while allowing edit distances not allowed during geocoding. From queries and results spelling variants can be derived accurately enough so that when they are indexed, precision and recall metrics are improved. Especially the precision is improved with this approach more than with allowing edit distances. Though to a smaller extent, recall is growing too. The experiment also shows that this process is not an iterative one: Queries that have been used to enhance the index once, should not be used again. For a production system, however, that is not a problem as new queries flowing in can be recorded continuously. A look into the correctness of results for which spelling variants are derived shows that this approach works best with queries with one single spelling variant in them. Disregarding queries without any spelling variant, that is exactly the kind of queries that a production system will likely experience the most too.

The proposed approach is executed off-line, first using a live system to collect spelling variants and enhancing it in a separate step. That makes it easy to integrate into existing geocoding solutions. Similarly, a geocoding system could continuously augment itself with spelling variants derived from every query in real time. As geocoding is an error-prone process, it is clear that some spelling variants will be derived based on incorrect results. The experiment proves, however, that the fraction of such cases is not harming more than enhancing the index with spelling variants helps, even if queries with two or three spelling variants make 50% of all queries encountered.

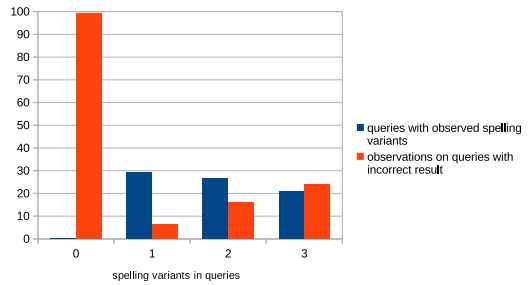

Figure 3: Ratios of queries allowing deriving spelling variants and ratios of incorrect results spelling variants are derived on, by the number of spelling variants in queries.

On the flip side, the proposed approach does not obsolete the need to standardize. While the model in (Clemens, 2018) derived commonly used abbreviations as spelling variants from user clicks, in the proposed approach only spelling variants in an edit distance of two can be derived. Most abbreviations are more than two edits apart from the fully spelled out word, like, e.g., "*st*" and "*street*".

7 FUTURE WORK

An angle to further improve the method is to fine-tune the thresholds used. Language- or region-specific thresholds might reduce the fraction of spelling variants derived from incorrect results while growing the fraction of queries for which spelling variants can be derived. Considering additional aspects of responses, besides the score of the top result, could be beneficial too.

A clear lack of the proposed method is to extrapolate derived spelling variants to addresses other than those in results of queries. The mechanism derives spelling variants per query and result, only enhancing the result address itself, even if the spelling variant derived was for a region or city name. All other addresses in that region or city do not benefit from the derived spelling variant.

Another enhancement possibility is the time factor. Some spelling variants are made often and continuously, others are only made once. The geocoding system could keep track of the derived spelling variants and weight them based on the number of their occurrences. Rarely made spelling variants would have a smaller impact compared to those made often, possibly reducing the negative impact of falsely derived spelling variants. With the same goal, spelling variants that were not observed for some time could expire and be removed from the documents.

REFERENCES

Borkar, V., Deshmukh, K., and Sarawagi, S. (2000). Automatically extracting structure from free text addresses. *IEEE Data Engineering Bulletin*, 23(4):27–32.

Can, L., Qian, Z., Xiaofeng, M., and Wenyin, L. (2005). Postal address detection from web documents. In *International Workshop on Challenges in Web Information Retrieval and Integration, 2005. (WIRI'05)*, pages 40–45. IEEE.

Clemens, K. (2013). Automated processing of postal addresses. In *GEOProcessing 2013: The Fifth International Conference on Advanced Geograhic Information Systems, Applications, and Services*, pages 155–160.

Clemens, K. (2015a). Geocoding with openstreetmap data. *GEOProcessing 2015: The Seventh International Conference on Advanced Geograhic Information Systems, Applications, and Services*, page 10.

Clemens, K. (2015b). Qualitative Comparison of Geocoding Systems using OpenStreetMap Data. *International Journal on Advances in Software*, 8(3 & 4):377.

Clemens, K. (2016). Comparative evaluation of alternative addressing schemes. *GEOProcessing 2016: The Eighth International Conference on Advanced Geograhic Information Systems, Applications, and Services*, page 118.

Clemens, K. (2018). Enhanced address search with spelling variants. *Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM*, pages 28–35.

Coetzee, S., Cooper, A., Lind, M., Wells, M., Yurman, S., Wells, E., Griffiths, N., and Nicholson, M. (2008). Towards an international address standard. 10th International Conference for Spatial Data Infrastructure.

Davis, C. and Fonseca, F. (2007). Assessing the certainty of locations produced by an address geocoding system. *Geoinformatica*, 11(1):103–129.

Drummond, W. (1995). Address matching: Gis technology for mapping human activity patterns. *Journal of the American Planning Association*, 61(2):240–251.

Duncan, D. T., Castro, M. C., Blossom, J. C., Bennett, G. G., and Gortmaker, S. L. (2011). Evaluation of the positional difference between two common geocoding methods. *Geospatial Health*, 5(2):265–273.

Elastic (2017). Elasticsearch. https://www.elastic.co/products/elasticsearch.

Fang, L., Yu, Z., and Zhao, X. (2010). The design of a unified addressing schema and the matching mode of china. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010*. IEEE.

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

Ge, X. et al. (2005). Address geocoding.

geo poet (2017). http://geo-poet.appspot.com/.

Goldberg, D., Wilson, J., and Knoblock, C. (2007). From Text to Geographic Coordinates: The Current State of Geocoding. *URISA Journal*, 19(1):33–46.

Google (2017). Geocoding API. https://developers.google.com/maps/documentation/geocoding/.

HERE (2017). Geocoder API Developer's Guide. https://developer.here.com/rest-apis/documentation/geocoder/.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Mayrhofer, A. and Spanring, C. (2010). A uniform resource identifier for geographic locations ('geo'uri). Technical report, RFC 5870, June.

National Imagery and Mapping Agency (2004). Department of Defense, World Geodetic System 1984, Its Definition and Relationships with Local Geodetic Systems. In *Technical Report 8350.2 Third Edition*.

OpenStreetMap Foundation (2017a). Nomatim. http://nominatim.openstreetmap.org.

OpenStreetMap Foundation (2017b). OpenStreetMap. http://wiki.openstreetmap.org.

PostGIS (2017). http://postgis.net/.

PostgreSQL (2017). http://www.postgresql.org/.

Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple BM25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 42–49. ACM.

Roongpiboonsopit, D. and Karimi, H. A. (2010). Comparative evaluation and analysis of online geocoding services. *International Journal of Geographical Information Science*, 24(7):1081–1100.

Salton, G., Yang, C.-S., and Yu, C. T. (1975). A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, 26(1):33–44.

Sengar, V., Joshi, T., Joy, J., Prakash, S., and Toyama, K. (2007). Robust location search from text queries. In *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, page 24. ACM.

Srihari, S. (1993). Recognition of handwritten and machine-printed text for postal address interpretation. *Pattern recognition letters*, 14(4):291–302.

Universal Postal Union (2017). http://www.upu.int.

what3words (2017). what3words. https://map.what3words.com/.

Yahoo! (2017). BOSS Geo Services. https://developer.yahoo.com/boss/geo/.

Yandex (2017). Yandex.Maps API Geocoder. https://tech.yandex.com/maps/geocoder/.

Yang, D.-H., Bilaver, L. M., Hayes, O., and Goerge, R. (2004). Improving geocoding practices: evaluation of geocoding tools. *Journal of medical systems*, 28(4):361–370.