Using Natural Language Processing to Perform Spatial Searches of Open Street Map Features In ArcGIS

Faculty Advisors: Dr. Alex Klippel
Dr. Jan Wallgrün
Gary Huffman
grh145@psu.edu
Presentation Overview

- Problem Overview
- Search Using ArcGIS Desktop “Out of the box”
- Open Street Map
- Natural Language Processing
- Spatial Language and Spatial Representations
- Proposed System Architecture and Implementation
- Next Steps and Follow-on Work
Problem Overview

Motivation

- Limitations of search in ArcGIS Desktop
  - SQL-based -- RDBMS and shapefiles
  - Slow with large databases
- Pervasiveness of Open Street Map
  - Lots of data
  - Contains places of interest that are not available in other data sets
  - Increasing popularity: Apple iPhoto & Four Square
  - Basemap option in ArcGIS Desktop (yet no search)
- User familiarity with natural language search

Objective

- I will integrate aspects of Natural Language Processing into ArcGIS to search Open Street Map data
  - Spatial Search and Topological Relationships
  - Attribute Search
Search Using ArcGIS Desktop

- **Select by Attribute**
  - Users construct SQL statements
  - Requires basic understanding of database schema and SQL
  - Use of ‘like’ in query gets close matches

- **Find**
  - Fuzzy search
  - Inflexible (Starbucks != starbucks)
  - Slow with large datasets
Search Using ArcGIS Desktop

- Select by Location
  - Spatial and topological relationships
    - Containment (In/On)
    - Intersection
    - Equality
    - Nearby (Proximity distance)
- Specify source and target layers
- Features selected beforehand
- Differences and meanings of the spatial selection methods
Open Street Map

- The Wikipedia of geospatial information
- User contributed and moderated data
- Roughly 21GB of compressed XML formatted geospatial data
  - Nodes (Points)
  - Ways (Lines and Polygons)
  - Relations (Lines and Polygons)
- On-line search interface (Nominatim) and a Web Service API
- Available as a basemap layer in ArcGIS Desktop
  - All or nothing
  - Cannot search the basemap
- Available on-line at www.openstreetmap.org
Natural Language Processing

- Natural language - It’s how humans talk
  - We say: “Where are the Starbucks in Vienna?”
  - We don’t say: “Select * where Name = ‘Starbucks’ and City = ‘Vienna’ and State = ‘VA’”

- Natural Language Processing
  - Part computer science, part linguistics
  - Goal is to get computers to understand human language
  - Non-trivial problem
    - Reading
      - Noun as in “He gave a reading.”
      - Verb as in “I was reading earlier today.”
      - Proper noun (place name) as in “Reading, Pennsylvania”
Natural Language Processing (cont.)

- NLP systems try to understand the linguistic, grammatical and semantic meaning inherent in language
  - Parts of Speech
  - Named Entities
  - Parsing and Tokenization

- Consider the following statements that use the preposition IN:
  - The crack in the jar.
  - The flowers in the vase.

- Systems implementing NLP are all around us - we use them daily
  - Spam/junk e-mail filter
  - Calendar events from e-mail messages
  - Internet search
    - How to repair Maytag dishwasher with leaky door?
  - Internet map searches for geographic information
    - Bing Maps - Where are the Starbucks in Vienna?
On-line Maps and Search

Where are the Starbucks near Vienna, VA

Not what you wanted?

SPONSORED LISTINGS

-75% to Starbucks Shop! We Are Giving 75% Starbucks Coupons. Click Here Offer Expires Tomorrow. GiftCardWorld.org/Starbucks

1. Starbucks
   881 Leesburg Pike, Vienna, VA
   (703) 893-5125
   Website
   Directions: Save: Send: Menu

2. Starbucks
   8520 Leesburg Pike Ste D, Vienna, VA
   (703) 763-7037
   Website
   Directions: Save: Send: Menu

3. Starbucks
   107 Maple Ave W, Vienna, VA
   (703) 242-0890
   Website
   Directions: Save: Send: Menu

4. Starbucks
   207 Maple Ave E, Vienna, VA
   (703) 938-1003
   Website
   Directions: Save: Send: Menu
How do we describe where things are in the world?

- In language, often through the use of spatial prepositions
- Where are the Starbucks \textit{IN} Vienna?

