Keyword-Based Querying of Geo-Textual Data

Christian S. Jensen

www.cs.aau.dk/~csj

Center for Data-intensive Systems

The Web Is (Mostly) Mobile

- A quickly evolving mobile Internet infrastructure.
  - Mobile devices, e.g., smartphones, tablets, laptops, navigation devices
  - Communication networks and users with access
- Rapidly increasing device sales (millions)
  - Smartphones: 2010: 310: 2011: 490; 2012: 680; 2013: 969; 2014: 1,245; 2015: 1,424
  - PCs (desktop, laptop): 2010, 2011, 2012: 350; 2013: 316; 2014: 318; 2015: 321; 2016: 333
  - Tablets: 2011: 66; 2012: 116; 2013: 195; 2014: 216; 2015: 233; 2016: 259
- Mobile is a mega trend.
  - Google went "mobile
  - Mobile data traffic 2020 = mobile data traffic 2010 x 1000

Mobile Is Spatial

- Increasingly sophisticated technologies enable the accurate geo-positioning of mobile users.
  - GPS-based technologies
  - Positioning based on Wi-Fi and other communication networks
  - New technologies are underway (e.g., GNSSs and indoor).

Outline

- Background and motivation
- Top-*k* spatial keyword queries
- Continuous top-*k* queries
- Accounting for co-location
- Aggregate queries, including collective and group queries
- Summary and challenges

(Acknowledgments and references are given at the end.)

Spatial Web Querying

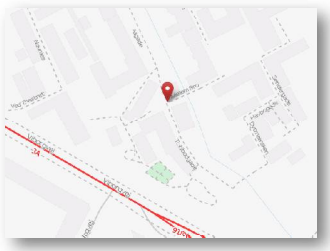
- Total web queries
  - Google: 2011 daily average: 4.7 billion (uncertain)
- Queries with local intent
  -
  - Google: ~20% of desktop queries
  - Bing: 50+% of mobile queries
- Vision: Improve web querying by exploiting accurate user and content geo-location
  - Smartphone users issue keyword-based queries
  - The queries concern websites for places
- Balance spatial proximity and textual relevance
- Support different use cases

Spatial Web Objects

- Objects: $p = \langle \lambda, \psi \rangle$ (location, text description)
- Example:

λ = *(56.158889, 10.191667)*

ϕ = Den Gamle By Open-Air Museum Den Gamle By - "The Old Town" – was founded in 1909 as the world's first open-air museum of urban history and culture…



Spatial Web Objects Sources

- Web pages with location
- Online business directories
  - Business name, location, categories, reviews, etc.
  - Example: Google Places
- Geocoded micro-blog posts
  - Example: Twitter
  - Messages with up to 140 characters.

***Top-k* spatial keyword querying**

Top-*k* Spatial Keyword Query

- Objects: $p = \langle \lambda, \psi \rangle$ (location, text description)
- Query: $q = \langle \lambda, \psi, k \rangle$ (location, keywords, # of objects)

- Ranking function

$$rank_q(p) = \alpha \frac{\| q.\lambda, p.\lambda \|}{\max D} + (1-\alpha)(1 - \frac{tr_{q.\psi}(p.\psi)}{\max P}) \qquad 0 \le \alpha \le 1$$

  - Distance: $\| q.\lambda, p.\lambda \|$
  - Text relevancy: $tr_{q.\psi}(p.\psi)$
    - Probability of generating the keywords in the query from the language models of the documents

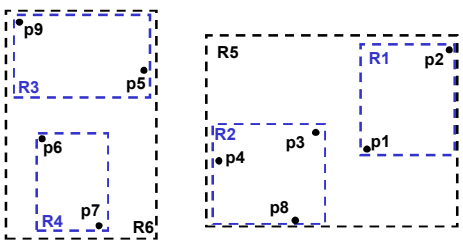
- Generalizes the *k*NN query and text retrieval

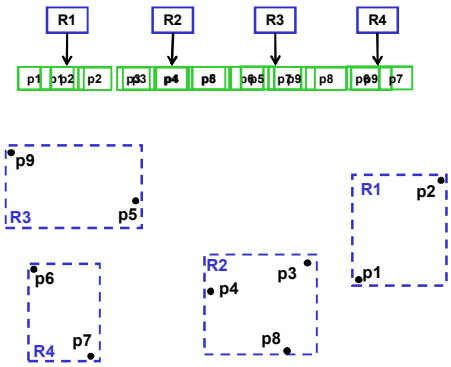
Spatial Keyword Query Processing

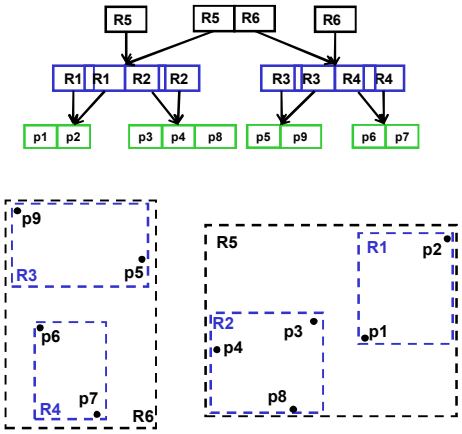
- How do we process spatial keyword queries efficiently?

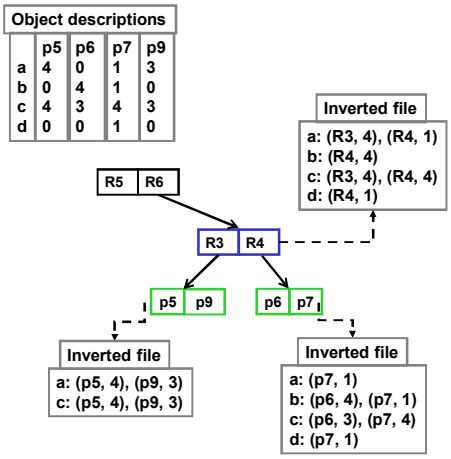
- Proposal
  - Prune both spatially and textually in an integrated fashion
  - Apply indexing to accomplish this

- The IR-tree [Cong et al. 2009 ; Li et al. 2011; Wu et al. 2012]
  - Combines the R-tree with inverted files
  - R-tree: good for spatial
  - Inverted files: good for text







