Geo-tagging Non-Spatial Concepts

Amgad Madkour
Purdue University
West Lafayette, USA
amgad@cs.purdue.edu

Walid G. Aref
Purdue University
West Lafayette, USA
aref@cs.purdue.edu

Mohamed Mokbel
University of Minnesota - Twin Cities
Minneapolis, USA
mokbel@cs.umn.edu

Saleh Basalamah
Umm Al-Qura University
Makkah, KSA
smbasalamah@uqu.edu.sa

ABSTRACT

Concept Geo-tagging is the process of assigning a textual identifier that describes a real-world entity to a physical geographic location. A concept can either be a spatial concept where it possesses a spatial presence or be a non-spatial concept where it has no explicit spatial presence. Geo-tagging locations with non-spatial concepts that have no direct relation is a very useful and important operation but is also very challenging. The reason is that, being a non-spatial concept, e.g., crime, makes it hard to geo-tag it. This paper proposes using the semantic information associated with concepts and locations such as the type as a mean for identifying these relations. The co-occurrence of spatial and non-spatial concepts within the same textual resources, e.g., in the web, can be an indicator of a relationship between these spatial and non-spatial concepts. Techniques are presented for learning and modeling relations among spatial and non-spatial concepts from web textual resources. Co-occurring concepts are extracted and modeled as a graph of relations. This graph is used to infer the location types related to a concept. A location type can be a hospital, restaurant, an educational facility and so forth. Due to the immense number of relations that are generated from the extraction process, a semantically-guided query processing algorithm is introduced to prune the graph to the most relevant set of related concepts. For each concept, a set of most relevant types are matched against the location types. Experiments evaluate the proposed algorithm based on its filtering efficiency and the relevance of the discovered relationships. Performance results illustrate how semantically-guided query processing can outperform the baseline in terms of efficiency and relevancy. The proposed approach achieves an average precision of 74% across three different datasets.

1. INTRODUCTION

Around 80% of all data contains some reference to spatial locations [5]. The Web of Data [7] is mostly comprised of a set of single concepts or real-world *things* termed *concepts*. Some of the concepts in the Web of Data have an associated spatial dimension or location, *e.g.*, the White House or the San Diego Zoo. We refer to these concepts as *spatial concepts*. We assume that each location that is itself a concept and we refer to it as a spatial concept. In contrast, other concepts do not have an associated spatial dimension or location, *e.g.*, pollution, crime, traffic, and health. We refer to these concepts as *non-spatial concepts*. Non-spatial concepts can have an *implicit* relation with other spatial concepts. For example, "crime", a non-spatial concept, can be related to spatial concepts that have the following types, *e.g.*, bus-stops and avenues. "Crime" and bus stops do not conceptually belong to the same type. The question that this paper addresses is the following: *Given a non-spatial query concept, say X, how can we identify spatial concept types that are related to X?* For example, consider the following query: "Find Pollution in NYC". In the query, "Pollution" is the non-spatial query concept. The answer to the query is a list of spatial results that have the following types, *e.g.*, bus stops, railroads, and garages. Given a location of interest such as "NYC" of the query, if we can find the locations of the bus stops, the railroads, and the garages, we can now geotag the non-spatial concept "pollution" on the matching location types in NYC.

The co-occurrence of these concepts within textual resources provides evidence for identifying the implicit relations between spatial and non-spatial concepts. Most keyword-based search engines retrieve relevant results based on the query keywords occurring in the textual resources. For example, answering the query: "Find Education in Seattle" can obtain spatial results, *e.g.*, schools and Universities that can only have the keyword "Education" (or its derivative forms) appearing explicitly in the corresponding textual resources.

In this paper, we propose to answer this type of queries by identifying the relation between the non-spatial query concept and the spatial concepts types. We refer to this process as *type relatedness*. For example, consider the query: "Find Pollution in Indiana". The spatial concept "Wabash Valley Power Authority" that has Type "Power Plant" is related to the non-spatial query concept "Pollution", and

hence needs to considered by the query. In order to be able to answer these types of queries, this paper addresses the following two challenges. The first challenge is related to representing the co-occurrences of spatial and non-spatial concepts within the same textual resources. We propose to create an undirected weighted graph that contains an edge between concepts that occur in the same textual resource. The second challenge is related to the traversal of the graph to infer the types of spatial concepts that are semantically related to the non-spatial concept in the query. We propose a series of Linked-Data filters for pruning the results and presenting the user with the most relevant *types* that relate spatial concepts to the non-spatial query concept.

This paper introduces a system for Geo-tagging concepts. Geo-tagging being the process of assigning a textual identifier, namely a concept, to a location.The proposed system, termed Concept Geotagger, dubbed "*CGTag*", operates in two phases: (*i*) an offline phase, and (*ii*) an online phase. In the offline phase, CGTag extracts the co-occurring concepts in every textual resource. Then, CGTag creates a clique graph among these co-occurring concepts for every textual resource. We refer to these clique graphs as the *local graphs*. An edge between two concepts indicates the existence of a co-occurrence relation and is assigned a weight of how frequent the relation has appeared across documents. The weight does not include the number of occurrences of the relation in the same document. For example, if a relations appeared 4 times within the first document and 3 times in another document then the final weight will be 2, disregarding the number of times it appeared in each document. Then, these relations are stored in a database, termed the *knowledge store*. This knowledge store contains the aggregation of the smaller local graphs. We refer to the graph in the knowledge store as the *global graph*. If a relation between two concepts is identified in a local graph, where the same relation already exists in the global graph, then the weight (*i.e.*, frequency of occurrence) of the relation in the global graph is increased by one.

