Encontrando os locais de interesse com maior popularidade a partir do critério espacial e textual

Finding the most popular places of interest from the spatial and textual judgment

Cláudio Moisés Valiense de Andrade[1], João B. Rocha-Junior[1]

[1]Pós-Graduação em Computação Aplicada - Universidade Estadual de Feira de Santana

claudiovaliense@gmail.com, joao@uefs.br

Abstract. *Spatial data is increasingly present in our daily lives. We use various applications that use this data, such as Google Maps and Uber. There are a huge number of interesting questions that can be performed on these data, for example, a user is interested in staying at the hotel that has the largest number of restaurants in their neighborhood, in this query their current location (latitude and longitude), radius (e.g. 100 m) and the keyword "restaurant". This project proposes a new query type that can select the best spatial objects taking into account the number of relevant spatio-textual objects, for a given set of query keywords, in its neighborhood. We present algorithms to process this query efficiently and evaluate the algorithms proposed in real datasets.*

Keywords*: Spatial Database. Space-textual Query. Preferential Query.*

Resumo. *Dados espaciais estão cada vez mais presentes em nosso dia a dia. Usamos diversas aplicações que utilizam esses dados, como o Google Maps e Uber. Há um grande número de perguntas interessantes que podem ser realizadas nesses dados, por exemplo, um usuário tem interesse de ficar hospedado no hotel que tenha a maior quantidade de restaurantes na sua vizinhança, nesta consulta utiliza-se sua localização atual (latitude e longitude), raio (e.g. 100 m) e a palavra-chave "restaurante". Este projeto propõe um novo tipo de consulta que pode selecionar os melhores objetos espaciais levando em conta o número de objetos espaço-textuais relevantes, para um determinado conjunto de palavras-chave de consulta, em sua vizinhança. Apresentamos algoritmos para processar essa consulta de forma eficiente e avaliar os algoritmos propostos em conjuntos de dados reais.*

Palavras-chave: *Banco de Dados Espacial. Consulta Espaço-textual. Consulta preferencial.*

1. INTRODUÇÃO

Com o passar do tempo as máquinas estão realizando uma quantidade maior de tarefas e criando novas facilidades aos humanos, por exemplo, a utilização do *smartphone* para visualizar o mapa de uma cidade. Outras pesquisas estão voltadas a otimizar o gerenciamento de recurso de um cidade inteira [Costa el al. 2018].

No nosso dia a dia, utilizamos diversas aplicações que manipulam dados, por exemplo, quando estamos dirigindo pela cidade e usamos o aplicativo *Waze*[1] para ser informado em qual parte do trajeto existem buracos ou sinalizações. O aplicativo está realizando consultas espaciais sobre os dados, gerando informações para atender a intenção de pesquisa do usuário.

Com a popularidade de dispositivos móveis com *GPS* (Global Positioning System), a quantidade de dados produzidos com referências geográficas (latitude e longitude) tem crescido rapidamente. Além da localização, outras informações podem estar associadas aos objetos (e.g. nome, tamanho, tipo, preço, mensagem) [Yiu et al. 2007]. Objetos que possuem localização espacial (referência geográfica) e texto são chamados de objetos espaço textuais.

Existem diversos tipos de consultas capazes de selecionar dados espaciais, denominadas consultas espaciais [Du et al. 2005, Zhang et al. 2006, Yiu et al. 2007, Rocha-Junior et al. 2010, Rocha-Junior et al. 2011, de Almeida and Rocha-Junior 2016]. Estas consultas permitem selecionar objetos espaciais de interesse, a partir de um raio, vizinhança ou texto.

Uma consulta que vem sendo bastante estudada é a consulta espacial preferencial [Yiu et al. 2007, de Almeida and Rocha-Junior 2016]. Funciona da seguinte forma, dado um conjunto de objetos espaciais de interesse (locais) e um conjunto de objetos espaciais de referência, esta consulta retorna os k melhores objetos de interesse, levando-se em consideração o maior escore entre os objetos de referência dentro da região espacial de interesse fornecida pelo usuário (e.g. 100m). Nesta consulta, cada objeto de referência tem um escore que é definido por um provedor de classificação específico (e.g. *ZAGAT*[2],*IFOOD*[3]).

A Figura 1, contém objetos de interesse p (hotéis) e objetos de referência f (bares), o círculo delimita a região espacial de interesse (raio). Assumindo que um usuário deseja ficar hospedado em um hotel que tenha o bar com maior classificação, considerando o entorno do hotel (e.g. raio de 100m). Ao utilizar a consulta espacial preferencial para retornar os dois melhores hotéis (k=2), a consulta retorna os hotéis p_1 em primeiro e p_3 em segundo. O hotel p_1 é retornado em primeiro, porque no seu entorno tem um bar f_1 com escore 0.9, enquanto que o melhor bar no entorno de p_3 é f_5 com escore 0.8. Esta consulta considera apenas o escore do melhor objeto de referência dentro da região espacial de interesse.