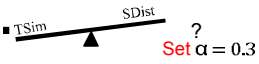


Why Not Top-k Spatial Keyword Query I

$$f(q,p) = \alpha\,(1 - SDist(q,p)) + (1-\alpha)TSim(q,p)$$

- ❑ Top-2 "clean/comfortable" hotels near COEX ($\alpha = 0.5$):
  - Rank 1: Intercontinental
    Rank 2: Oakwood
  - Rank 3: Park Hyatt (not returned)
- ❑ Refined query:
  - Use larger k? Set k to 3 or larger

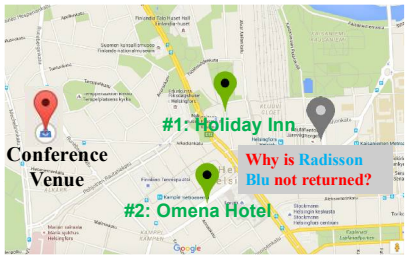
	1-SDist()	TSim()
Intercontinental	0.8	0.8
Oakwood	**0.7**	0.6
Park Hyatt	0.3	**0.9**



? Set $\alpha = 0.3$

- Modify both k & α?

Why Not Top-k Spatial Keyword Query II



- ❑ Top-2 "Clean, Comfortable" hotels near Conference Venue:
  - Rank 1: Holiday Inn
    Rank 2: Omena Hotel
  - Rank 3: Raddison Blu (not returned)
- ❑ Refined query:
  - Use a larger k?
    Set k to 3 or larger
  - User other query keywords?
    Query with "Clean, Comfortable, Luxury"
  - Modify both k & q. ?

Continuous top-*k* querying

Continuous Spatial Keyword Queries

- Objects: $p = \langle \lambda, \psi \rangle$ (location and text description)
- Query: $q = \langle \lambda, \psi, k \rangle$ (location, keywords, # of objects)
- A continuous query where argument changes continuously
- Ranking function

$$rank_q(p) = \frac{\| q.\lambda, p.\lambda \|}{tr_{q.\psi}(p.\psi)}$$

Euclidean distance (changes continuously)

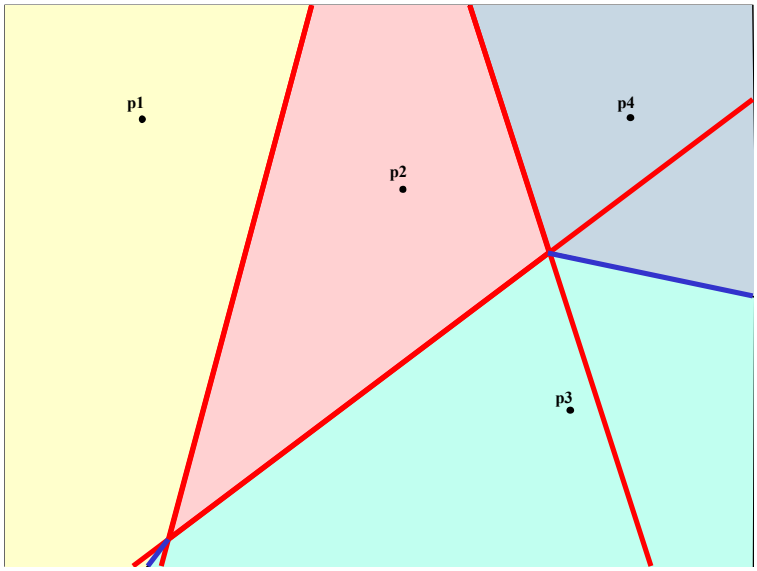
Text relevancy (query dependent)

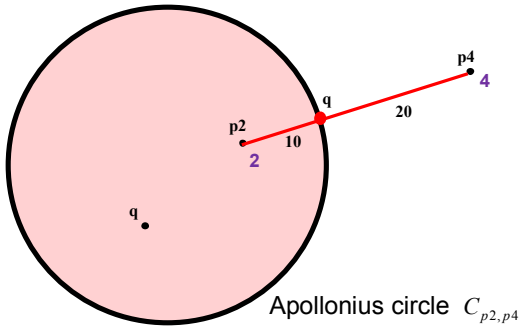
Continuous Spatial Keyword Queries

- How can we process such queries efficiently?
  - Server-side computation cost
  - Client-server communication cost
- While the argument changes continuously, the result changes only discretely.
  - Do computation only when the result may have changed
- Use safe zones
  - When the user remains within the zone, the result does not change.
  - The user requests a new result when about to exit the safe zone.

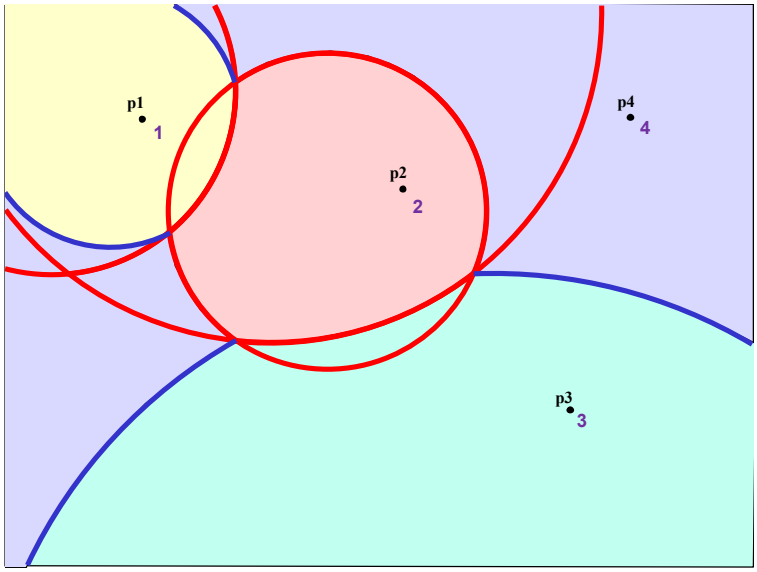
Processing Continuous Queries

- Compute results
  -
- Compute corresponding safe zones
  - Integrate with result computation
- Prune objects that do not contribute to the safe zone without inspecting them
  - Use the IR-tree
  - Access objects in border-distance order
  - Prune sub-trees
  - Terminate safely when a stopping criterion is met





