Graphetteer – A conceptual model for a graph driven gazetteer

Manuel Bär
MSc GIScience, University of Zürich
Winterthurerstrasse 190
CH-8057 Zürich
+41 44 635 51 11
manuel.baer@uzh.ch

ABSTRACT

In this paper, I describe the state of the art in geographic information retrieval (GIR) and present a conceptual model for a fast, scalable graph based gazetteer, to be used in various domains of geographic information retrieval. The majority of available gazetteers are stored as relational databases, providing easy access but limiting the complexity of queries. In this paper I create and discuss a conceptual model of a gazetteer using a Neo4J graph database, which allows queries of high complexity to be efficiently run with low response time. A Neo4J graph database is installed onto a Linux server and is populated with a limited dataset to illustrate different aspects of the conceptual model. Queries are written using the generic Neo4J querying language CYPHER. Special focus is given to storing and querying semantic relationships between locations to allow queries about vague spatial relations on different scales. The proposed conceptual model has the potential to solve various state of the art limitations of available gazetteers discussed in this paper, but is accompanied by new limitations needing further attention.

Categories and Subject Descriptors

E.1 [**Data Structures**]: Graphs and networks – *Graph database, Neo4J*; H.2.3 [**Languages**]: Query Languages – *Neo4J Cypher*; H.2.8 [**Database Applications**] – *Graph database, gazetteer, spatial relationships*; H.3.0 [**General**] – *Geographic information retrieval*; H.3.3 [**Information Search and Retrieval**]: Retrieval models – *conceptual model, gazetteer, geographic information retrieval*; H.3.4 [**Systems and Software**]: Performance evaluation – *relational database, graph database*

General Terms

Geographic Information Retrieval, Gazetteer, Conceptual Model, Graph Database, Neo4J, Cypher, State of the Art, Design, Theory, Meta-Review

Keywords

Gazetteer, Graph Database, Neo4J, Cypher, GeoNames, GIR, Query, Spatial, Semantics, Relationships

University of Züriich, January 15, 2016, Zürich, Switzerland.

1. INTRODUCTION

Every day an enormous number of searches are performed on the internet [1], mostly by querying search platforms like Google[1] or Bing[2]. Considering that 13-15% of these queries contain toponyms [2], geographic information retrieval (GIR) is becoming an important scientific discipline. Dealing with spatial queries is accompanied by various fundamental challenges including the detection of toponyms in unstructured data, disambiguating place names, interpreting vague place names or vague spatial language, spatially indexing documents and creating effective user interfaces [2]. In the field of geographic information retrieval collections of placenames, typically gazetteers, play a central role in providing a concept for solving or circumnavigating several of the mentioned fundamental challenges. Disambiguating names and comparing document and query footprint are among the key operations where a gazetteer enjoys a widespread use. Typically gazetteers contain triplets of place names, geographic footprints and feature types [3]. They are often seen as a key component in geographic information systems and geographic search facilities [2]–[4], but "lack the capabilities to fully integrate […] vernacular geographic information, as well as to support complex queries" [3]. Geographic information queries are often of higher complexity and contain more words [5] than the non-spatial counterpart and have been characterized as a triplet of a topic of interest combined with a place name and a spatial preposition: <theme> <spatial relationship> <location> [2] or as [6] argues a <what, relation, where> triple.

Alongside the evolution of geographic information retrieval, there has been an increase in the use of graph databases to model complex relationship structures [7]. Modern services handling large datasets of information (e.g. Facebook[3], Google[4], Twitter[5]…) have reached the limitations of relational databases and have migrated their systems to an underlying graph database system [7], [8]. These database systems have high computational advantages over conventional relational databases, especially for

[1] www.google.com
[2] www.bing.com
[3] www.facebook.com
[4] www.google.com
[5] www.twitter.com

performing complex or relational queries [9]. In particular, a graph database is extremely adaptive and excels at scaling with an application. Queries to a relational database take longer to process with growing database size, whereas queries to a graph database are more or less constant, because the query only needs to be executed in a subsection of the graph and not on the whole graph [7], [9].

In this paper, I will compare the use of a Neo4j Graph Database to a relational database as a foundation for a next generation gazetteer structure. I not only aim to show performance differences but will explore query building and especially highlight the possibilities of a new gazetteer structure using the power of the node-relation structure of graph databases by presenting a conceptual model. This paper will revolve around the following hypotheses':

- Graph databases are more suited to capture the growing connectedness and relationships between places than relational databases
- Graph databases significantly outperform relational databases on medium to large datasets
- Graph databases are viable as a data storage for gazetteers

In the first section I have presented an introduction to this topic which is followed by the second section comprised of the state of the art and meta-review of the comparisons of graph databases with relational databases. The conceptual model including a discussion regarding structure, query formulation and limitations for a next generation gazetteer using NEO4J graph database is found in the third section which is followed by a fourth section containing the discussion. Finally, the paper ends with a conclusion and further research.