Finally, the constructed relations are stored in the global graph of the knowledge store. In the online phase, CGTag offers a web interface that captures the user query. The query parameters are the non-spatial concept of interest, *e.g.*, pollution, crime, etc., and a location of interest, *e.g.*, Los Angeles. The location of interest helps restrict the results to a specific region. These two parameters are passed to a query processing algorithm that learns the types of spatial concepts that are most related to the query. This learning process takes place on the global graph in the knowledge store. First, the query processor filters out the spatial concepts that do not belong in the location of interest from the global graph. Next, the query processor filters the remaining concepts based on a proposed set of semantic predicates (*i.e.*, relations). Finally, the query processor identifies the *types* of the remaining set. These types are used to geotag the non-spatial query concept with spatial concepts in the location specified by the query.

1.1 Contribution

The contributions of this paper are as follows.

- We propose CGTag, a system for geotagging a non-spatial concept query with spatial concepts based on *type relatedness*.
- We propose a semantic query-processing algorithm that uses several Linked-Data filters.
- We propose an evaluation method for type relatedness in addition to a baseline to determine the correctness of the results.

The rest of paper proceeds as follows. Section 2 presents the related work. Section 3 illustrates how CGTag represents the relation between co-occurring concepts. Section 4 presents the architecture of CGTag and discusses its main components. Sections 5 and 6 present the experimental setup and experimentalresults, respectively. Finally, Section 7 contains concluding remarks.

2. RELATED WORK

There has been a variety of studies performed for constructing large knowledge bases. Some of these knowledge bases are constructed in an automated fashion. These knowledge bases utilize unsupervised techniques that process web resources. For example, Linkedgeodata [16] converts data from OpenStreetMap to an RDF model. Linkedgeodata derives a lightweight ontology from the OpenStreetMap data. Linkedgeodata also provides an *interlinking* dataset that links its concepts with DBpedia, GeoNames, and other datasets. Linkedgeodata also provides simple spatial semantic predicates (*i.e.*, relations) based on proximity and the containment of points. DeepDive [14] employs statistical learning and inference to construct a knowledge base. GeoDeepDive [18] employs unsupervised techniques over geographically specific textual resources to observe aspects of rock formation where a rock formation is based on two or more minerals.

In terms of semantic search and querying, various studies discuss how to capture the query semantics or intent. Egenhofer [3] advocates the concept of *Semantic Geospatial Web*, where he states the need to explicitly represent the query intent through different predicates. This implies a more precise retrieval based on the semantics of the data rather than the query's explicitly stated keywords only. Lim *et al.* [10] discuss how relational databases queries are not usually precise where users often have a vague understanding of what they are querying. They propose a Query-By-Example approach that allows capturing the query semantics in a relational database setting. Calderon-Benavides *et al.* [2] suggest the use of facets or dimensions to capture the query intent when searching for information over the Web. The selection of the facets is performed by observing a set of queries. Among these dimensions is the spatial sensitivity that indicates the interest of the user in spatial locations. Fernandez *et al.* [4] propose a semantic search model that integrates semantic knowledge within traditional information retrieval ranking models. They propose a ranking function that combines the semantic similarity with keyword-based similarity. Lim *et al.* [11] study the issue of expressing queries against an ontology in SQL. Their approach relies on asking the user for a small number of examples that satisfy the query so that the system infers the exact query intent automatically. The approach consists of three steps, namely providing examples that satisfy the query, using machine learning techniques to mine the query semantics, and apply-

ing the query semantics over the data in order to generate the query result.

3. REPRESENTING CO-OCCURRENCE

The hypothesis adopted in CGTag is that *all concepts mentioned in the same textual resource are implicitly related to each other* [6,13]. A clique can be used to represent the relations between the co-occurring concepts. A clique contains edges between all pairs of vertices. A clique is used to represent the concepts co-occurrences where the vertices represent the concepts and the edges represent the co-occurrence relation with an assigned weight. The same weight is assigned to all the edges between the unique nodes in the initial relation representation. The assumption is that concepts co-occurring in the same textual resource have equal importance. The intuition behind creating a clique is to indicate a single co-occurrence relation between the concepts and each other. As illustrated in the following sections, weight filtering (*i.e.*, threshold) is used to discard co-occurring relations with a low weight.

The co-occurring concepts are referred to as *candidate concepts*. The *types* of a candidate concept can be used by CGTag to infer the concept type relatedness to a non-spatial query concept. For example, if the non-spatial query concept is "Pollution", some of the candidate concepts can have types such as "Factory" or "Bus Station".