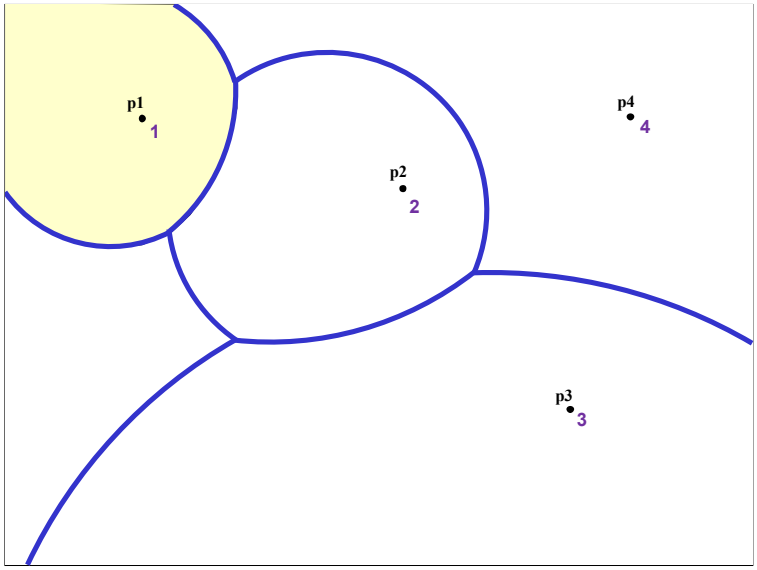
Apollonius circle $C_{p2,p4}$



Representation of a Multiplicatively Weighted Voronoi Cell

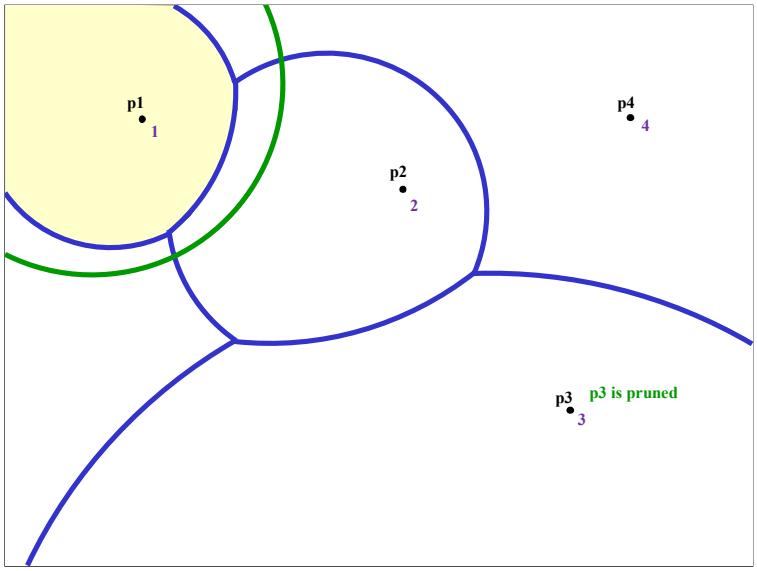
Influence Objects

$I^+ \cup I^o \cup I^-$



Pruning Objects p_+ with Higher Weights

$$\exists p' \in I^+(C_{p^*,p_+} \supseteq C_{p^*,p'})$$



Pruning Objects with Equal Weights

$$\exists p' \in I^+(\perp_{p^*,p_o} \supseteq C_{p^*,p'})$$
$$\exists p' \in I^o(\perp_{p^*,p_o} \supseteq \perp_{p^*,p'})$$

Pruning Objects with Lower Weights

$$\exists p' \in I^+(C_{p_-,p^*} \cap C_{p^*,p'} = \varnothing)$$
$$\exists p' \in I^-(C_{p_-,p^*} \subseteq C_{p',p^*})$$
$$\exists p' \in I^o(C_{p_-,p^*} \cap \perp_{p^*,p'} = \varnothing)$$

Prestige-based ranking

Accounting for Co-Location

- So far, we have considered data objects as independent, but they are not.
- It is common that similar places co-locate.
  - Markets with many similar stands
  - Shopping centers, districts
  -
  - Restaurant and bar districts
  - Car dealerships
- How can we capture and take into account the apparent benefits of co-location?

Top-*k* Spatial Keyword Query

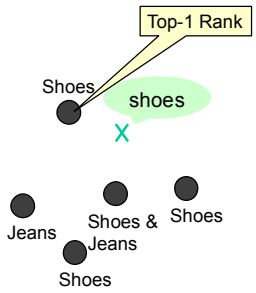
- Objects: $p = \langle \lambda, \psi \rangle$ (location, text description)
- Query: $q = \langle \lambda, \psi, k \rangle$ (location, keywords, # of objects)

- Ranking function

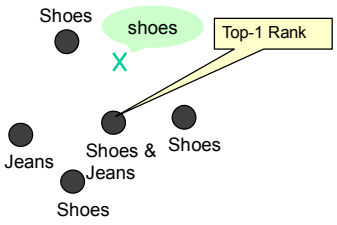
$$prrank_q(p) = \alpha \frac{\| q.\lambda, p.\lambda \|}{\max D} + (1-\alpha)(1 - pr_{q.\psi}(p.\psi)) \qquad 0 \le \alpha \le 1$$

  - Distance: $\| q.\lambda, p.\lambda \|$
  - Text relevancy: $pr_{q.\psi}(p.\psi)$
    - PR score: prestige-based text relevancy (normalized)

Standard Retrieval Approach



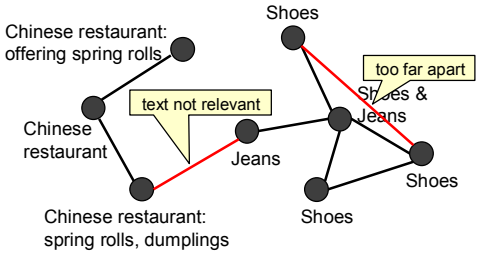
Prestige-Based Retrieval



Prestige-Based Ranking

- Prestige propagation using a graph **G** = (V, E, W)
  - Vertices V: spatial web objects
  - Edges E: connect objects that meet constraints
  - Distance threshold: $\| p_i.\lambda, p_j.\lambda \| \le \lambda$
  - Similarity threshold: $sim(p_i.\psi, p_j.\psi) \le \xi$ (vector space model)
  - Edge weights W: $\| p_i.\lambda, p_j.\lambda \|$
- Use Personalized PageRank for ranking [Jeh & Widom, 2003]

Prestige-Based Ranking



Experimental Study

- Local experts are asked to provide query keywords for locations and then to evaluate the results of the resulting queries.
- The studies suggest that the approach is able to produce better results than is the baseline without score propagation.