[1]www.waze.com
[2]www.zagat.com
[3]www.ifood.com.br

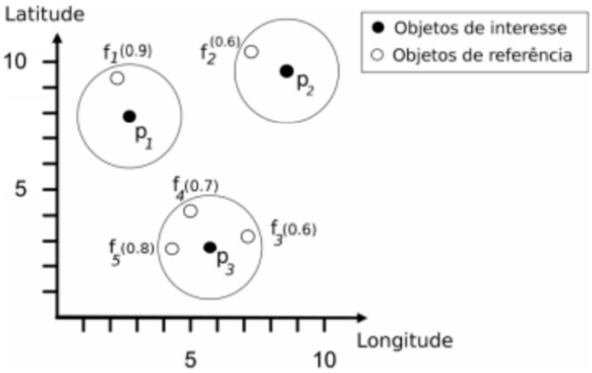

Figura 1. Representação hotéis e pontos de referência

A consulta espacial preferencial tradicional tem dois problemas principais: 1) não leva em consideração a quantidade de objetos de referência na vizinhança espacial, apenas o escore do melhor objeto de referência e 2) não é possível selecionar os objetos de referência à partir da descrição textual dos objetos. A consulta espacial preferencial tradicional assume que o escore dos objetos são números estáticos. Entretanto, para a maioria das aplicações estes objetos estão associados a um texto descritivo. Neste caso, o escore de um objeto ou a sua relevância pode ser computada, levando-se em consideração a similaridade textual destes objetos com palavras-chave de busca.

A consulta proposta nesta pesquisa, denominada de Consulta Espaço-Textual Preferencial Por Popularidade (CETPP). Diferente da consulta tradicional de [Yiu et al. 2007], que retorna o objeto de interesse analisando o objeto de referência com maior escore no seu entorno, a *CETPP* leva em consideração todos os objetos espaço-textuais de referência que tem relevância textual maior que o mínimo pré-definido e que estão na região espacial de interesse definida pelo usuário. A partir disso, o escore dos objetos de interesse é calculado, somando todos os objetos de referência cuja a relevância textual fique acima do limiar e que estejam presentes na região de interesse.

A Figura 2 é composta por objetos espaço-textuais de interesse p e objetos de referência f. Assumindo que um cliente deseja comprar um apartamento (objeto de interesse) que tenha muitas escolas (objetos de referência) na sua proximidade, ele pode então fornecer uma palavra-chave de interesse “escola” e indicar a região que considera próxima (e.g. 1km). Ao realizar esta consulta na base de dados da Figura 2, ele tem como resposta o objeto p_3 com melhor escore, visto que p_3 possui 2 objetos de referência na sua vizinhança espacial que satisfaz a palavra-chave de busca, enquanto que p_2 tem apenas 1 e p_1 não tem nenhum.

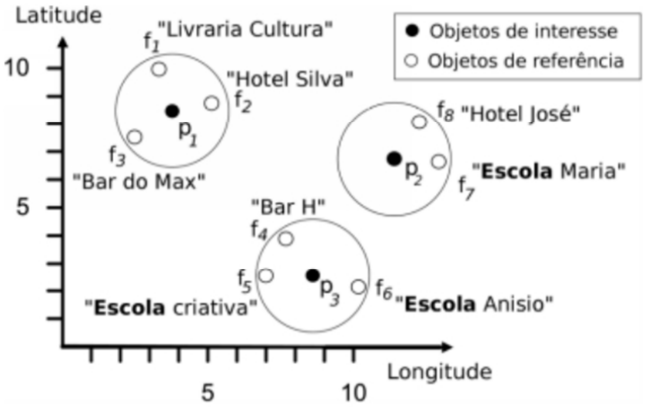

Figura 2. Objetos de interesse (apartamento) e objetos espaço-textuais de referência

As principais contribuições deste artigo são:

1. Especificar a nova consulta "Espaço-Textual Preferencial Por Popularidade";

2. Desenvolver algoritmos para processar essa nova consulta de forma eficaz;

3. Avaliar os algoritmos propostos utilizando bases de dados reais.

O restante deste artigo esta organizado da seguinte forma: Seção 2 apresenta os trabalhos relacionados; Seção 3 especifica a consulta deste trabalho; Seção 4 apresenta as bases de dados utilizadas; Seção 5 apresenta os algoritmos para processar a consulta deste trabalho; Seção 6 apresenta a avaliação experimental; Seção 7 as considerações finais.

2. TRABALHOS RELACIONADOS

Nesta seção, são apresentadas algumas pesquisas com pontos similares deste trabalho. Uma consulta espacial bastante estudada na literatura foi definida por [Du et al. 2005, Xia el al. 2005], denominada de *Maximum Influence Optimal Influence Query*, busca o ponto de máxima influência ao adicionar um novo objeto de interesse. Este novo objeto de interesse precisa atender a maior quantidade de objetos de referência. Por exemplo, assumindo que a empresa *McDonald's* tem interesse em abrir uma nova unidade, na Figura 3 os objetos espaciais *c* representam as casas de clientes, enquanto que *p* são os locais possíveis para instalação da nova unidade do *McDonald's*. Dada esta distribuição dos objetos *c* e *p*, o local que atende a maior quantidade de clientes é o p_1. Este é o local de máxima influência.