4. ARCHITECTURE

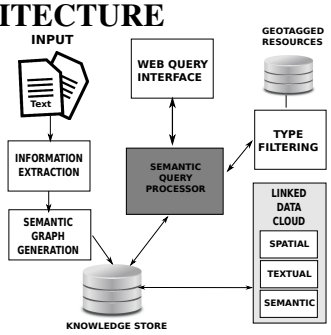

Figure 1: System architecture

Figure 1 illustrates the proposed architecture. The input is a set of textual resources. For example, a textual resource can be a Wikipedia article where the title represents the concept name. First, the textual resources are passed through an information extraction component that identifies the other concepts in the textual resources. Second, the identified concepts are passed to a clique construction where a local graph is generated. Third, the clique is "mashed-up" with the global graph of the knowledge store. Fourth, the web interface is used to issue the queries (*i.e.*, non-spatial concept) over a user-specified location (if needed). The location is used to narrow the results to spatial concepts in a specific region in space. If the location is omitted, CGTag attempts to match against all the spatial concepts. Finally, the query processing algorithm returns the type-related spatial concepts to the user.

4.1 Information Extraction

The offline phase starts by extracting the co-occurring concepts. The information extraction phase (*IE*, for short) identifies the candidate concepts in textual resources. IE performs the following three main tasks: (*i*) identification, (*ii*) disambiguation, and (*iii*) linked-data concept (URI) assignment to the identified concept. The assigned (URI) can be used to look-up additional information for the concept in the Linked Data cloud. In this study, a statistical information extraction technique is adopted that achieves good accuracy when compared to rule based approached.

4.2 Graph Construction

The second step of the offline phase is to construct a local graph clique representation and append the clique relations to the global graph. The following algorithm illustrates the steps involved in the graph construction.

Algorithm 1 Graph Construction

Input: *cconcepts*,*graph*
  for $i = 0; i < cconcepts.length;\ i++$ **do**
    for $j = i+1; j < cconcepts.length;\ j++$ **do**
      if exists(*graph*, *cconepts*[*i*], *cconcepts*[*j*]) **then**
        $weight \leftarrow getweights(graph, cconcepts[i], cconcepts[j])$
        $weight \leftarrow weight + 1$
        updateweights(*graph*,*cconcepts*[*i*],*cconcepts*[*j*],*weight*)
      else
        $weight \leftarrow 1$
        addnodes(*graph*,*cconcepts*[*i*],*cconcepts*[*j*],*weight*)

Algorithm 1 illustrates how the local graph is constructed and appended to a global graph. The algorithm requires two inputs. The first input is the set of candidate concepts of the clique. The second input is the global graph. If a relation already exists between any pair of the candidate concepts in the global graph, the weight of the relation in the global graph is incremented by one. Otherwise, a new edge representing the co-occurrence between the two concepts is introduced into the global graph with a weight of one.

4.3 Knowledge Store

In the offline phase, the knowledge store component is used to store the concepts and relations. In the online phase, the knowledge store is used by the query processing component to answer the user queries. In addition, the knowledge store aggregates the spatial predicates (*i.e.*, relations) of the concepts such as latitude and longitude.

4.4 Semantic Query Processing

The clique construction creates a substantial number of relations that need to be filtered. A set of semantic filters are proposed to filter the irrelevant concepts from the global graph. Given a non-spatial query concept, the semantic query processor infers the types of spatial concepts in the global graph that are most related to the non-spatial concept query.

Algorithm 2 illustrates the filtering steps and their order of execution. The input to the semantic query processor is (*i*) the non-spatial concept query, and (*ii*) a location of interest The function *getcandidates* queries the concepts (*i.e.* candidate concepts) that are related to the query concept. The relation is defined by the existence of an edge between the query concept and other queries. Given that there could

Algorithm 2 Semantic Query Processing

Input: *concept*
Input: *location*
Output: *matches*
matches ← []
cthreshold ← *val*1
sthreshold ← *val*2
matches ← *getcandidates*(*concept*)
matches ← *filterbycooccurance*(*matches*, *cthreshold*)
matches ← *expandbytype*(*matches*)
matches ← *filterbytype*(*matches*)
matches ← *filterspatially*(*matches*, *location*)
matches ← *filterbysimilarity*(*matches*, *sthreshold*)

be an immense amount of candidate concepts, the semantic query processor applies a set of Linked Data filters to prune the candidate concepts set. Finally, the remaining set represents the spatial concepts that have a *type* that is most related to the non-spatial query concept.

The filtering steps in Algorithm 2 are grouped into three main filters. The first filtering step filters concepts based on the co-occurrences frequency (*i.e.*, weight). The second filtering step filters concepts based on some Linked-Data properties. Finally, the third filtering step filters the concepts based on the similarity between their textual resources.

4.4.1 The Co-occurrence Threshold Filter

The weight of the relation between two concepts represents how frequent the two concepts co-occur together. This weight reflects a measure of how much two concepts are related. In other words, the higher the weight, the more relevant the concepts are to each other. A cut-off (*i.e.*, threshold) is defined to filter out the relations with low weights. The threshold value is defined based on the required number of returned results. This threshold is known as the *co-occurrence threshold*. For example, if we are interested in k-results, we adaptively set a threshold value that would return the k-highest concepts.