2. STATE OF THE ART

2.1 Information Retrieval

Information retrieval as the task of getting query relevant data is not a new discipline and is argued to date back to the third century BC Greek poet Callimachus, who organized "the works of the authors […] in alphabetical order […] [into the] first ever library catalogue" [10]. Information retrieval became of increasing importance in the late 18th, early 19th century, which led to the invention of "a machine that searched for a pattern of dots or letters across catalogue entries stored on a roll of microfilm" [11] by Emanuel Goldberg. The interest in information retrieval systems grew constantly through the 19th century (see [11]) enjoying an exponential increase in interest with the emergence of the internet. Information retrieval is now an essential part of modern digital societies. According to Gigaom [12] Google[6] alone recorded 114.73 billion search requests in December 2012 and as can be seen [Figure 1] every major search engine has had a substantial increase in search queries, except for Yahoo[7]. Geographic information retrieval is a rather new subcategory of information retrieval, dealing with queries which have spatial relevance. It is estimated that 13% - 15% of all queries contain

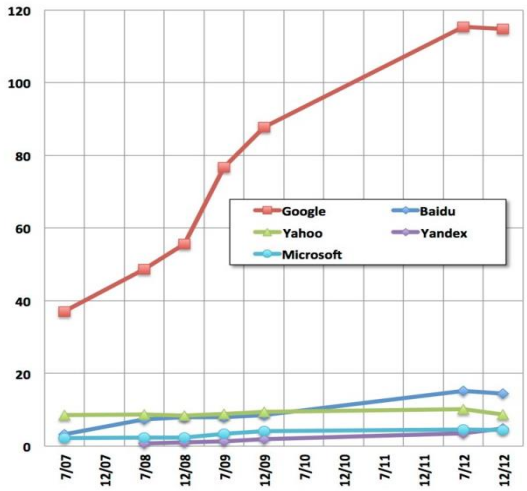

Figure 1: Number of monthly searches worldwide, in billions. Source: [1]

toponyms [2], which is verified in [5] where 18.6% of queries are found to contain geographic terms and 14.8% place names. This calls for systems to efficiently deal with geographic queries and retrieve not only contextually relevant but also spatially relevant data and documents. A large amount of research has been conducted on different strategies of geographic information retrieval [2], [5], [6], [11], [13], in which gazetteers mostly have a prominent role. Gazetteers are often one of the main building blocks for a geographic information retrieval systems [3] and are mainly used for toponym resolution [14]. "Given the modern search engine's sub-second query response time, the expectation will also be similar for GIS search engines" [15] and seeing the importance of gazetteers, it is vital for large gazetteers to have a rapid response time. Taking into account that it is of growing importance to be able to store linked data and the relationships between spatial entities, various authors [3], [16]–[19] have contributed to the discussion by presenting conceptual models or implementations of a new type of gazetteer or how to store and deal with spatial ontologies.

2.2 Graph vs Relational Databases

Realising the importance of not only efficiency but also of storing and querying linked data and handling spatial ontologies, the question of the underlying infrastructure of a gazetteer emerge. Gazetteers are predominantly stored in databases as a triple of "placenames (N), geographic footprints (F), and feature types (T)" [3]. Databases can display large differences in terms of efficiency, storage capacity [Figure 2] and query time [Figure 3Figure 2] needed for the same data, especially when dealing with large, internally linked datasets [20].

This has major impacts on user experience when using a system designed to retrieve geographically relevant information, in particular regarding toponym recognition. Various authors [7]–[9] now argue, graph databases are becoming more important, especially with growing spatially relevant data collections to be queried.

[6] www.google.com

[7] www.yahoo.com

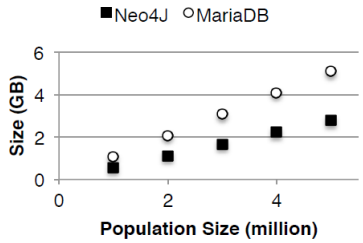

Figure 2: Database size on disk. Comparison of Neo4J (graph database) and MariaDB (relational database). Source: [20]

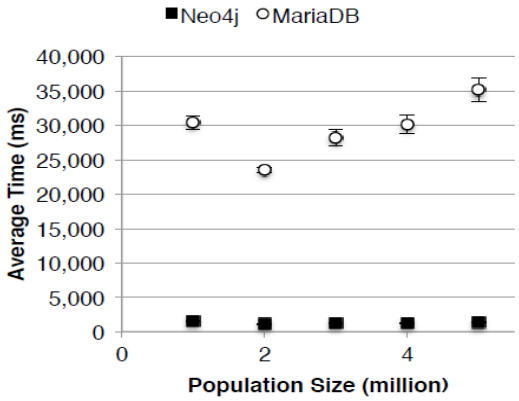

Figure 3: Time to retrieve relatives to distance 5 for 10 people, 20 warm trials. Comparison of Neo4J (graph database) and MariaDB (relational database). Source: [20]

3. GRAPHETTEER – A CONCEPTUAL MODEL

[18] summarise the limitations of online gazetteers mentioned in [21]–[23] as "(1) the limited spatial representation (a point or a rectangle) and absence of support for spatial relationships, (2) the absence of support for semantically complete, but geographically imprecise locations, such as "south of France" or "upstate New York" [and] (3) the lack of intra-urban detail, including places often mentioned in natural language text and possibly know by non-residents, such as monuments or tourist attractions". The proposed conceptual model aims to minimise mentioned limitations. Although predominantly focusing on solving or minimising limitation (1) regarding representations and relationships, the proposed model should also contribute to solving limitations (2) and (3). [18] also point out the varying predefined hierarchical feature types of different gazetteers, and the resulting strengths and weaknesses such predefined hierarchies accompany.