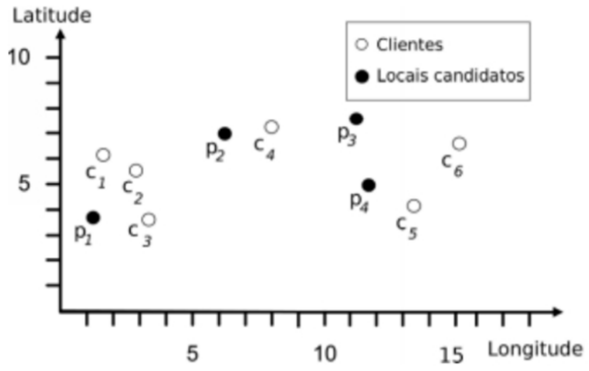

Figura 3. Exemplo de localização de máxima influência. Fonte: Adaptada [Xia et al. 2005]

Algumas consultas têm o objetivo de relacionar dados espaciais e temporais, Cho e Chung (2007) propôs uma consulta espaço temporal para responder questões como, "Qual é o número total de acidentes no raio de 1km de cada escola em Junho de 2005?" [Cho and Chung 2007]. Diferente da consulta apresentada nesta pesquisa, esta consulta não utiliza palavras-chave para verificar a similaridade textual dos objetos.

Outra consulta tem o objetivo de relacionar dados espaciais com textuais. Definida por de Almeida e Rocha-Junior (2016), denominada de *Top-k Spatial Keyword Preference Query* tem o objetivo de buscar os top-k objetos de interesse de acordo com a relevância textual dos objetos de referência presentes na sua vizinhança espacial. Utiliza as palavras-chave para filtrar os objetos de referência [de Almeida and Rocha-Junior 2016].

Exemplo. Na Figura 4, supondo que um usuário está interessado em alugar um apartamento próximo a um objeto relevante para as palavras-chave "escola" e "infantil", e indicar a região que considera próxima (e.g. 1km). Ao realizar esta consulta na base de dados da Figura 4, ele tem como resposta o objeto p_2 e p_3, sendo que p_3 é o objeto mais relevante, pois possui na sua vizinhança espacial f_4 que é mais relevante para suas palavras-chave de busca.

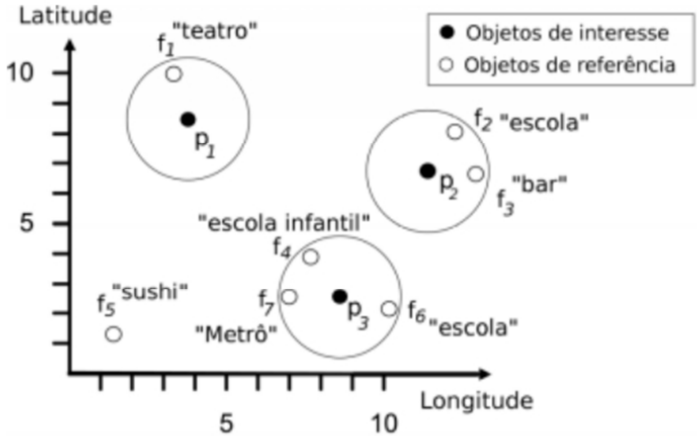

Figura 4. Objetos de interesse (apartamento) e objetos espaço-textuais de referência

Assim como a consulta espacial preferencial tradicional, a consulta proposta por [de Almeida and Rocha-Junior 2016] também só considera o escore do melhor objeto de referência. Onde o escore do objeto de referência é computado, levando em consideração a similaridade textual entre o texto dos objetos de referência e as palavras-chave da consulta. Diferente desta consulta, a consulta *CETPP* utiliza um limiar de relevância textual e um critério de vizinhança espacial para selecionar os objetos de referência e computa o escore dos objetos de interesse de acordo com o número de objetos de referência que atendem os dois critérios.

3. ESPECIFICAÇÃO DA CONSULTA

Definição. Dado um conjunto de objetos de interesse P, onde cada objeto $p \epsilon P$ possui uma coordenada espacial $p = (p.x, p.y);$ e um conjunto de objetos espaço-textuais de referência F, onde cada objeto $f \epsilon F$ possui uma coordenada espacial $(f.x, f.y)$ e um texto $f.D, f = (f.x, f.y, f.D)$. A Consulta Preferencial por Popularidade Q tem 4 parâmetros, $Q = \{Q.D, Q.r, Q.k, Q.\sigma\}$, onde $Q.D$ é o conjunto de palavras-chave de interesse, $Q.r$ é o limiar que define o valor máximo da distância entre um objeto de interesse e de referência (raio), $Q.k$ é o número de resultados esperados e $Q.\sigma$ é o limiar que define o valor mínimo de similaridade textual que um objeto de referência deve ter para ser considerado textualmente relevante.