In this paper, we use the co-occurrence threshold filter as a baseline for measuring concept-to-type accuracy. The reason is that the weight represents the frequency of co-occurrence between two concepts across all the analyzed textual resources. For example, CGTag can consider only relations that have a weight of at least three. In other words, when a non-spatial concept query is issued, the corresponding related concepts are fetched from the knowledge store that have a weight of at least three (*i.e.*, the threshold is set to be three or lower).

4.4.2 Linked-Data Filter

The Linked-Data cloud includes a rich set of semantic predicates (*i.e.*, relations) for concepts. It also includes a rich set of ontologies that describe the concepts and predicates. The Linked-Data cloud can be used as a mean for understanding the *spatial semantics* of concepts. Some knowledge bases provide *type* classes and ontologies that aid in understanding the hierarchical nature of concepts, *e.g.*, see [1,8]. Others provide useful predicates, *e.g.*, spatial information [16], abstracts of documents, and references. A subset of predicates and ontology properties are used to learn what constitutes a relevant relation between concepts. The Linked-Data filter includes three main steps: (*i*) expand by type, (*ii*) type filtering, and (*iii*) spatial filtering.

Many collections in the Linked-Data cloud provide a "type" predicate [1] [1, 8]. For example, DBPedia provides a type predicate to describe the concepts it contains. In DBPedia, a concept. e.g., "The White House" has the types ArchitecturalStructure, Place, and Building. This rich information can aid the semantic query processor when filtering based on the type. Some relations for a non-spatial query concept can themselves be types. For example, there can be a relation between a non-spatial concept "The White House" and another concept named "Building". where building is also itself a type. This is still a very useful relation as it allows us to know that a "Building" itself is an important type to consider in the result set.

Some ontologies include a predicate that describes the *type* of the concepts in a specific collection. In most collections, the types can be described in a hierarchical manner indicating super-type and sub-type relationships. For example, the DBPedia ontology has the type "Building", which is a superclass of (Hotel, Restaurant, ShoppingMall, Castle, HistoricBuilding) among others.

Algorithm 3 Filter by Type

```
Input: oldmatches
Output: newmatches
  spatiallist ← [Place, Organisation, SpatialThing, Area]
  newmatches ← newlist()
  for concept in oldmatches do
      concepttypes ← gettypes(concept)
      if len(concepttypes) ← 0 then
          concept2type ← convert2type(concept)
          if concept2type in spatiallist then
              newmatches.append(concept)
          else
              lst ← getsuperclasses(concept2type)
              for itm in lst do
                  if itm in spatiallist then
                      newmatches.append(concept)
                      break
      else
          for typ in concepttypes do
              if typ in spatiallist then
                  newmatches.append(concept)
                  break
              else
                  lst ← getsuperclasses(typ)
                  for itm in lst do
                      if itm in spatiallist then
                          newmatches.append(concept)
                          break
```

Algorithm 3 illustrates how a concept is filtered based on its ontology type. The *spatial types*, e.g., "Place", "Organisation", "SpatialThing", and "Area" can be manually identi-

[1] http://www.freebase.com

fied in the ontology of interest. Each concept can have more than one type. In the type filtering step, the intuition is to filter all the concepts that are not a spatial type. The first step is to *filter by type* and determine if the evaluated concept has a type that is among the specified spatial types list. The second step is to *expand by type* in order to determine if the type of a concept is a subclass of a spatial type. Finally, the matching concepts are returned for further filtering.

4.4.3 Similarity Filtering

We assume that there exists a textual resource that describes what a concept is. This applies for both spatial and non-spatial concepts. Given that textual resources tend to share common keywords, we suggest using these textual resources to measure a level of similarity between concepts. This becomes very useful when the number of filtered concepts is large. For example, a non-spatial concept query, e.g., "Bioinformatics", has a spatial concept result of Type "Education" including Schools and Universities. The results can be narrowed down by performing a pairwise document similarity between the textual resources of the concepts. For example, the "Bioinformatics" concept is compared against the other textual resources matching the spatial concepts of type "University" and "School", and the results may contain only matches of type "University".

The similarity between a pair of concepts is calculated by generating a TF-IDF representation [15] for the non-spatial query concept and every filtered spatial concept list. TF-IDF stands for Term frequency/Inverse Document frequency and is used to indicate how important a term is with respect to a document in a collection or corpus. The first step is to generate the TF-IDF vectors for the concepts' textual resources. The second step is to compute the cosine similarity between the two resulting vectors.

$$cos\Theta = \frac{t1.t2}{\mid t1 \mid\mid t2 \mid} \quad (1)$$

Equation (1) indicates the cosine similarity calculation method. The parameters $t1$ and $t2$ represent the textual resource of the non-spatial concept and one of its candidates, respectively. The resulting score of the comparison is checked against a predefined threshold value to determine if the two textual resources of the concepts are similar. Section 5 illustrates some empirical results that help determine a reasonable threshold value to use.

