Problem Description

Working title:
1: Top-k relevant spatio-temporal retrieval on knowledge graphs using keyword search
2: Spatio-temporal keyword search on RDF graphs.

Knowledge graphs allows for easy extraction of facts from an entity, and the structure makes it efficient to find multi degree relations between different entities and facts. The task is to study indexing and search techniques on graphs containing spatio-temporal data, and propose effective ways to retrieve the data contained in the graph based on keyword searches.

Summary

Write your summary here...

Preface

Spatio-temporal keyword search on RDF graphs.

Table of Contents

List of Tables

List of Figures

Abbreviations

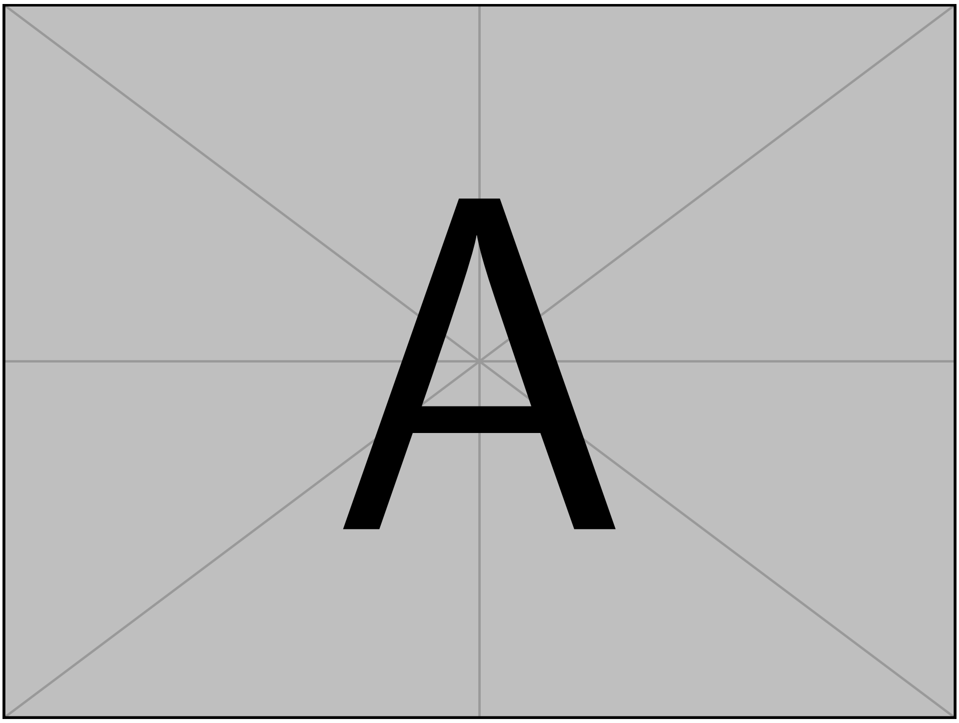
Symbol	=	definition

Chapter 1

Introduction

Knowledge bases are complex collections of information. This information can be both structured or unstructured. The information in such a collection can be used to create a graph with information in small chunks representing a fact, as well as the relation between such facts. There exists multiple such graphs today most are based on Wikipedia, such as Yago[5] and DBpedia[1].

Knowledge graphs is structured with a subject, predicate, and object. Both the subject and object are facts, and the predicate describes the relation between two facts. This structure makes the graph a directed cyclic graph. An example of the subject predicate object can be ”Nidaros Cathedral” ”is located in” ”Trondheim”.



One of the uses of knowledge graphs today is to find and display a info box in search engines. This information is a compact set of facts that tries to fit the search query. Because of the graph structure of knowledge graphs the information in the info box can be adapted to the query by choosing the predicates and related facts closest related to the query. This makes it possible to create a set of information that can give the user a quick overview of the information retrieved by the query.