3.1 Structure

The proposed conceptual model of a next generation graph based gazetteer uses Neo4J as the underlying graph database. Neo4J was chosen due to the large and active community, the vast documentation and most importantly because of the structure the data is stored in. Neo4J stores nodes and relationships between said nodes. The nodes and can have one or multiple labels and similarly the relationships can have one or more relationship types [Figure 4], which presents a useful gazetteer foundation. The proposed conceptual model using a Neo4J graph database is based on an adapted version of the OMT-G schema [24] illustrated in [18] [Figure5]. The schema has been adapted to better handle complex relationships and be compatible with the Neo4J CYPHER query language.

For this conceptual model the nine predefined feature types incorporated by GeoNames[8] [25], which are the top level of 645 sub-feature types [26], are used. Due to the nature of how Neo4J is structured, mentioned feature types provide a predefined list of viable node labels. All nodes must have a label corresponding to a predefined feature type. Predefining the list of node labels minimises redundancies and enhances computational efficiency. Furthermore, topological rules can be set up to further simplify user interaction and reduce digitalisation errors (e.g. the feature type or label island can only be given to a node, which also has the relation type of being in a water body; a node with the label country cannot have the relationship type "in" pointing from country to municipality).

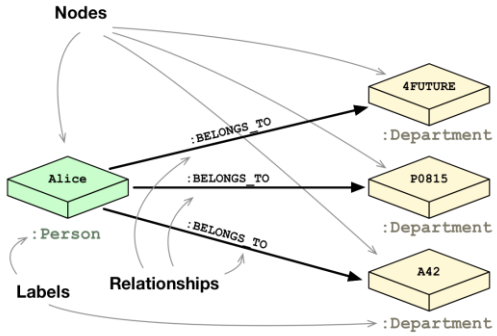

Figure 4: Neo4J database structure. Source: [32]

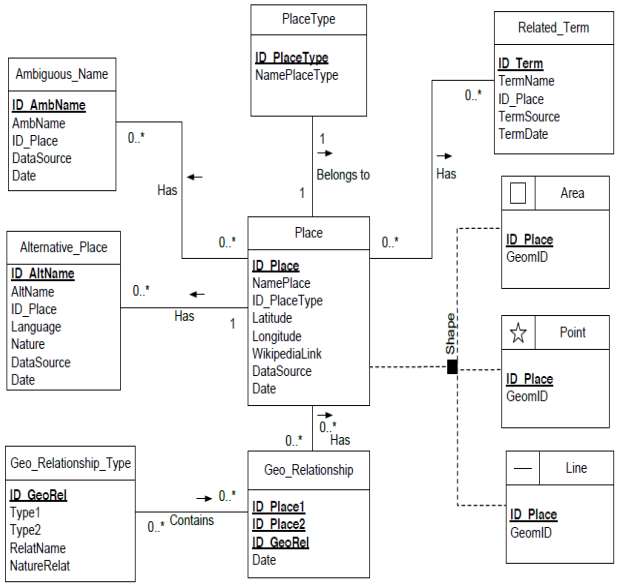

Figure 5: Gazetteer conceptual schema. Source: [18]

[8] www.geonames.org

Relationships are also predefined as to minimise redundancies and maximise query efficiency. I propose using only a minimal number of relationship types, namely: "in", "adjacent to", "ambiguous", "vernacular", "has area", "has line", "has point" and "has tags". I argue that the proposed relationship types comprise a well-rounded trade-off between complexity and useability.

3.1.1 Topological relationship types

The relationship "in" is used to determine a topological hierarchy between features (e.g. Biel is a municipality located IN the canton of Bern which is located IN the country of Switzerland which is IN the continent of Europe which is IN the world), whereas the relationship "adjacent to" has more of a topological tessellation character and is used as a vague distance relationship on different hierarchical levels (e.g. the municipal of Biel is ADJECENT TO the municipal of Nidau which is ADJECENT TO the municipalities Ipsach and Port; The country of Switzerland is ADJECENT TO the country of France) [Figure].

3.1.2 Toponym resolution relationship types

The relationship types "ambiguous" and "vernacular" are vital to perform computationally efficient toponym resolution tasks. The relationship type "ambiguous" connects all entries with the same names. When given the task of disambiguating a certain toponym, all nodes with the same name are connected. This means that the query only has to traverse along and out from the nodes and relationships connected by the "ambiguous" relation type, instead of iterating through the whole graph (e.g. If the user need is "hotels in Biel or Nidau" the query will start with one node with the name Biel and traverse along the relationship types "ambiguous". Once the query has iterated through all ambiguities, an algorithm will prioritise Biel in Switzerland and not Biel in Spain as the subject of interest, because the node Nidau is connected over only one "adjacency" relationship with the node Biel).