A consulta Q retorna o $Q.k$ objetos em P com os maiores escores. O escore de um objeto p, representado por $\tau\ (p)$, é a quantidade dos objetos de referência presentes na vizinhança espacial de interesse e que são textualmente relevantes para as palavras-chave de busca. A Equação (1) descreve essa especificação:

$$\tau\ (p) = \sum\ \{f \epsilon F \mid dist\ (p,f) \leq Q.r : \theta\ (f.D,Q.D) >\ Q.\sigma\ \} \qquad (1)$$

onde $\theta\ (f.D, Q.D)$ é a relevância textual (similaridade textual) entre o texto do objeto espaço-textual de referência $f.D$ e a palavras-chave de consulta $Q.D$. Nesta pesquisa, nós computamos a relevância textual como definida por [Rocha-Junior et al. 2011] e usamos a métrica *Haversine* na função da distância $dist\ (p,f)$ entre um objeto p e um objeto espaço-textual de referência f.

4. BASE DE DADOS

Os dados utilizados nesta pesquisa foram coletados no *OpenStreetMap*[4] (OSM), que é um projeto de mapeamento para representar locais e ruas, todo o sistema é desenvolvido pela comunidade, onde diversos usuários de todo o mundo fornecem informações de forma cooperativa. Através do sistema é possível selecionar uma área de interesse e exportar no formato *Extensible Markup Language (XML)*.

Neste trabalho, a partir da base do *OSM* foi selecionado três cidades: Feira de Santana, Salvador e São Paulo. Em cada experimento, os objetos do tipo restaurante é extraída para representar o conjunto de interesse P, os demais objetos (e.g. hotel, hospital, escola) são utilizados para compor o conjunto dos objetos de referência F. Foi selecionado o objeto de interesse do tipo restaurante por representar uma quantidade volumosa nas diferentes cidades.

[4] https://www.openstreetmap.org

Na Tabela 1 é apresentado os dados dos objetos de interesse e referência, a quantidade total de termos da coleção e a quantidade de termos únicos. Foi selecionado 3 cidades com pequeno, médio e grande porte na base dados do *OSM*.

Tabela 1. Base de dados do OpenStreetMap

Cidades	**P**	**F**	**Termos Total**	**Termos Únicos**
Feira de Santana (FSA)	85	494	1196	682
Salvador (SSA)	353	2512	5707	2609
São Paulo (SP)	1081	7956	20082	6424

5. ALGORITMOS

Nos algoritmos abaixo, é utilizado *P* para representar o conjunto de objetos de interesse, *F* o conjunto de objetos de referência, o conjunto *Q.D* as palavras-chave de busca, *Q.r* que representa a distância máxima de um objeto de referência, *Q.k* a quantidade de retorno de objetos de interesse e $Q.\sigma$ que representa o escore mínimo que um objeto de referência precisa ter para ser considerado. Os Algoritmos 1, 2 e 3 processam a consulta *CETPP* retornando o mesmo conjunto de saída.

5.1 Algoritmo Baseline

O Algoritmo 1, denominado de *Popularity Spatial Keywords Preference Query* (Baseline Algorithm) processa a consulta deste trabalho. Este algoritmo calcula o escore de cada objeto de interesse a partir dos objetos de referência que atenderam aos critérios espacial e textual. Para cada objeto de interesse é percorrido todos os objetos de referência.

Na linha 1, é criada a MinHeap H para armazenar no topo os elementos que tem o menor escore. Linha 2 percorre o conjunto de objetos de interesse, para cada objeto de interesse é realizada a contagem dos objetos de referência. Na linha 5, é calculada a distância *Haversine* entre o local de interesse e o local de referência, feito isso é verificado se a distância é menor ou igual à definida pelo usuário. Na linha 6, é calculado o escore do objeto de referência em relação à similaridade textual com as palavras-chave *Q.D*. Após calculado, é testado se o escore do objeto é acima do limiar mínimo. Caso seja, este objeto de referência entra na contagem. A condição na linha 8 existe para evitar a adição de objetos de interesse que não tiveram nenhum elemento de referência. Nas linhas 9-11 é adicionado o novo objeto e removido do topo o objeto que tem menor escore de contagem.

No cálculo da complexidade do Algoritmo 1, considere que *n* e *m* são as representações do conjunto dos objetos de interesse *P* e referência *F*. Este algoritmo tem complexidade $\theta\ (n*m)$ por ter que percorrer todos os objetos de referência para cada objeto de interesse.