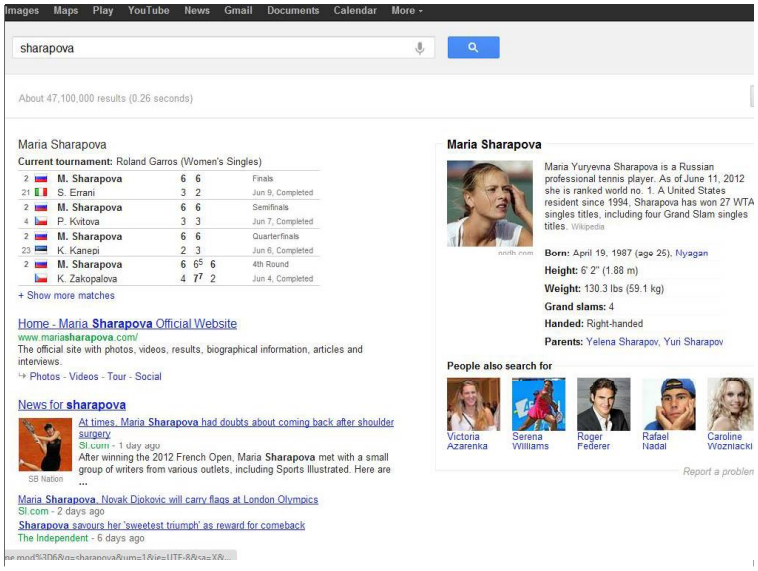
Digression: Methodology

- The same underlying methodology underlies the studies covered.
- Define precisely a problem of perceived real-world interest.
- Develop solutions
  - Concepts, data structures, algorithms
- Carry out mathematical analyses
  - Correctness, complexity, storage size
- Prototype the solutions and perform emprical studies
  - Often, real data is needed
  - Offers detailed insight in the design properties of the solutions
- Iterate!

Aggregate queries

Aggregate Spatial Keyword Querying

- So far, the granularity of a result has been a single object
- We may want to return *sets* of objects that collectively satisfy a query.
- Collective queries
  - Find a set of objects that collectively satisfy the query
  - Aggregate the result documents into a single document
  - Apply spatial proximity conditions to the result objects internally and with respect to the query
- Top-*k* groups queries
  - Find groups of objects that satisfy the query
  - Each object in a group is relevant to the keywords
  - Apply spatial proximity conditions to the result objects internally and with respect to the query

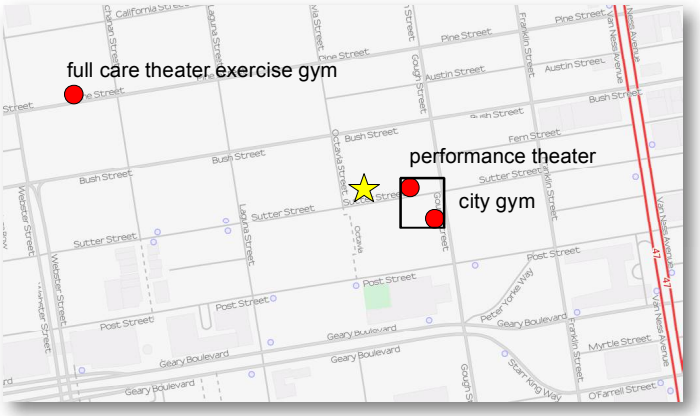


Collective Spatial Keyword Querying

- The spatial aspect offers natural ways of aggregating data objects and providing aggregate query results.
- We may want to return *sets* of objects that collectively satisfy a query.

The Collective Spatial Keyword Query

- Query location: ★
- Query keywords: theater, gym



The Collective Spatial Keyword Query

- Objects: $o = \langle \lambda, \psi \rangle$ (location and text description)
- Query: $Q = \langle \lambda, \psi \rangle$ (location and keywords)

- The result is a group of objects satisfying two conditions.
  - $Q.\psi \subseteq \bigcup_{o \in \chi} o.\psi$
  - $Cost(Q, \chi)$ is minimized.

- $Cost(Q,\chi) = \alpha C_1(Q,\chi) + (1-\alpha) C_2(\chi)$
  - $C_1(.,.)$ depends on the distances of the objects in χ to Q.
  - $C_2(.)$ characterizes the inter-object distances among objects in χ.
  - α balances the weights of the two components.

Collective Query Variants

- Type 1: cost function:

$$Cost(Q,\chi) = \sum_{o \in \chi} Dist(o,Q)$$

- Application scenario
  - The user wishes to visit the places one by one while returning to the query location in-between.
  - Go to the hotel between the museum visit and the jazz concert
  - NP-hard: proof by reduction from the Weighted Set Cover problem

- Type 2: Cost function:

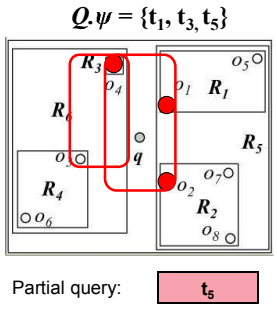
$$Cost(Q,\chi) = \max_{o \in \chi} Dist(o,Q) + \max_{o_i, o_j \in \chi} Dist(o_i, o_j)$$

- Application scenario
  - Visit places without returning to the query location in-between
  - E.g., go to a movie and then dinner
  - NP-hard: proof from reduction from the 3-SAT problem

Approximation Algorithm T1A1

- Exploit existing well known greedy algorithm
  - Partial query q_s: the unmatched part of the query keywords $Q.\psi$.

Object	Words	Dist(o,Q)
o_1	t_1, t_2	1.4
o_2	t_2, t_3	2.8
o_3	t_1, t_3	3
o_4	t_1, t_5	3.2
o_5	t_2, t_4	8
o_6	t_4, t_6	9
o_7	t_4, t_5	4.5
o_8	t_1, t_5	8



Exact Algorithm Without an Index T1E1

- Aim: Develop an exact algorithm with a running time that is exponential in the number of query keywords, not the number of objects.
  - The number of query keywords is small.