4.5 Type Filtering of Non-Spatial Concepts

The final step in the online phase is to filter the spatial concepts based on the types proposed by the query processing component. The type filtering component is responsible for determining the Geo-tagged resources (*i.e.*, spatial concepts) that have a type matching the types deduced by the semantic query processor. If a location is specified in the query, then the location acts as a filtering criteria for the spatial concepts to compare against. For example, if the query processor suggests the type "Art", then the spatial linking module attempts to match the type "Art" against the types of geo-tagged resources. If location is specified such as "NYC" then the linking is restricted to "NYC" only.

Interlinking allows linking non-spatial concepts to collections that include spatial information about the concepts. Interlinking supplies a URI for concepts. This allows fetching futher semantic information for the concepts from the Linked Data cloud. Many linked-data resources do not have any interlinking information. There are systems, *e.g.*, SILK [9], that support creating the interlinking automatically. Vilches-blazquez *et al.* [17] propose using co-reference resolution for interlinking geospatial Linked Data. Interlinking datasets are utilized in this work to discover concepts that have a spatial dimension (*i.e.*, latitude, longitude).

5. EXPERIMENTAL SETUP

5.1 Evaluation

CGTag is evaluated based on two overlapping factors: (*i*) query processing filtering efficiency, and (*ii*) the accuracy of the type relatedness.

The filtering efficiency of each strategy presented in Sections 4.4, 4.4.1 and 4.4.2 is evaluated separately and then in combination with each other. In specific, the number of remaining concepts are observed after each strategy has been applied. The Linked Data predicates are also evaluated separately (*i.e.*, type filtering, spatial filtering, similarity filtering).

In order to evaluate the accuracy of type relatedness, we presented 8 evaluators with 30 arbitrarily selected non-spatial concept queries. Given a non-spatial concept, the objective is to understand what would be the*expected types* of spatial concepts in the result. For example, given a non-spatial concept query, e.g., "Science" from Table 1 and a location, e.g., "California" (*i.e.*, query is "Science in California"), the expected results can include spatial concepts belonging to the type School and University. In specific, one would expect a spatial concept result such as "Stanford Univeristy" and "UC Berkley".

In this study, the Linkedgeodata interlinks dataset illustrated in Table 2 is used. The table indicates the number of spatial concepts that were assigned a specific type. The Linkedgeodata collection provides 7 types that match the types of the selected concepts (*i.e.*, spatial and non-spatial concepts). These types are "City","Island","Mountain","School","Stadiu and "University". This set of types is used as the evaluation set. Majority vote among the evaluations is used to determine the correct types for each non-spatial concept. The evaluators are instructed to indicate at least the two most common types that they are likely to find for the spatial concepts results.

Table 1 lists the 30 non-spatial queries that are selected for type relatedness evaluation and the corresponding majority vote judgments produced by the evaluators. The value of 1 indicates the evaluators' agreement on a certain type for a specific non-spatial concept. Precision is used as an evaluation measure in order to quantitatively measure the accuracy of the results over the queries. It is important to note that the evaluation is purely subjective, not relying on any ground truth except for the evaluators' knowledge.

5.2 Queries

Two sets of non-spatial concept queries are defined for the experiments. The first set is used for testing the filtering efficiency of the query processing algorithm. A total of 90 non-spatial concepts of Type "Activity" are arbitrarily selected from the DBPedia collection. Concepts of type "Activity" include a wide range of sub-classes that span multiple types. The queries are processed by the semantic query processing algorithm introduced in Section 4.4 and the output is averaged to produce the results reported in Section 6. For the second set of queries, a total of 30 non-spatial concepts are arbitrarily selected as the queries to evaluate the type relatedness. The selected 30 concepts in the rows of 1 are the top-ranking concepts in terms of number of relations they have with other concepts. This ensures a fair evaluation for queries of non-spatial concepts that have enough coverage from multiple documents. In other words, a higher weight indicates a higher co-occurrence between these concepts across multiple textual resources.

It is important to distinguish between types and non-spatial concepts. While types and non-spatial concepts may actually mean the same, it is important to note that types are a special kind of non-spatial concepts that can be selected by ontologies such as LinkedGeoData to represent a set of concepts. In this paper, we adopt the same distinction between non-spatial concepts and types.

5.3 Collections and Datasets

A set of 178K articles from Wikipedia are used as the primary source for identifying the co-occurrences among concepts. Wikipedia is a very relevant resource for the task as it contains full articles about concepts. The DBPedia collection is the RDF representation of Wikipedia. DBPedia provides an additional rich medium for interlinking the concepts mentioned in Wikipedia with other collections. Linkedgeodata [16] [2] is used for identifying the spatial information (*i.e.*, latitude and longitude). Interlinks between the spatial locations, *e.g.*, Linkedgeodata, and concepts, *e.g.*, DBPedia are realized using the interlinks datasets provided by Linkedgeodata. The dataset provides links between a DBPedia entry (representing a Wikipedia concept) and a Linkedgeodata entry (representing an Openstreetmap entry).