Algoritmo 1: Popularity Spatial Keywords Preference Query (Baseline Algorithm)

Input: P (Objects of Interest), F (Objects of Reference), $Q.D$ (Keywords), $Q.r$ (Radius), $Q.k$ (Number o results), $Q.\sigma$ (Limit Score text)
Output: MinHeap that maintains the k best objects of interest

```
1  MinHeap H ← ∅;
2  forall p ∈ P do
3      count = 0;
4      forall f ∈ F do
5          if dist(p,f) ≤ Q.r then
6              if θ(f.D, Q.D) ≥ Q.σ then
7                  count++;
8      if count > 0 then
9          H.add(p, count);
10         if H.size() > Q.k then
11             H.remove();
12 return H;
```

5.2 Algoritmo Aplicando Primeiro Filtro Textual

O Algoritmo 2, denominado de *Popularity Spatial Keywords Preference Query Text First* (Text First Algorithm) consiste em reduzir o conjunto de objetos de referência *F*, calculando inicialmente a similaridade das palavras-chave de busca com o texto dos objetos de referência.

A linha 2 cria *F'* como vazio que representa o novo conjunto de objetos de referência. Linha 3-5 adiciona apenas em *F'* os objetos que atenderam ao critério de similaridade textual. Linha 6 percorre o conjunto de objetos de interesse. Linha 8-10 percorre o conjunto de objetos de referência que passaram no limiar de similaridade, é verificado se estes objetos atende a distância mínima definida pelo usuário. A condição na linha 11 existe para evitar de adicionar objetos de interesse que não tiveram nenhum elemento de referência. Nas linhas 12-14 é adicionado o novo objeto e removido do topo o objeto que tem menor escore de contagem.

A grande vantagem deste algoritmo é que ele percorre o conjunto *F* uma vez para identificar os objetos que são textualmente relevantes *F'* e depois só percorre os objetos dentro de *F'* para verificar se eles atendem ao critério espacial. Entretanto, no pior caso, o tamanho de *F'* pode ser igual ao de *F*, logo a complexidade é a mesma do algoritmo anterior θ *(n*m)*.

Algoritmo 2: Popularity Spatial Keywords Preference Query First Similarity (Text First Algorithm)

Input: P (Objects of Interest), F (Objects of Reference), $Q.D$ (Keywords), $Q.r$ (Radius), $Q.k$ (Number o results), $Q.\sigma$ (Limit Score text)
Output: MinHeap that maintains the k best objects of interest

```
1  MinHeap H ← ∅;
2  F' ← ∅;
3  forall f ∈ F do
4      if θ(f.D, Q.D) ≥ Q.σ then
5          F'.add(f);
6  forall p ∈ P do
7      count = 0;
8      forall f ∈ F' do
9          if dist(p,f) ≤ Q.r then
10             count++;
11     if count > 0 then
12         H.add(p, count);
13         if H.size() > Q.k then
14             H.remove();
15 return H;
```

5.3 Algoritmo Aplicando Primeiro Filtro Espacial

No Algoritmo 3, denominado de *Popularity Spatial Keywords Preference R-tree* (Spatial First Algorithm), utiliza a estrutura *R-tree* com o objetivo de retornar de forma mais rápida o conjunto de objetos de referência que atenda ao critério de distância espacial *Q.r.*

Na entrada deste algoritmo, a *RTree_F* representa o conjunto dos objetos de referência no formato de uma *R-Tree*. Na linha 1 é criado uma *MinHeap H* para armazenar os objetos de referência. Na linha 4-7 é retornado o conjunto de objetos de referência que atenderam ao critério espacial. Para cada objeto de referência, é calculado a similaridade textual, caso este atinja o limiar textual, é contado no escore do objeto de interesse. A condição na linha 7 existe para evitar de adicionar objetos de interesse que não tiveram nenhum elemento de referência. Nas linhas 9-11 é adicionado o novo objeto e removido do topo o objeto que tem menor escore de contagem.

Para o cálculo de complexidade deste algoritmo, como é preciso percorrer cada objeto de interesse e realizar a consulta na *R-tree* dos objetos de referência que atenderam ao critério espacial, apesar da consulta na *R-Tree* ter complexidade de θ *(log n)* como mostrado por [Arge et al. 2008], mas o resultado retornado tem tamanho *m*, portanto a complexidade do algoritmo é θ *(n * m)*.

Algoritmo 3: Popularity Spatial Keywords Preference Query R-tree (Spatial First Algorithm)

Input: P (Objects of Interest), *RTree_F* (RTree of reference objects), $Q.D$ (Keywords), $Q.r$ (Radius), $Q.k$ (Number o results), $Q.\sigma$ (Limit Score text)

Output: MinHeap that maintains the k best objects of interest

```
1  MinHeap H ← ∅;
2  forall p ∈ P do
3      count = 0;
4      foreach f ∈ RTree_F.search(p.x,p.y,Q.r) do
5          if θ(f.D, Q.D) ≥ Q.σ then
6              count++ ;
7      if count > 0 then
8          H.add(p, count);
9          if H.size() > Q.k then
10             H.remove();
11 return H;
```

6. Avaliação experimental

Um ambiente de teste foi montado para rodar os experimentos, que constitui de um computador com a seguinte configuração: Processador Core i3-6100 3.7 GHz; HD 500 GB; Memória 4GB. Os resultados apresentados neste trabalho não considera o tempo de criação das estruturas de dados para realizar a consulta espacial, é analisado apenas o tempo da consulta.