Exact Algorithm Without an Index – T1E1

Find all objects that cover parts of the query. Subsets of size 1 that yield the lowest costs.

t_1	t_3	t_5	t_1, t_3	t_1, t_5	t_3, t_5	t_1, t_3, t_5
o1	o2	o4	o3	o8	null	null
1.4	2.8	3.2	∞	∞	∞	∞

Find the lowest costs for subsets of size 2.

t_1	t_3	t_5	t_1, t_3	t_1, t_5	t_3, t_5	$t_1,$
o1	o2	o4	o3	o1,o4	o2,o4	nu
1.4	2.8	3.2	3	4.6	6.0	∞

t_1	t_3	t_5	t_1, t_3	t_1, t_5	t_3, t_5	t_1, t_3, t_5
o1	o2	o4	o3	o1,o4	o2,o4	o3,o4
1.4	2.8	3.2	3	4.6	6.0	6.2

$Q.\psi = \{t_1, t_3, t_5\}$

Partition	Objects	Cost
t_1, t_3, t_5	∅	null
t_1	t_3, t_5	$o_1+(o_2+o_4)$	7.4
t_3	t_1, t_5	$o_2+(o_1+o_4)$	7.4
t_5	t_1, t_3	o_4+o_3	6.2

Better than {o1, o2, o4} as found by T1A1.

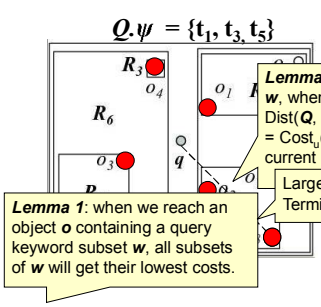
Exact Algorithm Using an Index T1E2

- Drawbacks of the exact algorithm without an index (T1E1)
  - Checks too many objects that do not contain a query keyword, e.g., o5 and o6
  - All the objects containing part of the query keywords are read, which is unnecessary

Improvements?

- Use the IR-tree to avoid reading unnecessary objects.
- Process objects in ascending order of their distance to the query to avoid reading all objects that cover part of the query.

Exact Algorithm Using an Index – T1E2



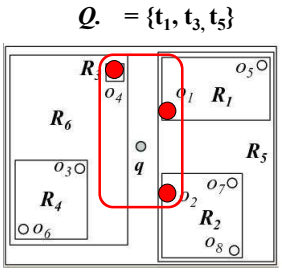
Object	Words	Dist(o,Q)
o_1	t_1, t_2	1.4
o_2	t_2, t_3	2.8
o_3	t_1, t_3	3
o_4	t_1, t_5	3.2
o_5	t_2, t_4	8
o_6	t_4, t_6	9
o_7	t_4, t_5	4.5
o_8	t_1, t_5	8

t_1	t_3	t_5	t_1, t_3	t_1, t_5	t_3, t_5	t_1, t_3, t_5
o1	o2	o4	o3	o4	o2,o4	o3,o4
1.4	2.8	3.2	3	4	6	6.2
M	M	M	M	M	M	M

- M: subset that already gets the lowest cost
- V: subset that has an upper bound cost value

Approximation Algorithm 1 T2A1

- For each query keyword, find the nearest object covering it using an IR-tree. The group of these object serve as the result set.

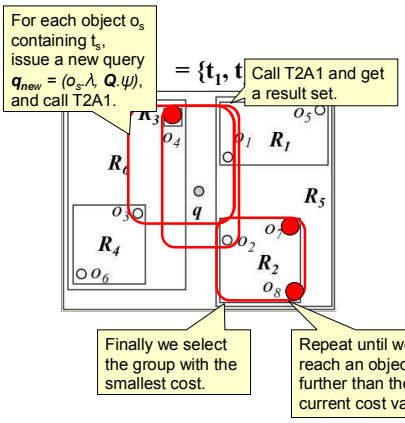


Object	Words
o_1	t_1, t_2
o_2	t_2, t_3
o_3	t_1, t_3
o_4	t_1, t_5
o_5	t_2, t_4
o_6	t_4, t_6
o_7	t_4, t_5
o_8	t_1, t_5

$$Cost(q,\chi) = Dist(o_4,q) + Dist(o_2,o_4)$$

Approximation Algorithm 2 – T2A2

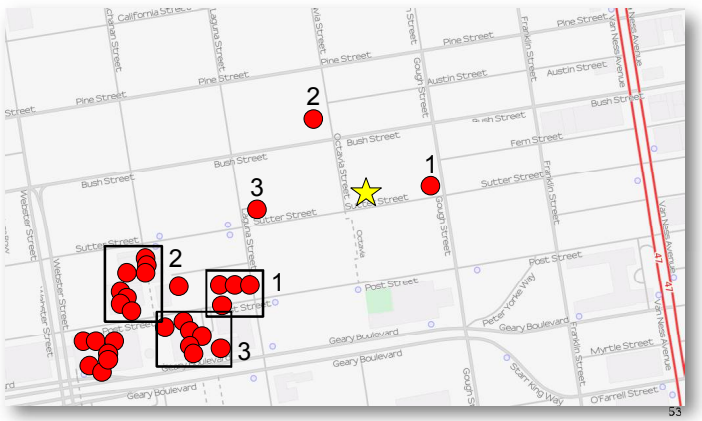
- Utilize the first approximation algorithm:



Object	Words
o_1	t_1, t_2
o_2	t_2, t_3
o_3	t_1, t_3
o_4	t_1, t_5
o_5	t_2, t_4
o_6	t_4, t_6
o_7	t_4, t_5
o_8	t_1, t_5

Top-*k* Groups Query Illustration

- Query location: ★ (Kenmore Hotel, SF)
- Query keyword: Restaurant



Top-*k* Groups Query

- Objects: $p = \langle \lambda, \psi \rangle$ (location, text description)
- Query: $q = \langle \lambda, \psi, k \rangle$ (location, keywords, # of objects)

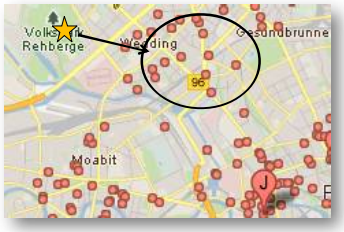
- Ranking function

$$rank_q(G) = \alpha \frac{\beta dist(q.\lambda, G) + (1-\beta) diam(G)}{\max D} + (1-\alpha) TR_G(q.\psi, G)$$