Vernacular name recognition and resolution are growing of increasing importance in geographic information retrieval but pose major challenges [27], [28]. Although vernacular names are often vague terms, there is "often some common agreement about the functionality of these places and their spatial relations to other places" [27]. Vernacular names can even be the predominantly used name of a specific place (e.g. in Nidau there is a park next to the lake officially called "Seematte" but predominantly called "Hundämättäli"). In this proposed conceptual modal, the relationship type "vernacular" can be used to link multiple vernacular names to an official name of a specific place, similar to the OMT-G schema [24]. Not only can this be of great importance when dealing with place name disambiguation, but I argue that it can also shed light on temporal shifts in the perception of an environment (e.g. Is downtown Zürich still perceived to have a similar geometry than 10 years ago, or has the perception of where downtown Zürich is shifted?).

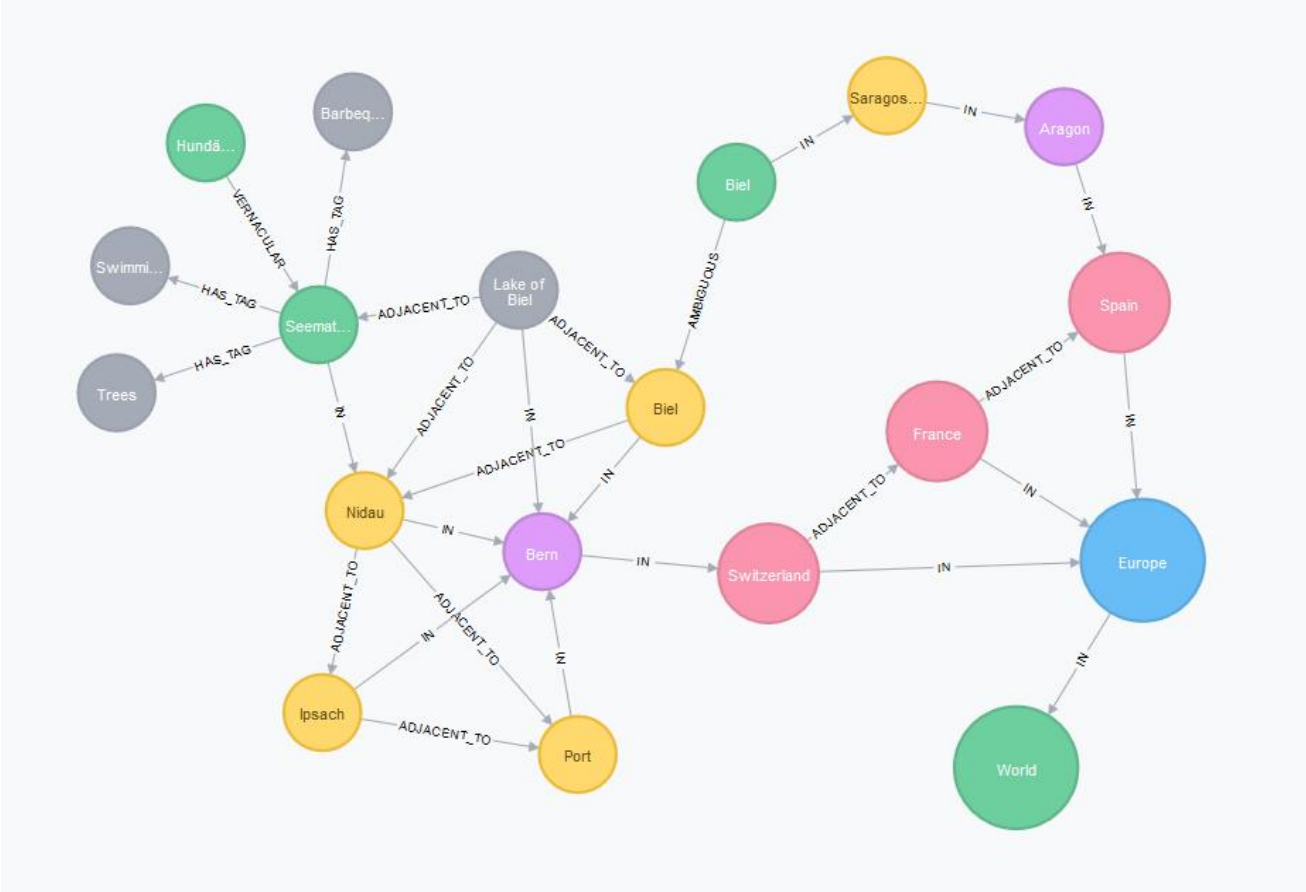

Figure 6: Example of the relationship types "in", "adjacent to", "ambiguous", "vernacular" and "has tags" in a Neo4J graph database.

3.1.3 *Spatial relationship types*

The relationship types "has area" (e.g. bounding box, convex hull, alpha shape, true boundaries), "has line" (e.g. river, road, train tracks, power lines) and "has point" (e.g. centroid, GPS point) are used to store spatial footprint information. It is to be noted, that every feature can have multiple spatial footprint relationships. The Neo4J community offers a Neo4J spatial plugin[9] allowing points, lines and polygons to be stored and more importantly to be spatially queried. With the Neo4J spatial plugin, queries such as contain, cover, covered by, cross, disjoint, intersect, intersect window, overlap, touch, within and within distance can be performed [29]. This highlights the power of Neo4J and the potential of becoming the standard underlying database for next generation gazetteers. Using the spatial plugin introduces various powerful and complex query possibilities, enabling fast and efficient querying not only of semantic relationships but also in combination with spatial queries (e.g. which municipalities are IN the canton of Bern but not further than 10km from Biel?). Not only does the spatial component add a vast variety of new query possibilities, but it also greatly improves debugging and minimises errors (e.g. An administrative unit with the relationship type "in" with a country must also spatially be "contained" in said country).