- Why this is important
- What is newish in this paper

1.1 Previous work

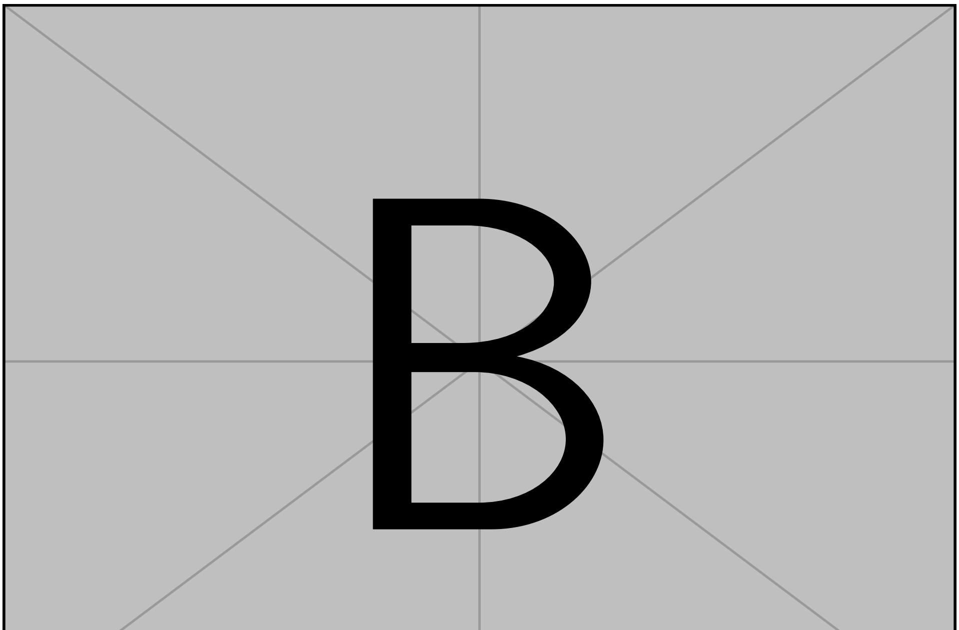
There has been done some work keyword search on RDF graphs. This includes [7] and [2]. None of these takes spatial or temporal data into consideration. The methods for keyword queries outlined in the papers are based on a combination of indexing the graph and traversing subgraphs. The indexing is similar in both papers, but retrieval, scoring and traversal methods varies. One method of retrieval and scoring that is used as a baseline is a breadth first search for traversal and a minimal subgraph for scoring. These methods are easy to implement and understand, but often inefficient, and inaccurate.
In paper [4] some methods for indexing, searching and ranking keyword searches on spatial RDF graphs is proposed. These methods builds upon the methods outlined in [7, 2].

The most substantial difference is the use of R-trees to index the spatial dimension of the graph. This is done so that a subgraph containing the keywords can be retrieved given a location. This method starts by finding the closest spatial vertex and uses that as the root for the subgraph. Subgraphs rooted in a spatial node requires different ranking, and there is proposed a set of rules in the paper.
R-trees are designed around the spatial dimension, but some work on modifications to include a temporal dimension has been done. Some research done on spatio-temporal tree structures include [6, 8]. Most of the spatio-temporal trees are however created to be able to query and index moving or frequently updated objects. To include a temporal dimension some differences in the tree structures are made. Most approaches builds on the R-tree, but modifies the tree in different aspects. For the purpose of this thesis a different tree structure is not needed.

Chapter 2

Data structures

2.1 Knowledge graphs



2.1.1 Yago

Yago[5] is a knowledge graph created from a set of different sources. The graph is built from Wikipedia, WordNet, GeoNames and Wikidata. This combination of sources makes it possible to get information in multiple languages, and spatio-temporal data.

Chapter 3

Basic retrieval methods

3.1 Indexing

3.1.1 Spatial indexing

Most spatial indexing is done using R-trees. R-trees are based on a B-tree, but is optimized for indexing spatial dimensions. R-trees are able to effectively index spatial data by dividing space into smaller and smaller sets, or bounding boxes, and having the leaf nodes of the trees representing the smallest unit of space used in the data set, usually a small area or a point.

3.1.2 Temporal indexing

In the dataset used in this thesis objects do not move, or move rarely. This makes indexing time unnecessary and it can be treated as a fact in the object with a start, and possible end time. This allows time to be retrieved as a fact when traversing the graph the same way as other facts.