Cada consulta foi executada com 100 repetições, com o intuito de evitar problemas de performance por abertura de algum processo inesperado em um determinado momento.

Na Tabela 2 é mostrado os parâmetros que serão estudados nos experimentos. O valor padrão esta em negrito, esses valores são frequentemente utilizado por usuários em mecanismos de pesquisa.

Tabela 2. Parâmetros utilizados na consulta com os valores padrões destacados em negrito

Parâmetros	Valores
Número de resultados (k)	1, **3**, 5, 10
Número de palavras-chave	1, 2, **3**, 4, 5
Valores do raio em metros	100, 500, **1000**, 2000, 3000
Limite de similaridade textual	0.1, 0.2, **0.3**, 0.4, 0.5
Bases de dados	FSA, **SSA**, SP

6.1 Variando o Número de Resultados k

Esta seção avalia o tempo de resposta variando o número de resultados. Na Figura 5 o *Baseline Algorithm* apresenta o pior tempo de resposta por precisar percorrer todo o conjunto dos objetos de referência para cada objeto de interesse.

O Algoritmo *Text First Algorithm* percorre uma vez o conjunto dos objetos de referência F para identificar os objetos que são textualmente relevantes F', após isto, percorre para cada objeto de interesse o conjunto reduzido F'.

O Algoritmo *Spatial First Algorithm* indexa os objetos de referência F, para posteriormente realizar a busca destes objetos de forma mais eficiente em relação ao critério espacial.

Os algoritmos não sofreram variação significativa no tempo de resposta ao alterar a quantidade de resultados retornados (k), isto ocorre porque todos os algoritmos precisam percorrer todo o conjunto de objetos de interesse independente do valor de k.

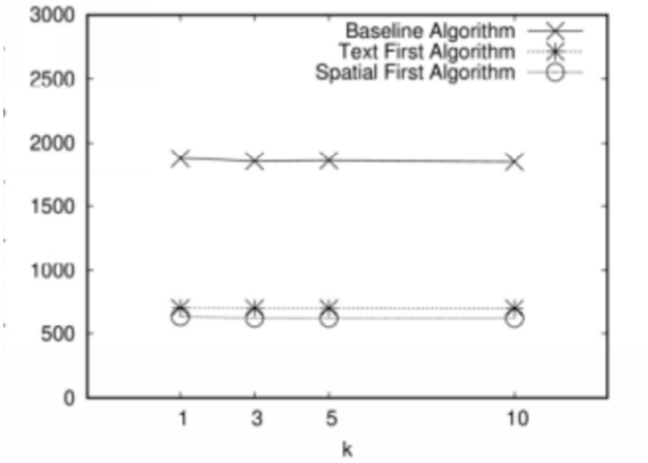

Figura 5. Variando o número de resultados *k*.

6.2 Variando a Quantidade de Palavras-Chave

Esta seção avalia o tempo de resposta variando a quantidade de palavras-chave na consulta. Analisando a Figura 6, o Algoritmo *Baseline Algorithm* não tem variação

significativa na sua resposta em relação ao número de palavras-chave de busca, isto ocorre porque este algoritmo percorre o mesmo conjunto de objetos de referência independente do número de palavras-chave.

O Algoritmo *Text First Algorithm* apresenta melhor tempo conforme aumenta a quantidade de palavras-chave de busca. Isto ocorre porque quanto maior a quantidade de palavras-chave na busca, é menor a possibilidade que o objeto de referência tenha na sua descrição todas as palavras-chave, implicando em um menor escore de similaridade com os objetos de referência. Como o Algoritmo *Text First Algorithm* reduz o conjunto *F* identificando os objetos que são textualmente relevantes, com o aumento das palavras-chave são reduzidos o conjunto dos objetos relevantes.

O Algoritmo *Spatial First Algorithm* apresenta crescimento pouco significativo no tempo de resposta. O aumento esta relacionado à quantidade de comparações com um maior número de palavras-chave nos objetos de referência.

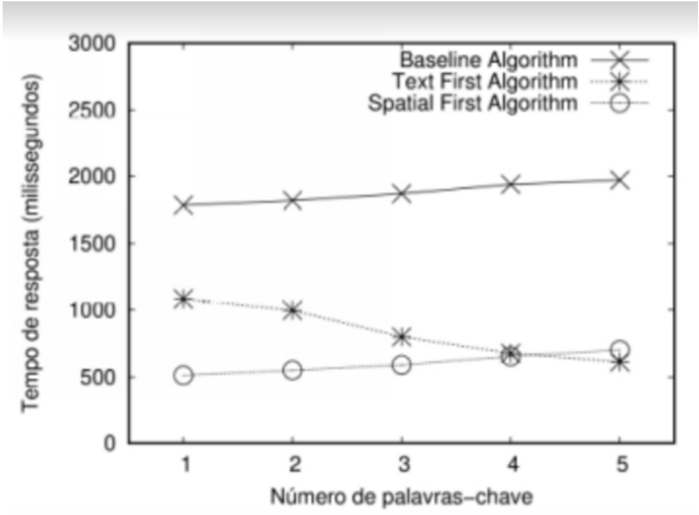

Figura 6. Variando a quantidade de palavras-chave na consulta

6.3 Variando o Raio na Consulta

Esta seção avalia o tempo de resposta variando o raio da consulta. Na Figura 7, o aumento do tempo de resposta nos Algoritmos *Baseline Algorithm* e *Spatial First Algorithm* ocorre porque com o aumento do limiar da distância espacial, há o crescimento da quantidade de objetos de referência que atendem à condição espacial, isto implica em uma quantidade maior de objetos de referência que serão calculados a similaridade textual.