3.1.4 *User generated content relationship types*

[30] introduced the term volunteered geographic information (VGI), a special case of user generated content and stated the growing importance of and public and commercial interest in this kind of content. It is argued that user generated content or volunteered geographic information has various uses in both the commercial and the academic sector. The proposed "has tags" relationship type is used to store additional user generated, enrichening the data to not only be a collection of placenames and the spatial footprint thereof, but loading locations with additional qualities (e.g. possible tags for the node "Seematte" could include "Swimming", "Barbeque" and "Trees"). The proposed "has tags" relationship is comparable to the OMT-G schemas "Related Term" [24] table. It offers a graph database compatible structure of storing user generated content in form of tags. The underlying aim of this approach is to further broaden the types of queries which can be performed on the gazetteer, allowing queries to search through the user generated content (e.g. where can I go swimming and have a barbeque in Bern? [**Figure**]).

```
match (m)-[:IN*]->(l:ADM1{name:"Bern"})
where (m)-[:HAS_TAG]->({name:"Barbeque"})
AND (m)-[:HAS_TAG]->({name:"Swimming"})
return m
```

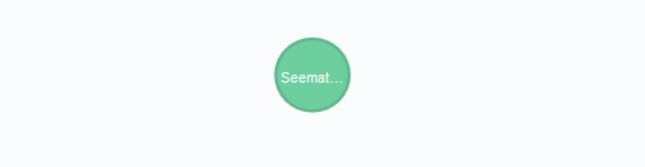

Figure 7: Example of CYPHER query to receive locations with the tags "Swimming" AND "Barbeque" in the canton of Bern and the result

[9] https://github.com/neo4j-contrib/spatial

3.2 Querying

As is suggested in the literature [2]–[4], querying a gazetteer is a vital part of geographic information retrieval. Today most gazetteers allow limited querying possibilities and are not able to handle more complex queries [3]. Complex queries however need be performed to be able to efficiently derive needed information. The possible queries are also linked to the interface, which encourages different types of query behaviour [31].

The proposed conceptual model uses a graph database as an underlying storage and thus allows highly complex querying using Neo4Js generic querying language CYPHER. Although CYPHER has a somewhat steep initial learning curve, the query statements seem more natural and closer to natural spoken language than SQL. Using the generic Neo4J syntax allows the user to traverse multiple relationships without using JOINS.

The Query "Parks in Bern" would be as simple as:

```
Match (r:Park)-[:IN*]->(n:ADM1{name:"Bern"})
return r
```

Which translates to: Return all nodes with the label "Park", which have any given number of relationships of the type "in" leading to the node with the label "first level administrative unit" and the name "Bern".

A similar query with SQL would involve querying a number of tables with multiple joins and depending on the complexity of the query would result in significantly longer processing time.

Not only specific queries but also vague queries become possible when using the proposed model. I argue that the concept of near can be described as environmental entities which, on the same scale, are no further than two "adjacency" relations apart (e.g. Biel is NEAR but not ADJECENT to Ipsach and Port [Figure]). Although simple, a such definition can lead to major improvements of efficiency, whilst having similar recall and precision to conventional models. For queries including vague cardinal directions (e.g. to the southwest of Switzerland), Neo4Js

```
$ match (n:ADM2{name:"Biel"})-[:ADJACENT_TO*1..2]->(r) return r
```

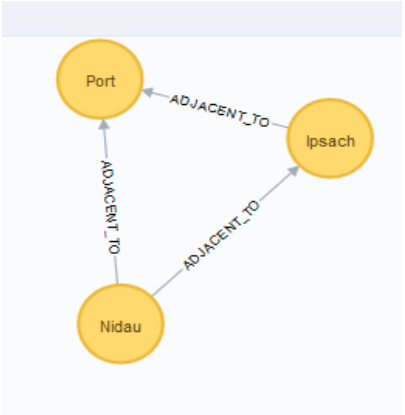

Figure 8: Example of CYPHER query to receive ADM2s "near" Biel and the result

spatial plugin in combination with node relationships can be used (e.g. Southwest of Switzerland would take Switzerland as the scale, iterate through all adjacency relationship types up to two levels and compare the spatial footprints of the nodes with the spatial footprint of the mentioned location, here Switzerland. In this example, France and Spain would be returned as probable candidate results). The notion of vague querying can be extended to fit the need of the information system by defining different concepts of vague spatial relationships. The concept of near can also be defined dynamically based on algorithmic guessing of the user need (e.g. if the Query is "Shops near Switzerland" then it is assumed that the user issuing the query is looking for shops near the border of Switzerland but not in Switzerland. To successfully return relevant information the query has to be analysed and the system must be able to determine that shops is on the street level scale and Switzerland on the country level scale, meaning the street level scale in proximity of the country border of Switzerland, but not in the country Switzerland should be taken into account).