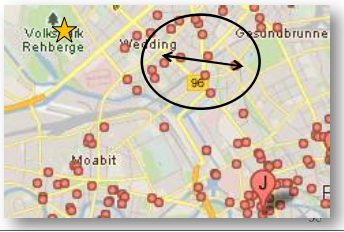
  - $0 \le \alpha \le 1$ and $0 \le \beta \le 1$
  - Distance: $dist(q.\lambda, G) = \min_{o \in G} \| q.\lambda, o.\lambda \|$
  - Diameter: $diam(G) = \max_{o_1, o_2 \in G} \| o_1.\lambda, o_2.\lambda \|$
  - The text relevance function favors large groups and groups where the query keywords are distributed evenly among group objects.
  - Groups are disjoint

Problem Definition

- Distance to the group
  - Distance to the nearest object

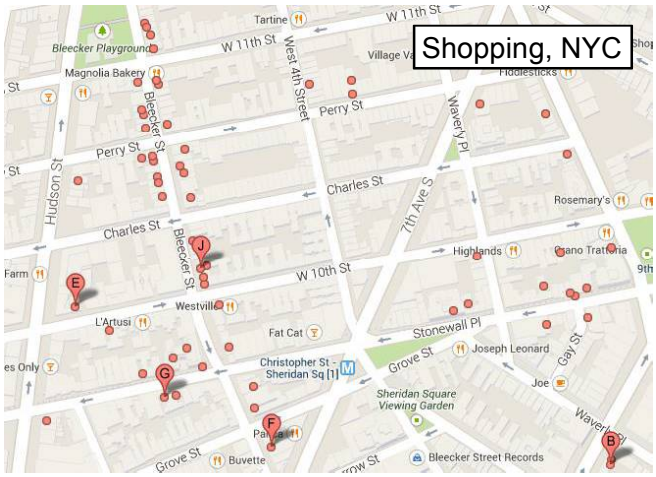


- Group diameter
  - Maximum distance between two objects



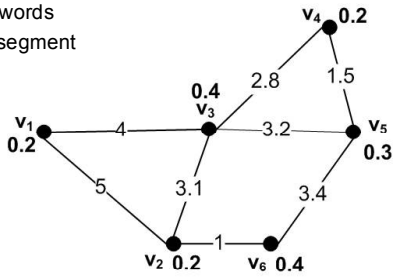
Road Networks

Road Networks

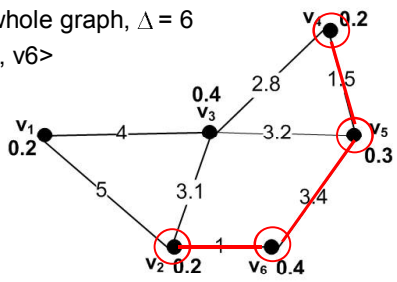


Problem Formalization

- Road network graph *G*
  - A node represents a road junction point or a location, associated with a set of keywords
  - An edge represents a road segment
- Region *R*
  - A connected subgraph of *G*
- Nodes are weighted
  - Relevance to the query
  - Query-independent weights (e.g., popularity or rating) are also possible



- $q = \langle \lambda, \psi, \Delta \rangle$
  - λ: a rectangular query range
  - ψ: keywords
  - Δ: a road segment length constraint
- Retrieves the region with largest weight given the length constraint and the query range
- Example: λ = the whole graph, $\Delta = 6$
- Result: <v2, v4, v5, v6>



Place ranking using GPS records, directions queries

Finding Spatial Web Objects

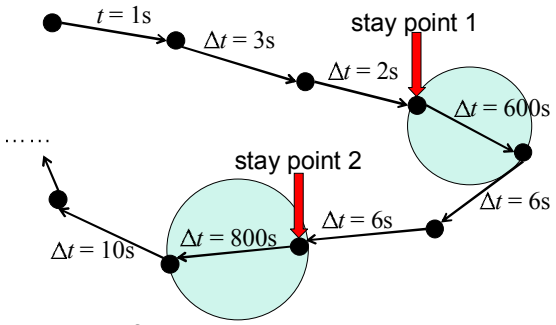
- Massive volumes of location samples from moving objects are becoming available.
  - GPS location records *(oid, x, y, t)*
  - Location records based on Wi-Fi and cellular positioning
- How can we utilize this content for identifying spatial web objects?
  - Can be used as a supplement to business directories
  - Potential benefit: more up to date

From GPS Records to Places

- Step 1: Extract stay points from raw trajectories
- Step 2: Cluster stay points with existing algorithms
- Step 3: Sample stay points from clusters, reverse geocode them, and obtain their semantics from yellow pages
- Step 4: Split and merge clusters to obtain semantic locations

Step 1: Extract Stay Points

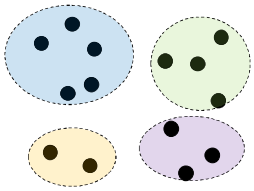
- Stay: two consecutive records with a time gap larger than some threshold t_{th} (e.g., 10 minutes)



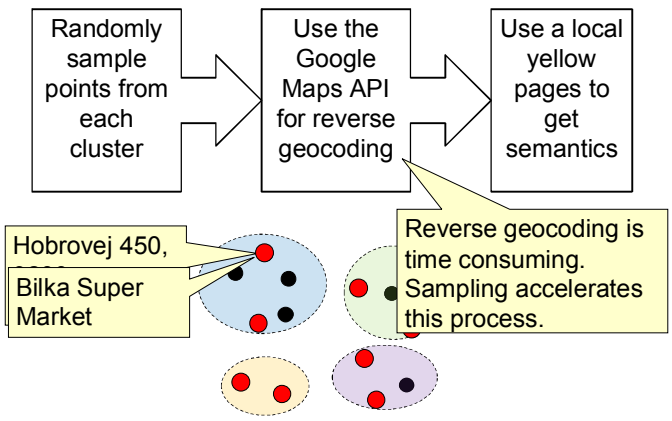
- Stay point: the first point in a stay (the end point)
- Data set: 76,139 stay points

Step 2: Cluster Stay Points

- Use existing spatial clustering algorithms
  - K-means: 7056 clusters
  - OPTICS: 7088 clusters