O Algoritmo *Text First Algorithm* não tem impacto significativo com relação ao limiar da distância espacial, porque a vantagem deste algoritmo é reduzir o conjunto de objetos de referência *F* com a similidade textual. Portanto, para o algoritmo *Text First Algorithm*, independente do limiar espacial serão percorridos a mesma quantidade de objetos referência.

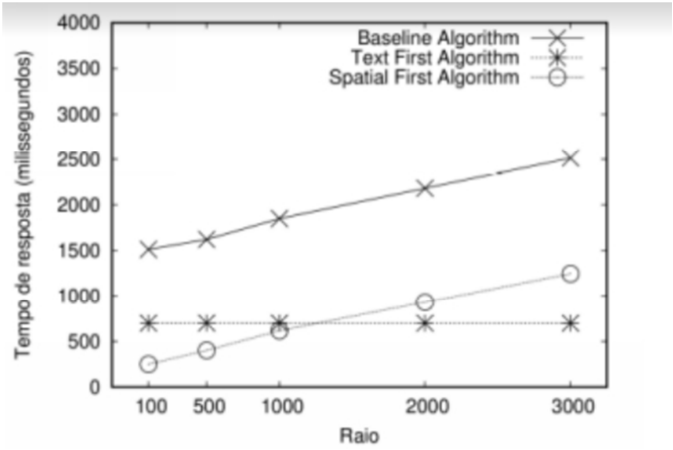

Figura 7. Variando o raio na consulta

6.4 Variando o limiar da Similaridade Textual da Consulta

Esta seção avalia o tempo de resposta variando o limiar de similaridade textual da consulta. Na Figura 8, o Algoritmo *Baseline Algorithm* e *Spatial First Algorithm* não tem variação no seu tempo de resposta, isso ocorre porque é necessário percorrer o mesmo conjunto de objetos de referência independente da variação de similaridade textual.

O Algoritmo *Text First Algorithm* apresenta um melhor tempo de resposta conforme é aumentado o limiar de similaridade. Isto acontece porque quanto maior o limiar de similaridade textual, é mais difícil que o objeto de referência atinja essa condição, com isso diminuindo o tamanho do conjunto dos objetos de referência que atingiram esse limiar. Como este algoritmo reduz o conjunto dos objetos de referência *F* identificando os objetos textualmente relevantes, se beneficia com este acréscimo por analisar um conjunto reduzido. Note que o limite de similaridade superior a 0.3, o Algorimo *Text First Algorithm* supera o *Spatial First Algorithm* em tempo de resposta.

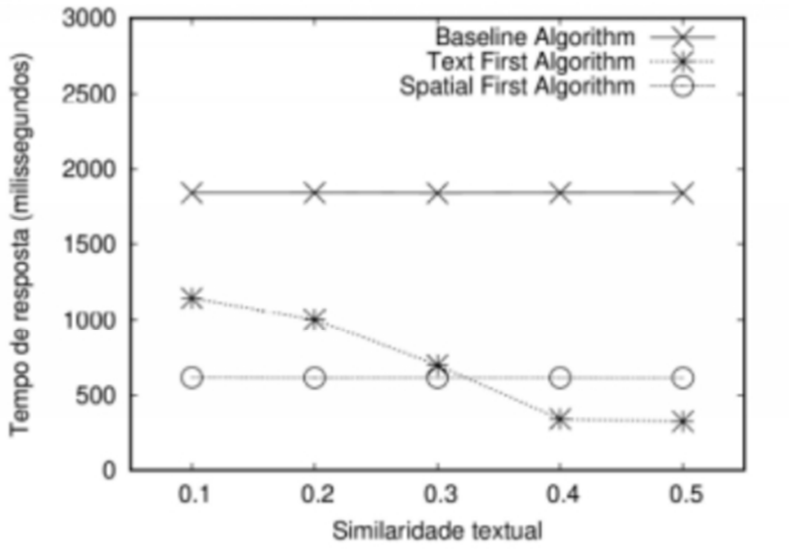

Figura 8. Variando a similaridade textual da consulta

6.5 Variando as bases de dados

Neste experimento, nos estudamos o impacto no tempo de resposta das diferentes bases de dados. O eixo *y* (tempo de resposta) da Figura 9, é apresentado em escala logarítmica, devido a grande diferença entre os resultados dos algoritmos nas bases de dados.