As mentioned, most available gazetteers can only reliably process simple queries (e.g. what is where?), but in the increasingly complex and interconnected world, demand to handle complex queries is constantly growing. There is especially a need for systems to be able to handle complex queries including semantic information and relationships. User needs in regard to information retrieval frequently involve geographic terms or references [2], [5] and systems must be able to handle such needs. Using Neo4J as an underlying data structure, various complex queries can be efficiently processed (e.g. Where is a park in Bern next to a lake where I can have a barbeque? [Figure]). With growing content, the possible queries increase in complexity but processing should not lose efficiency. I argue that relationships between entities is of growing importance in geographic information retrieval and is a resource not yet fully utilised.

```
match (n:Park)-[:HAS_TAG]->(m:Tag{name:"Barbeque"}),
(n:Park)-[:ADJACENT_TO]-(r:Lake),
(n:Park)-[:IN*]->(p:ADM1{name:"Bern"})
return n
```

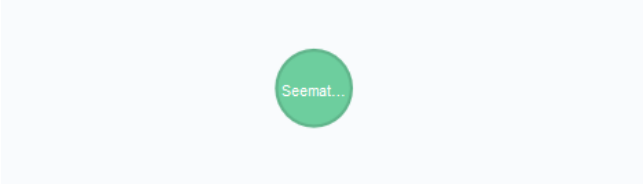

Figure 9: Example of CYPHER query to receive parks with the tag "barbeque" next to a lake in Bern

3.3 Limitations

As mentioned, the proposed conceptual model of a gazetteer with a graph database as underlying data structure has the potential to eliminate various fundamental limitations of existing gazetteers, but is accompanied by various new limitations including the strong dependency on the visual user interface, data mining and automated importing, artefact and redundancy elimination, creating or deleting nodes within an already linked portion of the graph, geometry related limitations and further development issues.

As with all interactive applications, the form, useability, design and structure of the user interface have a large influence on how intuitive the system is and on what type and to what complexity the user is encourage to formulate queries. By comparing Google[10] with Local[11] large differences not only in layout and design, but in input options are identified. Google implements one single search input field without any information about what kind of queries are permitted. Out of the perspective of geographic information retrieval Google has to process the submitted query to a greater extent than Local, because no pre query separation of contextual user need and location user need is made. Local presents the user with two separate input fields with a minimal amount of information on what type of query is expected (e.g. Input field 1: "Who or what?"; input field 2: "Where?"). These subtle differences can have a major impact on the precision, recall and complexity of a system, including the proposed conceptual model.

Another noteworthy limitation is the issue of data mining and automated import. Various different gazetteers already exist, posing the question of how available data can be imported into a new gazetteer model or how existing gazetteers can be updated to use the proposed model. By automating the import process, various methodologies must be considered to correctly link nodes with viable relationships producing a minimal number of errors.

Furthermore, it is vital for any data storage system to incorporate a pipeline to deal with artefacts and redundant data. Artefacts being pieces of information no longer connected to any other piece of data and thus mostly not regarded by queries (e.g. if the node n:Park with the name "Seematte" is deleted, the three tags belonging to said deleted node and the vernacular name will still be present but not linked to the rest of the data. This poses the question if these nodes should also be deleted; If the node Saragossa is deleted, the question arises if Biel should be given the relationship "in" with Aragon). I argue that there are different approaches to this limitation. One possibility would be to carefully design rules to automatically create new relationships, another possibility would be to keep the nodes unconnected but have an interface showing all unconnected nodes and letting users reconnect mentioned nodes. Redundancies can be hard to distinguish (e.g. two different users create the nodes "Seematte" and "Hundämättäli" with no knowledge of the other. Even though these locations are redundant, the system has near to no possibility to detect said nodes as redundant) and are a growing problem, unnecessarily using storage space.

Due to the dynamic nature of online content, a gazetteer has to allow frequent updates on different levels of granularity. If a node is split into two separate nodes, questions of assigning the underlying nodes to the two new nodes arise (e.g. when The Republic of South Sudan declared independence from Sudan, the node Sudan would have to be split into two separate "Country" nodes. The question then is, how a system can detect how nodes with former relationships to the former node Sudan can correctly be linked to the two new nodes; If the municipality of Biel would

[10] www.google.com

[11] www.local.ch

fuse with the municipality of Nidau, how should the different relationships be managed and attributed?)

Varying geometries can be a major limitation for the proposed model. Entries with long geometries will distort the discussed concept of near (e.g. due to the size and extend of the country Russia, Finland will be returned as a country near North Korea, because Finland is adjacent to Russia which is adjacent to North Korea. Only two adjacency relationships have to be traversed.) I argue that this can be adjusted using the spatial footprints and additionally calculating the Euclidian distances, but with a major loss in efficiency. Another geometry related problem are complex polygons with islands and holes. Should an island polygon be given the relationship "in" to the same or higher hierarchical level (e.g. is the island polygon of the Canton of Bern in which Clavaleyres lies "in" the country of Switzerland or "in" the canton of Fribourg or in the canton of Vaud)?

The last limitation lies in the expertise of users. Seeing that Neo4J is a relatively new technology, there are only a limited number of experts, able to affectively design and maintain large systems. SQL on the other hand is widely known and documented, which begs the question if the proposed model can be created and maintained over a long time period.