Query	Airport	City	Island	Mountain	School	Stadium	University
Science	0	0	0	0	1	0	1
Medicine	0	0	0	0	1	0	1
Business	1	1	0	0	1	0	0
Fishing	0	1	1	0	0	0	0
Canal	0	1	1	1	0	0	0
Dormitory	0	0	0	0	1	0	1
English_studies	0	0	0	0	1	0	1
Agriculture	0	1	1	0	1	0	1
Training	0	0	0	0	1	1	1
Research	0	0	0	0	1	0	1
Population	0	1	1	0	0	0	0
River	0	1	1	1	0	0	0
Train_station	1	1	0	0	0	0	0
Village	0	1	1	1	0	0	0
College	0	1	0	0	1	0	1
High_school	0	1	0	0	1	0	1
Student	0	0	0	0	1	1	1
Lake	0	1	1	1	0	0	0
Suburb	0	1	1	0	0	0	0
Museum	0	1	0	0	0	0	1
Baseball	0	0	0	0	1	1	1
Education	0	1	0	0	1	0	1
Unincorporated_area	0	1	1	1	0	0	0
Road	0	1	0	1	0	0	0
Neighborhood	0	1	0	0	1	0	1
History	0	1	0	0	1	0	1
Bridge	0	1	1	1	0	0	0
Broadcasting	1	1	0	0	0	1	1
Law	1	1	0	0	1	0	1
Association_football	0	0	0	0	1	1	1

Table 1: Test set used for type relatedness evaluation

Criteria	USA	DEU	UK
Airport	3128	27	109
City	8469	7409	4521
Island	92	0	45
Mountain	887	76	587
School	2026	7	154
Stadium	55	6	8
University	70	4	25
Total	14727	7529	5449

Table 2: Interlinks Dataset Statistics

Table 2 illustrates the various types that the interlinks dataset provides. Three spatial concepts datasets are used for: (*i*) United States (USA), (*ii*) United Kingdom (GBR), and (*iii*) Germany (DEU).

5.4 Concept Extraction

Concept extraction is performed using *DBPediaSpotlight* [12] [3], an unsupervised learning tool for identifying DBPedia topics in textual resources. As indicated earlier, every concept has an associated textual resource. For example, a concept such as the White House has a Wikipedia page that represents the textual resource of that concept. The White House concept can have other candidate concepts mentioned in its textual resource. We processes 178K concepts where we extract the distinct candidate concepts from every textual resource of a concept. The tool includes a support parameter that indicates the prominence of a concept in Wikipedia. The support parameter indicates how important a concept is based on the concepts that link to it. We fine-tune the support parameter of the tool to highlight concepts with high prominence only. We set the support parameter to be larger than 100, indicating our interest in concepts that have more than 100 pages linking to this concept.

5.5 Baseline

The Co-occurrence threshold (THR) is used as the baseline. Initially, the threshold is set to 3 (*i.e.*, 3 minimum relations between two concepts) in order for a concept to qualify as candidate concept. The threshold is changed adaptively if less than 10 results per non-spatial query is obtained.

6. RESULTS AND DISCUSSION

6.1 Type Relatedness Evaluation

In this section, type relatedness of the query results is discussed given different filtering strategies. In specific, the

[2] http://www.linkedgeodata.org
[3] http://spotlight.dbpedia.org

type relatedness accuracy is measured when using: (*i*) the Linked-Data strategy without similarity filtering (R1), (*ii*) the Linked-Data strategy with similarity filtering (R2), (*iii*) the co-occurrence threshold strategy (R3), (*iv*) Linked Data + co-occurrence strategies without the Linked Data similarity filtering (R4), and (*v*) Linked Data + co-occurrence strategies with similarity filtering (R5). Precision is used as an accuracy measure for this evaluation. The reason is because precision conveys the accuracy of the presented types. Recall on the other hand is not used as we are not currently focusing on how many types can be but rather how correct one or a few can be. Correctness is defined as the number of types that are most relevant to the query concept and are not below a certain user-defined threshold.

Technique	USA	GBR	DEU
LD-wo-Similarity	0.43	0.43	0.42
LD-w-Similarity	0.69	0.7	0.78
THR(3)	0.78	0.68	0.06
LD-wo-Similarity+THR(3)	0.536	0.53	0.52
LD-w-Similarity+THR(3)	0.72	0.73	0.76

Table 3: Precision of results

Table 3 gives the precision values for each of the five combinations (*i.e.*, R1,R2,R3,R4,R5) over the three datasets. The R5 strategy that considers the similarity predicate achieves nearly consistent results across the three datasets. When compared to the baseline, R5 achieves better results except in the case of the USA dataset. This is due to the tight constraints at the semantic predicate level. It can be observed that some relations are missed because the Linked-Data type filtering generates diverse types. The missed relations is due to the nature of concepts included in the USA dataset. The missed relations would also be less apparent if the relations are created over a domain specific dataset. For the THR(3) entry for Germany dataset, the result is very low compared to the other two datasets due to the large limited concepts relations in that dataset.

6.2 Clique Size

We study the effect of clique sizes on the number of relations. The clique size is the number of concepts considered when generating the clique. We measure the average number of relations available for the 90 arbitrarily chosen non-spatial concept queries. The clique sizes considered are 10, 50 and 100 concepts (i.e. nodes in a graph where a node represents a concept).