3.2 Search

3.2.1 Bottom up BFS search

The most basic method of finding a match for keywords is using a bottom up breadth first search (BFS) [3]. For each keyword in the query the algorithm will find all vertices that contain the keyword. From that set of vertices the BFS search finds the first vertex that can connect all the vertices in the set. The vertex that connects the the rest of the set is the one that best fits the keywords in the search. There is however no guarantee that this set of vertices contains any spatial or temporal data. If this is the case, the search will continue until the a vertex containing spatial and temporal data is found. The first set of vertices containing all the keywords, a place, and time will be the best result using this search. The

BFS search can however contain multiple times, or multiple places. To determine what time or what place best fits the query, a separate ranking algorithm can be used. BFS is also a time consuming algorithm and is used as a proof of concept and baseline in this thesis.

Algorithm 1 Bottom up BFS

```
procedure BFS Insert algo here
```

Chapter 4

Optimized methods

4.0.1 Pruning

When using the BFS search method, all possible spatial vertices close to the queried place will be explored. This is expensive and many of the vertices will be irrelevant. Pruning the potential place vertices will reduce the amount of subgraphs traversed, and will in turn reduce the overall time used to find results for the query.

Chapter 5

Methodology

5.1 Datasets

5.1.1 YagoFacts

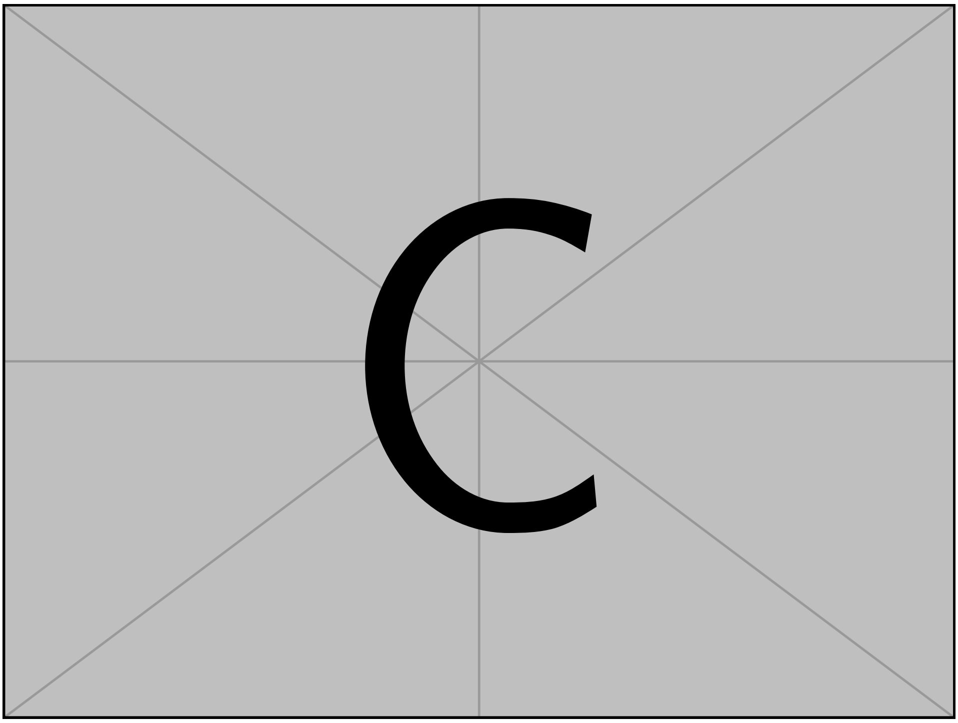
Data set containing all facts in Yago that hold between instances. This data set is the largest that is used, and holds all information.

5.1.2 GeoNames

All facts in Yago containing geographical coordinates and names. This data set contains all the place information and is used to index places and is used for the spatial indexing and used when finding a place that fits with the query or used as a starting place for spatial queries.

5.1.3 DateFacts

All facts in Yago containing dates.