4. DISCUSSION

The presented conceptual model has the potential to minimise the limitations of a gazetteer presented by [18]. The first limitation regarding "(1) the limited spatial representation (a point or a rectangle) and absence of support for spatial relationships" [18] is solved by the presented model by allowing multiple geometries for a single entry. Complex spatial representations are possible by allowing simple (e.g. points, bounding boxes) and complex (lines, polygons) geometries to be stored. In addition, Neo4Js spatial plugin allows for complex spatial queries and further increases the potential for complex gazetteer queries. Not only spatial representations but also spatial and semantic relationships can be stored and queried. This leads to a solid foundation for efficiently performing complex spatial relationship queries.

The second limitation, namely "(2) the absence of support for semantically complete, but geographically imprecise locations, such as "south of France" or "upstate New York"" [18] is addressed by introducing vernacular names as proposed by the OMT-G schema [24] in combination with spatial footprint information and semantic relationships. With the presented conceptual model, regions belonging to the "south or France" can be linked to the vernacular name node "South of France" or every entry which is in the south of France can be given the tag "south France". Another approach is to utilise the available spatial footprint information and do spatial calculations based on the user need. More interestingly, vague spatial terms such as "near" can be defined as nodes connected by no more than two "adjacency" relationship types on a given hierarchical level (e.g. Biel is near Ipsach, because only two "adjacency" relationships need to be traversed).

The third limitation, "(3) the lack of intra-urban detail, including places often mentioned in natural language text and possibly know by non-residents, such as monuments or tourist attractions" [18] is addressed by allowing varying complexities of geometries, tags and vernacular names to be stored. In addition, the proposed conceptual model uses the predefined feature types presented by GeoNames[12], which includes various structure types such as monuments, castles and museums allowing for such intra-urban details not only to be stored, but also to be set in relationship with other entries (e.g. the monument ADJECENT to the museum IN the park in Zürich).

Although many limitations can be circumnavigated or solved by using the proposed conceptual model, new limitations arise and need to be addressed. Especially the mentioned limitations of creating or deleting nodes within an already linked portion of the graph and geometry related limitations need further attention for the proposed conceptual model to provide a solid foundation for a next generation graph based gazetteer. Further limitations will arise with increasing complexity and connectedness of the graph.

5. CONCLUSION

I believe the proposed conceptual model has great potential to facilitate fast and computationally efficient querying on enormous datasets of locations and the relationships between locations. The conceptual model has the power to redefine the function and structure of gazetteers by transitioning gazetteers from being a mere collection of triples of toponyms, spatial footprints and feature types to a vast network of connected nodes, not only storing location based information but also the relationships between different locations. Furthermore, I believe the presented conceptual model has the ability to store not only locations and their relationships, but could also be adapted to store various different types of information including meteorological or demographic data, making it not only interesting to geographic information retrieval but also to other disciplines dealing with large spatial or non-spatial linked data collections.

6. FURTHER RESEARCH

In this paper I have presented a conceptual model for a next generation gazetteer, able to deal with complex queries. I have discussed the problems the proposed model could potentially solve and I have presented various new limitations. These new limitations are not only limitations of the presented conceptual model, but pose fundamental challenges in the domain of geographic information retrieval. There is a high demand for further research to be done regarding geographic information retrieval and gazetteer modelling. I especially call for research to be done using a graph based gazetteer structure with a high number of nodes and relationships to be able to make empirically tested statements about efficiency, possible query complexity and useability. I would also like to highlight the limited amount of available research analysing and discussing graph based gazetteers and would appreciate more research be conducted, seeing the increasing importance of this domain. Lastly I would like to point out that it would be beneficial for further research to be conducted in an interdisciplinary manner, due to the different aspects of geographic information retrieval and gazetteers (e.g. language and semantics, computation, system architecture, interface design).

7. ACKNOWLEDGMENTS

My thanks to ACM SIGCHI for allowing me to modify templates they had developed.

[12] www.geonames.org

8. REFERENCES

[1] D. Sullivan, "Google Still World's Most Popular Search Engine By Far, But Share Of Unique Searchers Dips Slightly," 2013. [Online]. Available: http://searchengineland.com/google-worlds-most-popular-search-engine-148089. [Accessed: 04-Jan-2016].

[2] C. B. Jones and R. S. Purves, "Geographical information retrieval," *Int. J. Geogr. Inf. Sci.*, vol. 22, no. 3, pp. 219–228, 2008.

[3] C. Keßler, K. Janowicz, and M. Bishr, "An agenda for the next generation gazetteer: Geographic information contribution and retrieval," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in Geographic Information Systems*, 2009, pp. 91–100.

[4] B. Sundheim and R. Irie, "Resources for Place Name Analysis," *Analysis*, vol. 92152, pp. 317–320, 2004.

[5] M. Sanderson and J. Kohler, "Analyzing geographic queries," *J. Am. Soc. Inf. Sci.*, pp. 8–10, 2004.

[6] A. Henrich and V. Luedecke, "Characteristics of geographic information needs," in *Proceedings of the 4th ACM workshop on Geographical information retrieval - GIR '07*, 2007, p. 1.

[7] J. Joishi and A. Sureka, "Vishleshan: Performance Comparison and Programming Process Mining Algorithms in Graph-Oriented and Relational Database Query Languages," *Int. Database Eng. Appl. Symp. (IDEAS 2015)*, pp. 0–5, 2015.