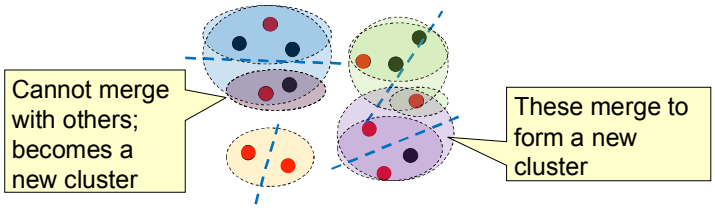


Step 3: Sampling, Reverse Geocoding, Semantics



Step 4: Splitting and Merging

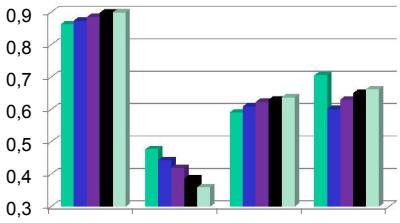
- Splitting
  - Cluster points in a cluster to obtain sub-clusters
  - Split a cluster if it has sub-clusters with different semantics
- Merge two clusters with similarity larger than a threshold
  - Similarity: consider user lists, semantics lists, average entry times, average stay durations



Experimental Study

- Data
  - Colle…enmark
  - 119 …
  - Sam…
  - 105,…
- Step 1
  - 76,1…
- Steps 2
  - ~6,5…
  - Clus…

Performance



Sampling 2 points in the clusters achieved by K-means.

High purity means that most objects in a cluster belong to the same class.

…alized mutual …ation – …e cluste… …e groun… smaller the better.

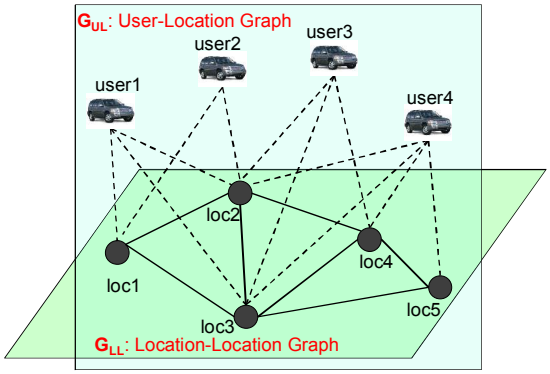
The number of clusters (x 10k).

GPS-Based Place Ranking

- Step 5 ranking
  - Ranking metrics: Precision@n, MAP, nDCG, Runtime
- Exploit different aspects of the location records
  - The more visits, the more significant
  - The longer the durations of visits, the more significant
  - The more distinct visitors, the more significant
  - The longer the distances traveled to visit, the more significant
  - The more "near- … place is.
  - The more a place is visited by objects that visit significant places, the more significant it is.

Two-Layered Graph

- G_{LL} : a link represents a trip between two locations
- G_{UL}: a link represents a visit of a user to a location



Results

	Rank-by-visits	Rank-by-durations	HITS-based
MAP	0.2020	0.2126	0.062
P@20	0.45	0.45	0.1
P@50	0.36	0.38	0.12
nDCG@20	0.8261	0.8324	…
nDCG@5…	…	0.7747	…
Runtime (…	…	107	…

	…	…	Unified	ST-Unified
MAP	…	…	0.4060	0.4274
P@20	…	…	0.9	0.95
P@50	…68	0.52	0.74	0.76
nDCG@20	0.9411	0.9031	0.9678	0.9897
nDCG@50	0.9226	0.8827	0.9402	0.9717
Runtime (ms)	2209	3540	4234	4318

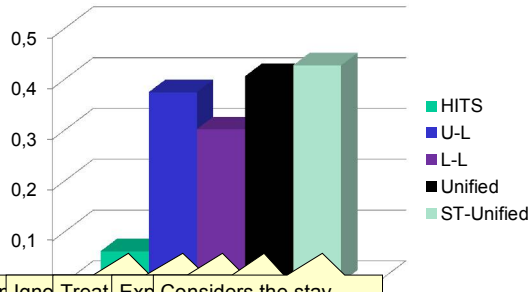
15 significant locations in the top-20 result given by the annotators.

…nt with the ideal ranking according to user annotations.

ST-Unified performs the best

Results

- MAP (Other metrics exhibit the same trend)



No r… loca… time… user… rank…

Igno… amir… not … relat… loca…

Treat… does … exper…

Exp… link… bett… L-L…

Considers the stay durations and distances between locations; performs the best.

Directions Query Based Place Ranking

- How can we use directions queries for assigning significance to places?
  - The queries will proliferate as navigation goes online.
- Idea: a query is a vote that *y* is an important place.
- Exploit different aspects of the queries
  - Count-based: The more queries to *y* @ *t*, the more significant *y* is (@ *t*).
  - Distance-based: The longer the distances the more the more significant y is.
  - Locality-based: The more queries $x \rightarrow y$, the more significant *y* is for users close to *x*.

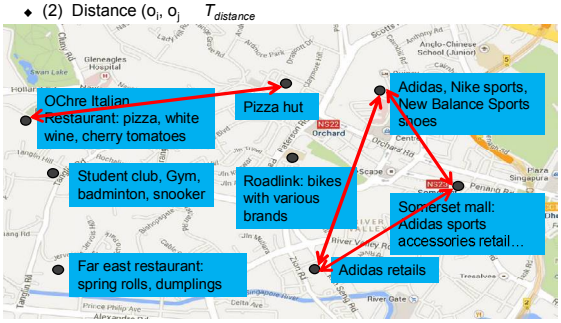
Experimental Study

- Using query logs from Google
- The most obvious competitor is reviews and ratings.
- Similar quality as reviews
- Better coverage than reviews
- Better temporal granularity than reviews
  - Examples of finer temporal granularity: after-work bar, weekday lunch restaurant

Other Functionality

Spatio-Textual Similarity Join

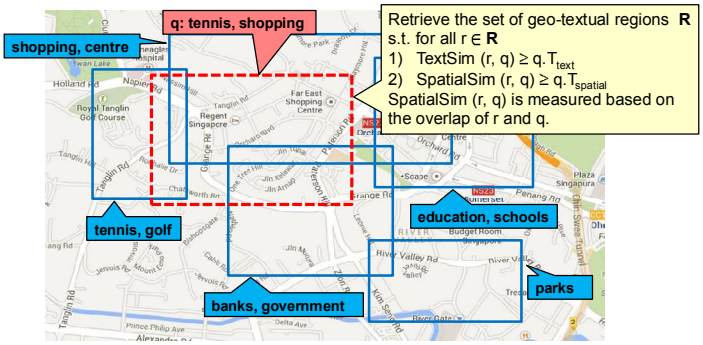
- Text Similarity Threshold T_{text}
- Spatial Distance Threshold $T_{distance}$
- Objective: Retrieve all pairs of geo-textual objects (o_i, o_j) s.t.
  - (1) TextSim (o_i, o_j) T_{text}
  - (2) Distance (o_i, o_j $T_{distance}$