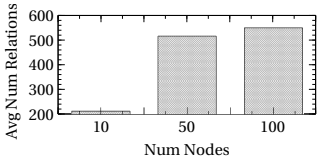

Figure 2: Average number of relations available over 90 non-spatial queries

Figure 2 illustrates how the number of relations are affected by the size of the clique graph. A larger clique generates a higher number of relations. This will in turn affect the number of generated results. Observe that the significance of the clique size does not matter as the number of concepts considered increases. This shows that considering more concepts does not necessarily add significantly more relations.

The number of concepts to consider for the clique graph depends on a number of aspects. The first aspect is the domain that a concept covers. Some textual datasets tend to have concepts mentioned sparsely across documents. This calls for considering more concepts per document (i.e. bigger clique size). Another alternative is to choose a smaller set of concepts per document (i.e. smaller clique size) and use ranking techniques to select the most representative concepts.

The second aspect for selecting the number of nodes (i.e. concepts) in the clique graph is the type and size of the documents to be analyzed. Analyzing documents with abstracts will be different from analyzing blog entries or news articles. In the former scenario, most important concepts tends to reside in the abstract section while in the latter they tend to be scattered across the document.

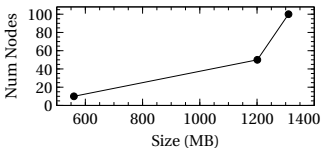

Figure 3: Database size based on number of nodes

The third aspect is the overall disk space occupied by the database. As the number of concepts increases, so does the storage requirement. This in turn leads to slower query response time. Figure 3 illustrates how the number of nodes affects the graph storage size.

6.3 Co-occurrence Threshold Selection

The following results evaluate the co-occurrence threshold (THR) approach. The effect of selecting various thresholds is illustrated with respect to the number of identified relations among candidate concepts. A graph size of 100 nodes is used in all of the remaining experiments.

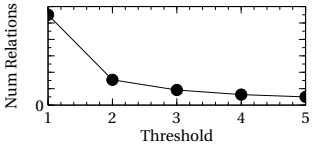

Figure 4: Number of relations discovered at different threshold values

A total of 90 queries are used in order to determine the number of relations among candidate concepts at each of the threshold values. Figure 4 illustrates how the number of relations significantly differs when the threshold is slightly varied.

6.4 Linked-Data Filtering

Next, the results of Linked-Data filtering is discussed. The first set of experiments evaluates the effectiveness of each strategy separately.

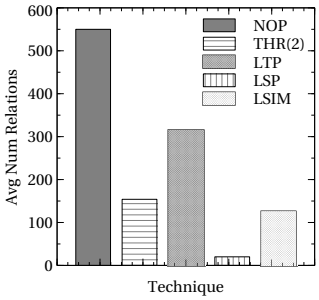

Figure 5: Individual Strategies - NOP: No filtering, THR(2): Threshold filtering of 2, LTP: Type filtering, LSP: Spatial filtering, LSIM: Similarity filtering

Figure 5 illustrates the effect of running each strategy separately on the number of candidate concepts relations. Co-occurrence threshold filtering (THR) is applied with a value of 2. Linked-Data type filtering (LTP) includes the type filtering results in addition to the expanded results of the Linked Data Expansion (LTE) step (*i.e.*, LTP = LTP + LTE). Threshold filtering performs better as it relies only on filtering entries that are seen in its graph. LTP filters out almost half the entries on average over the 90 queries. Linked-Data spatial filtering (LSP) exhibits the highest filtering effectiveness as it returns only the entries in a specific location. Linked-Data similarity (LSIM) filtering also exhibits high filtering effectiveness on the textual resource level. The effectiveness of LSIM is almost the same as that of THR when using a value of two. This would not be the case if the (THR) value used is different. For example, the no-filtering strategy may also be considered a THR approach with a value of one. Observe that changing the threshold slightly can achieve significantly different results.

The second set of experiments evaluates the combined effectiveness of the strategies. The four combinations that are evaluated include: (*i*) Linked-Data Expansion+Linked Data Type Filtering+ Linked Data Similarity filtering (*i.e.*, C1), (*ii*) Linked data Expansion+ Linked Data Type filtering + Linked Data Similarity filtering + Linked Data Spatial filtering (*i.e.*, C2), (*iii*) Linked Data filtering + Co-occurrence filtering (*i.e.*, C3), and (*iv*) Linked Data filtering + Co-occurrence filtering without spatial filtering (*i.e.*, C4). Figure 6 illustrates the filtering effect of each of these four combinations.

The first combination (*i.e.*, C1) is the least effective of the other four combinations (*i.e.*, C1, C2, C3, and C4). C1 does not consider the Linked-Data spatial filtering predicate (*i.e.*, not filtering based on a location). A significant decrease in the number of candidate relations occurs when using the spatial predicate in combination C2. This means that most of the candidate concepts relations corresponding to the non-spatial query are not specific to the query location. The effect of spatial filtering can also be observed in the results of the fourth combination C4. On the hand, the third combination C3, representing the Linked-Data and co-occurrence strategies with all its predicates, offers the best filtering efficiency. The co-occurrence threshold approach provides a good initial filtering criterion that the Linked-Data filter can

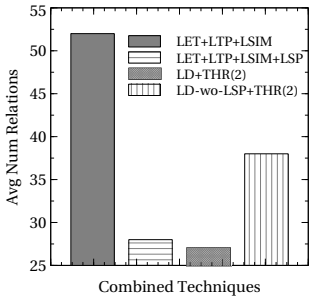

Figure 6: Combining techniques and measuring the average number of candidate concepts relations

build upon. The co-occurrence threshold filter provides frequent relations while the Linked-Data filter provides useful semantic processing.