[8] V. Kolomičenko, M. Svoboda, and I. H. Mlýnková, "Experimental Comparison of Graph Databases," *Proc. Int. Conf. Inf. Integr. Web-based Appl. Serv. - IIWAS '13*, pp. 115–124, 2013.

[9] C. Vicknair, M. Macias, Z. Zhao, X. Nan, Y. Chen, and D. Wilkins, "A comparison of a graph database and a relational database," *Proc. 48th Annu. Southeast Reg. Conf. ACM SE 10*, p. 1, 2010.

[10] S. Eliot and J. Rose, *A Companion to the History of the Book*. John Wiley & Sons, 2009.

[11] M. Sanderson and W. B. Croft, "The History of Information Retrieval Research," *Proc. IEEE*, vol. 100, pp. 1444–1451, 2012.

[12] D. Meyer, "Microsoft down to fifth place in comScore's global search stats, thanks to Yandex | Gigaom," 2013. [Online]. Available: http://gigaom.com/2013/02/06/microsoft-down-to-fifth-place-in-comscores-global-search-stats-thanks-to-yandex/. [Accessed: 04-Jan-2016].

[13] C. B. Jones, R. Purves, A. Ruas, M. Sanderson, M. Sester, M. Van Kreveld, and R. Weibel, "Spatial information retrieval and geographical ontologies an overview of the SPIRIT project," in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 2002, pp. 387–388.

[14] R. R. Larson, "Geographic Information Retrieval and Digital Libraries," *Lect. Notes Comput. Sci.*, no. Dl, pp. 461–464, 2009.

[15] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems," *Sci. Stat. Database Manag. 2007. SSBDM '07. 19th Int. Conf.*, no. Ssdbm, p. 16, 2007.

[16] F. Gantner, "A Spatiotemporal Ontology for the Administrative Units of Switzerland," p. 139, 2011.

[17] R. Laurini, "Geographic Ontologies, Gazetteers and Multilingualism," *Futur. Internet*, vol. 7, pp. 1–23, 2015.

[18] I. M. Machado, R. O. de Alencar, R. de O. C. Junior, and C. A. Davis Jr, "An ontological gazetter for geographic information retrieval," *GeoInfo*, no. Hill 2000, pp. 21–32, 2010.

[19] C. Jones, H. Alani, and D. Tudhope, "Geographical information retrieval with ontologies of place," *Spat. Inf. theory*, 2001.

[20] G. Kirby, A. Dearle, L. Williamson, C. De Kerckhove, J. Carson, and C. Dibben, "Comparing Relational and Graph Databases for Pedigree Data Sets," 2014.

[21] G. Fu, C. B. Jones, and A. I. Abdelmoty, "Ontology Based Spatial Query Expansion in Information Retrieval," *Lect. Notes Comput. Sci. - ODBASE2005*, vol. 3761, pp. 11466–1482, 2005.

[22] L. a. Souza, C. a. Davis Jr., K. a. V. Borges, T. M. Delboni, and A. H. F. Laender, "The Role of Gazetteers in Geographic Knowledge Discovery on the Web," *LA-WEB 2005 Third Lat. Am. Web Congr.*, pp. 157–165, 2005.

[23] K. A. V. V. Borges, A. H. F. F. Laender, C. B. Medeiros, and C. a. Davis Jr., "Discovering geographic locations in web pages using urban addresses," *GIR '07 Proc. 4th ACM Work. Geogr. Inf. Retr.*, pp. 31–36, 2007.

[24] K. A. V. Borges, C. A. Davis, and A. H. F. Laender, "OMT-G: An Object-Oriented Data Model for Geographic Applications," *Geoinformatica*, vol. 5, no. 3, pp. 221–260, 2001.

[25] GeoNames, "Feature Codes GeoNames." [Online]. Available: http://www.geonames.org/export/codes.html. [Accessed: 04-Jan-2016].

[26] GeoNames, "About GeoNames." [Online]. Available: http://www.geonames.org/about.html. [Accessed: 04-Jan-2016].

[27] F. a. Twaroch and C. B. Jones, "A web platform for the evaluation of vernacular place names in automatically constructed gazetteers," *Proc. 6th Work. Geogr. Inf. Retr. - GIR '10*, p. 1, 2010.

[28] L. Hollenstein and R. Purves, "Exploring place through user-generated content: Using Flickr to describe city

cores," *J. Spat. Inf. Sci.*, vol. 1, no. 1, pp. 21–48, 2010.

[29] Neo4J-Contrib, "Neo4J Spatial Plugin." [Online]. Available: https://github.com/neo4j-contrib/spatial#index-and-querying. [Accessed: 07-Jan-2016].

[30] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.

[31] A. G. SUTCLIFFE, M. ENNIS, and J. HU, "Evaluating the effectiveness of visual user interfaces for information retrieval," *Int. J. Hum. Comput. Stud.*, vol. 53, no. 5, pp. 741–763, 2000.

[32] Neo4J, "Graph Database vs Relational Databases." [Online]. Available: http://neo4j.com/developer/graph-db-vs-rdbms/. [Accessed: 04-Jan-2016].