Bouros et al. *Spatio-Textual Similarity Join*, PVLDB'12

Spatio-Textual Similarity Query

- A query region (rectangle)
- A set of keywords
- Thresholds of text similarity and spatial similarity



Fan et al.: *SEAL: Spatio-Textual Similarity Search*, PVLDB 12

More Types of Queries Examples

- **Approximate** String Search in Spatial
- Top-k Spatial Keyword Queries **on Road Networks.** Rocha-Junior and Nørvåg.
- **Diversified** Spatial Keyword Search On **Road Networks**. Zhang et al.
- Desks: **Direction-Aware** Spatial Keyword Search
- **Distributed** Spatial Keyword Querying on Road Networks. Luo et al.
- **Authentication** of Moving Top-k Spatial Keyword Queries. Wu et al.
- **Reverse Keyword Search** for Spatio-Textual Top-k Queries in Location-Based Services. Lin et al.
- Keyword-Aware **Continuous kNN** Query on **Road Networks.** Zheng et al.
- …

Summary and challenges

Summary

- The web is going mobile and has a spatial dimension.
- Many queries have local intent
- Spatial keyword queries
  - *k* nearest neighbor queries
  - Continuous *k* nearest neighbor queries
  - Using nearby relevant content for place ranking
  - Retrieve a set of objects that collectively best satisfy a query
  - Retrieve *k* sets of objects that best satisfy a query

Next Steps

- Which functionality to serve when?
  - Ex: mineral water, dumplings
  - How can context be used for determining user intent?
- More sophisticated ranking!
  - Which signals to use?
  - How to combine them into a function (e.g., as a sum)?
  - Which weight parameters to use (e.g., a weight for each term)?
  - What is the relevant context for this?
    - Dependence on location
    - Dependence on keywords
    - Dependence on search history
    - Dependence on social network
    - Dependence on time
- Evaluation?
  - Which functionality is best where and when and for who?

Further Steps

- Structured queries and Amazon-style and social queries
  - Ample opportunities for much more customization of results
- Build in feedback mechanisms
  - "Figuring out how to build databases that get better the more people use them is actually the secret source of every Web 2.0 company" –
- Avoid parameter overload
  - Problem vs. solution parameters
  - Hard-to-set, impossible-to-set parameters – relevance decreases exponentially with the number of such parameters

Acknowledgments and Readings

- Skovsgaard, A. and C. S. Jensen: Finding top-k relevant groups of spatial web objects, VLDB J. 24(4): 537-555 (2015)
- Cao, X., G. Cong, T. Guo, C. S. Jensen, and B. C. Ooi: Efficient Processing of Spatial Group Keyword Queries, ACM TODS, 40(2), 48 pages (2015)
- Cao, X., G. Cong, C. S. Jensen, M. L. Yiu: Retrieving Regions of Interest for User Exploration. PVLDB 7(9): 733-744 (2014)
- L. Chen, G. Cong, C. S. Jensen, D. Wu: Spatial Keyword Query Processing: An Experimental Evaluation. PVLDB 6(3): 217-228 (2013)
- Bøgh, K. S., A. Skovsgaard, C. S. Jensen: GroupFinder: A New Approach to Top-K Point-of-Interest Group Retrieval, PVLDB (2013)
- Wu, D., M. L. Yiu, C. S. Jensen: Moving Spatial Keyword Queries: Formulation, Methods, and Analysis. TODS, 38(1): 7 (2013)
- Cao, X., L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, M. L. Yiu: Spatial Keyword Querying. ER, pp. 16-29 (2012)
- Wu, D., M. L. Yiu, G. Cong, and C. S. Jensen: Joint Top-K Spatial Keyword Query Processing. TKDE, 24(1): 1889-1903 (2012)
- Cao, X., G. Cong, C. S. Jensen, J. J. Ng, B. C. Ooi, N.-T. Phan, D. Wu: SWORS: A System for the Efficient Retrieval of Relevant Spatial Web Objects. PVLDB, 5(12): 1914-1917 (2012)
- Wu, D., G. Cong, and C. S. Jensen: A Framework for Efficient Spatial Web Object Retrieval. VLDBJ, 21(6): 792-822 (2012)
- Wu, D., M. L. Yiu, C. S. Jensen, G. Cong: Efficient Continuously Moving Top-K Spatial Keyword Query Processing. ICDE, pp. 541-552 (2011)
- Cao, X., G. Cong, C. S. Jensen, B. C. Ooi: Collective Spatial Keyword Querying. SIGMOD, pp. 373-384 (2011)
- Cao, X., G. Cong, C. S. Jensen: Retrieving Top-k Prestige-Based Relevant Spatial Web Objects. PVLDB 3(1): 373-384 (2010)
- Cong, G., C. S. Jensen, D. Wu: Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects. PVLDB 2(1): 337-348 (2009)

Acknowledgments and Readings

- Dingming Wu, Byron Choi, Jianliang Xu, Christian S. Jensen: Authentication of Moving Top-k Spatial Keyword Queries. IEEE Trans. Knowl. Data Eng. 27(4): 922-935 (2015)
- Lei Chen, Xin Lin, Haibo Hu, Christian S. Jensen, Jianliang Xu: Answering why-not questions on spatial keyword top-k queries. ICDE 2015: 279-290
- Lei Chen, Jianliang Xu, Xin Lin, Christian S. Jensen, Haibo Hu: Answering why-not spatial keyword top-k queries via keyword adaption. ICDE 2016: 697-708
- Anders Skovsgaard, Christian S. Jensen: Top-k point of interest retrieval using standard indexes. SIGSPATIAL/GIS 2014: 173-182
- Anders Skovsgaard, Darius Sidlauskas, Christian S. Jensen: Scalable top-k spatio-temporal term querying. ICDE 2014: 148-159
- Anders Skovsgaard, Darius Sidlauskas, Christian S. Jensen: A Clustering Approach to the Discovery of Points of Interest from Geo-Tagged Microblog Posts. MDM (1) 2014: 178-188

- **Gao Cong, Christian S. Jensen: Querying Geo-Textual Data: Spatial Keyword Queries and Beyond. SIGMOD Conference 2016: 2207-2212**

Thank you for your attention.