A Figura 9 apresenta o tempo de resposta para três diferentes bases de dados. Este gráfico comprova a eficiência dos algoritmos nas diferentes bases e demonstra que quanto maior a base, maior o ganho destas abordagens. No gráfico 9, na cidade de Feira de Santana, o algoritmo *Spatial First Algorithm* é cerca de 2 vezes mais rápido que o *Baseline Algorithm*, enquanto que para a cidade de São Paulo é cerca de 10 vezes mais rápido que o *Baseline Algorithm*.

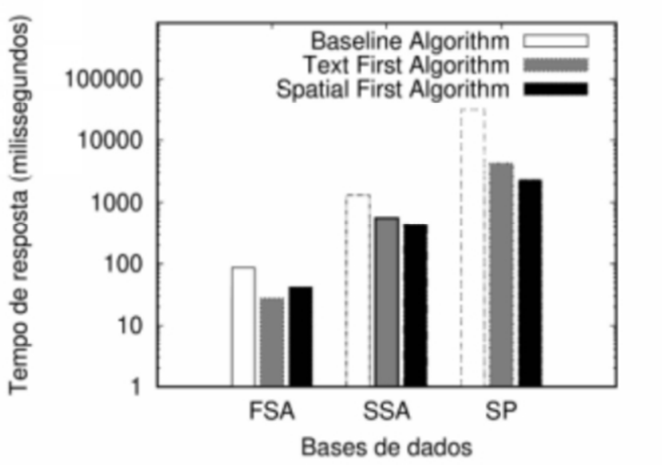

Figura 9. Variando as bases de dados

7. CONSIDERAÇÕES FINAIS

Neste artigo, nós apresentamos a Consulta Espacial Textual Por Popularidade (CETPP) que tem o objetivo de identificar os objetos de interesse de acordo com a popularidade de objetos espaço-textuais relevantes para as palavras-chave de busca em sua vizinhança espacial. Nós apresentamos três algoritmos para processar essa consulta, *Baseline Algorithm*, *Spatial First Algorithm* e *Text First Algorithm*. Esta consulta possui vantagens, como a verificação da relevância textual dos objetos de referência no momento da consulta, permite que o usuário utilize quaisquer termos para representar os objetos espaço-textuais em sua busca. A utilização de um limite mínimo para relevância textual auxilia na filtragem de resultados, eliminando os objetos de referência com pouca ou nenhuma relevância do conjunto resposta. Os algoritmos foram implementados e puderam ser validados utilizando bases de dados reais.

No futuro, nós pretendemos desenvolver novos algoritmos para processar a *CETPP* de forma mais eficiente. Pretendemos utilizar índices híbridos (textual e espacial) e paralelizar os algoritmos da Seção 5.

REFERÊNCIAS

[Arge et al. 2008] Arge, L., Berg, M. D., Haverkort, H., and Yi, K. (2008). The priority r-tree: A practically efficient and worst-case optimal r-tree. TALG, 4(1):9.

[Cho and Chung 2007] Cho, H.-J. and Chung, C.-W. (2007). Indexing range sum queries in spatio-temporal databases. Information and Software Technology, 49(4):324–331.

[Costa et al. 2018] Costa, D. G., Duran-Faundez, C., Andrade, D. C., Rocha-Junior, J. B., and Just Peixoto, J. P. (2018). Twittersensing: An event-based approach for wireless sensor networks optimization exploiting social media in smart city applications. Sen-sors, 18(4).

[de Almeida and Rocha-Junior 2016] de Almeida, J. P. D. and Rocha-Junior, J. B. (2016). Top-k spatial keyword preference query. Journal of Information and Data Manage- ment, 6(3):162.

[Du et al. 2005] Du, Y., Zhang, D., and Xia, T. (2005). The optimal-location query. In In-ternational Symposium on Spatial and Temporal Databases, pages 163–180. Springer.

[Rocha-Junior et al. 2011] Rocha-Junior, J. B., Gkorgkas, O., Jonassen, S., and Nørvåg, K. (2011). Efficient processing of top-k spatial keyword queries. In SSTD, volume 6849 of LNCS. Springer.

[Rocha-Junior et al. 2010] Rocha-Junior, J. B., Vlachou, A., Doulkeridis, C., and Nørvåg, K. (2010). Efficient processing of top-k spatial preference queries. Proceedings of the VLDB Endowment, 4(2):93–104.

[Xia et al. 2005] Xia, T., Zhang, D., Kanoulas, E., and Du, Y. (2005). On computing top-t most influential spatial sites. VLDB, pages 946–957.

[Yiu et al. 2007] Yiu, M. L., Dai, X., Mamoulis, N., and Vaitis, M. (2007). Top-k spatial preference queries. In Proceedings - ICDE, pages 1076–1085. IEEE.

[Zhang et al. 2006] Zhang, D., Du, Y., Xia, T., and Tao, Y. (2006). Progressive computation of the min-dist optimal-location query. VLDB, pages 643–654.