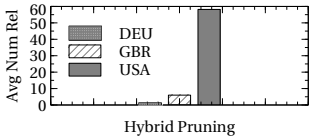

Figure 7: Linked Data+co-occurrence (*i.e.*, hybrid) filtering over 3 datasets

In Figure 7, the effect of filtering on the DEU dataset and the USA dataset can be observed. The USA dataset contains the highest number of candidate concepts relations. This illustrates the effect of running the experiments on non-homogeneous datasets where USA, DEU, and GRB have an unequally distributed set of concepts that possess diverse types.

6.5 Similarity Threshold Selection

In this experiment, different thresholds are illustrated to determine the similarity between a non-spatial query concept and its related spatial concepts.

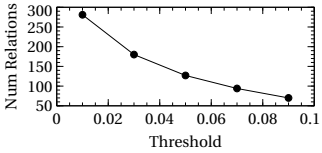

Figure 8: Number of relations left at different similarity thresholds

Figure 8 illustrates how the number of candidate relations decreases as the threshold increases. Clearly, it is important to use a low threshold as this may lead to missing some important candidate concept relations.

Determining the similarity threshold is also dependent on the type and length of the textual resources being used. As the textual resources length increases, so does the sparsity. In turn, this implies the need for a low-valued threshold.

7. CONCLUDING REMARKS

This paper presents CGTag, a system for discovering type relatedness between spatial and non-spatial concepts. CGTag presents a methodology for capturing the co-occurrence of concepts from textual resources. It demonstrates how these co-occurrences can be used as a means for discovering implicit spatial relationships between non-spatial and spatial concepts. CGTag uses local and global graphs for representing the co-occurrence relations. The global graph keeps growing as more concepts and relations are identified. CGTag has a query-processing algorithm that identifies the spatial types related to a query-specified non-spatial concept. Experimental results illustrate that concept co-occurrences within the same textual resource is a viable mean for capturing the implicit relation between non-spatial and spatial concepts.

8. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, and J. Lehmann. Dbpedia: A nucleus for a web of open data. *ISWC/ASWC*, pages 722–735, 2007.

[2] L. Calderon-Benavides, C. Gonzalez-Caro, and R. Baeza-Yates. Towards a Deeper Understanding of the User's Query Intent. *SIGIR*, 2010.

[3] M. Egenhofer. Toward the semantic geospatial web. *SIGSPATIAL*, pages 1–4, 2002.

[4] M. Fernández, I. Cantador, V. López, D. Vallet, P. Castells, and E. Motta. Semantically enhanced Information Retrieval: An ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434–452, Dec. 2011.

[5] C. Franklin. An introduction to geographic information systems: Linking maps to databases. *Database*, 15(2):12–21, Apr. 1992.

[6] Z. Harris. Distributional structure. *Springer*, 1981.

[7] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Synthesis Lectures on the Semantic Web. 2011.

[8] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, pages 28–61, 2013.

[9] A. Jentzsch, R. Isele, and C. Bizer. Silk–generating RDF links while publishing or consuming linked data. *ISWC*, pages 1–4, 2010.

[10] L. Lim, H. Wang, and M. Wang. Semantic queries in databases: problems and challenges. *CIKM*, 2009.

[11] L. Lim, H. Wang, and M. Wang. Semantic Queries by Example Categories and Subject Descriptors. *EDBT*, pages 347–358, 2013.

[12] P. N. Mendes, M. Jakob, A. Garcia-silva, and C. Bizer. DBpedia Spotlight : Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics).*, volume 95, pages 1–8, 2011.

[13] S. Mohammad and G. Hirst. Distributional measures of semantic distance: A survey. *CoRR*, 2012.

[14] F. Niu, C. Zhang, C. Ré, and J. Shavlik. DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. *VLDS 2012*, 2012.

[15] G. Salton and M. J. McGill. Introduction to modern information retrieval. *McGraw-Hill*, 1986.

[16] C. Stadler, J. Lehmann, K. Höffner, and S. Auer. LinkedGeoData: A core for a web of spatial open data. *Semantic Web*, 3:333–354, 2012.

[17] L. Vilches-Blázquez, V. Saquicela, and O. Corcho. Interlinking geospatial information in the web of data. In *Bridging the Geographic Information Sciences*, Lecture Notes in Geoinformation and Cartography, pages 119–139. Springer Berlin Heidelberg, 2012.

[18] C. Zhang and C. Ré. GeoDeepDive : Statistical Inference using Familiar Data-Processing Languages. *SIGMOD*, pages 1–4, 2013.