5.2 Evaluation

There are multiple possible methods for evaluating the indexing and search methods. A good method for evaluating the retrieval methods is time. The faster information can be retrieved the better the retrieval method should be, assuming the information retrieved is correct. For the purpose of this theses time complexity will be the main evaluation method, along with evaluation of how well the retrieved information fits the query.

5.2.1 Time complexity

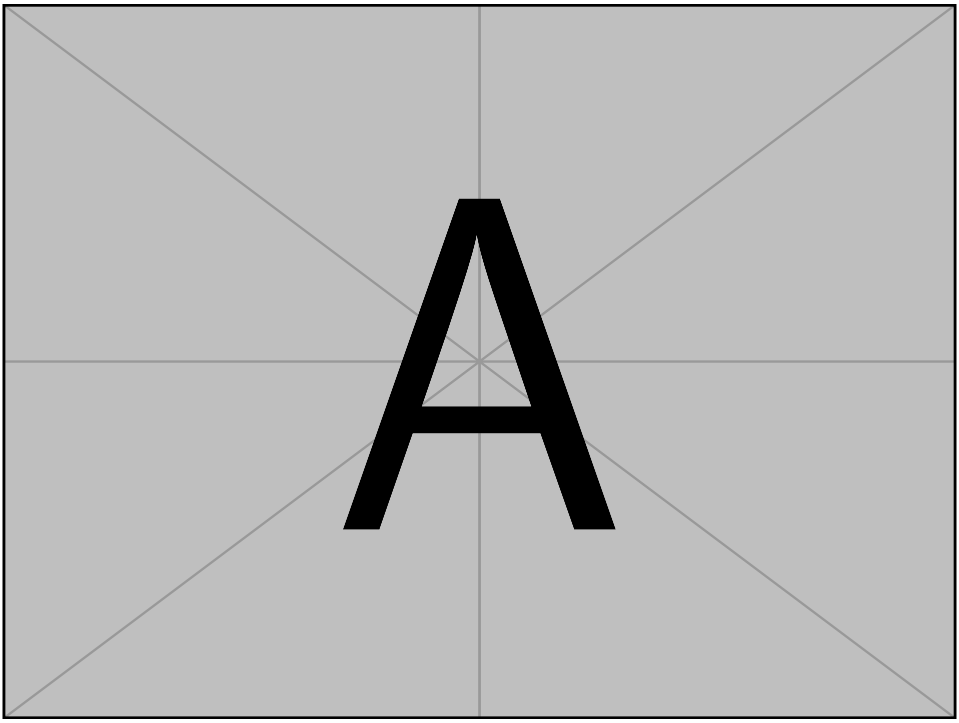
5.2.2 Space complexity

5.2.3 Information match

Chapter 6

Results

6.1 Indexing



6.2 Search

Chapter 7

Discussion

Chapter 8

Conclusion

Bibliography

[1]

[2] S. Elbassuoni and R. Blanco. Keyword search over rdf graphs. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 237–242, New York, NY, USA, 2011. ACM.

[3] H. He, H. Wang, J. Yang, and P. S. Yu. Blinks: Ranked keyword searches on graphs. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, pages 305–316, New York, NY, USA, 2007. ACM.

[4] J. Shi, D. Wu, and N. Mamoulis. Top-k relevant semantic place retrieval on spatial rdf data. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1977–1990, New York, NY, USA, 2016. ACM.

[5] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A Core of Semantic Knowledge. In *16th International Conference on the World Wide Web*, pages 697–706, 2007.

[6] Y. Tao, D. Papadias, and J. Sun. The tpr*-tree: An optimized spatio-temporal access method for predictive queries. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 790–801. VLDB Endowment, 2003.

[7] T. Tran, H. Wang, S. Rudolph, and P. Cimiano. Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *2009 IEEE 25th International Conference on Data Engineering*, pages 405–416, March 2009.

[8] S. Šaltenis, C. Jensen, C. (codirector, M. Böhlen, C. Dyreson, H. Gregersen, D. Simonas, J. Skyt, G. Slivinskas, K. Torp, R. (codirector, B. Moon, M. Soo, A. Com, and A. Timeconsult. R-tree based indexing of general spatio-temporal data. 01 2000.

Appendix

Write your appendix here...