Coventry and Garrod, 2004
Proposed System Functions

- **Information Retrieval System**
  - Ingest spatial data - OSM Ingestion Tool
    - Load OSM XML data into a geodatabase
  - Index database - Also part of OSM Ingestion Tool
    - Create index of searchable terms using Lucene Search Engine
  - Linguistically analyze query using NLP tools - Linguistic Analyzer
    - Part of Speech (POS) Tagger
    - Named Entity Recognition (NER)
    - OSM Special Phrases Dictionary

- **Search data - Search Engine**
  - Attribute query
  - Spatial query - point/polygon and polygon/polygon relationships

- **Present results - Handed as Search Engine results**
  - Expanded Plug-in window shows hits and allows visualization within ArcGIS Desktop map display
Ingest OSM into a Spatial Database

- **OSM Ingestion Tool** - ArcGIS Desktop .Net Plug-in

```xml
<nodet d='1667265750' timestamp='2012-03-09T13:30:02Z' uid='621860'>
  <tag k='amenity' v='fast_food' />
  <tag k='cuisine' v='burger' />
  <tag k='name' v='McDonald's' />
</node>
```

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECTID</td>
<td>OBJECT ID</td>
</tr>
<tr>
<td>Shape</td>
<td>Geometry</td>
</tr>
<tr>
<td>NAME</td>
<td>Text</td>
</tr>
<tr>
<td>TYPE</td>
<td>Text</td>
</tr>
<tr>
<td>GENERIC</td>
<td>Text</td>
</tr>
<tr>
<td>OSMID</td>
<td>Long</td>
</tr>
</tbody>
</table>
Index Spatial Database with Lucene

- Lucene is an Open Source full-text search library written in Java and .Net
- Uses an inverted index providing fast document retrieval
- Higher performance than traditional database SQL search
- Index is stored on file system and can be searched independent of database

### OSM Database Table

<table>
<thead>
<tr>
<th>ID</th>
<th>...</th>
<th>GENERIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>name=Starbucks</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>name=Dunkin Donuts</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>amenity=cafe</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>tourism=hotel</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>amenity=fast_food</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>name=Starbucks</td>
</tr>
</tbody>
</table>

### Inverted Index

<table>
<thead>
<tr>
<th>Token</th>
<th>Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>amenity=cafe</td>
<td>1, 2, 3, 6</td>
</tr>
<tr>
<td>name=Starbucks</td>
<td>1, 2</td>
</tr>
<tr>
<td>name=Dunkin Donuts</td>
<td>2</td>
</tr>
<tr>
<td>tourism=hotel</td>
<td>4</td>
</tr>
<tr>
<td>name=Homestead</td>
<td>4</td>
</tr>
<tr>
<td>name=Domino Pizza</td>
<td>5</td>
</tr>
<tr>
<td>amenity=fast_food</td>
<td>5</td>
</tr>
<tr>
<td>addr:street=Leesburg Pike</td>
<td>1</td>
</tr>
<tr>
<td>addr:street=West Maple Ave.</td>
<td>6</td>
</tr>
</tbody>
</table>
Linguistically Analyze Query String

- **Linguistic Analyzer** - ArcGIS Desktop Java plug-in
  - Parse query string (e.g., Starbucks in Vienna)
  - Determine query type: attribute or spatial
    - Spatial preposition **IN** or **ON** suggests a spatial query
    - Otherwise, attribute query
  - Determine feature types participating in query
    - Initially limited to points and polygons
  - Identify named entities and parts of speech using Stanford’s coreNLP Java library
    - NER Module: Organizations, Locations (e.g., Starbucks, Vienna)
    - POS Module: Prepositions, Nouns (e.g., in, Starbucks, Vienna)
  - Populate Query Object to pass on to the Search Engine
Linguistic Analyzer (cont.)

- Query string “Starbucks in Vienna”
  - POS Tagger: Starbucks/NNP in/IN Vienna/NNP
  - NER: Starbucks [ORGANIZATION] in [OTHER] Vienna [LOCATION]
  - Located spatial preposition IN → Spatial Query
  - Tokenize query string into phrases before and after preposition
    - Left side → Starbucks; Search Lucene index for points and polygons
    - Right Side → Vienna; Search Lucene index for polygons (only)
  - How to construct the search?
    - Named Entities (Organizations and Locations) are likely stored in a name tag as in name=Starbucks and name=Vienna
    - Unmatched entities are checked against Special Phrase Dictionary
Linguistic Analyzer (cont.)

- Special Phrases Dictionary
  - Built using OSM’s Nominatim User Contributed Special Phrases
  - Maps common OSM tag values to fully expanded search strings
    - cafe → amenity=cafe
    - hotels → amenity=hotel

- Query Object populated with parameters for Search Engine
  - Could be multiple objects depending on index search results

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERY TYPE</td>
<td>SPATIAL</td>
</tr>
<tr>
<td>SPATIAL PREPOSITION</td>
<td>IN</td>
</tr>
<tr>
<td>SOURCE FEATURE CLASS</td>
<td>POINT</td>
</tr>
<tr>
<td>SOURCE SELECTION STRING</td>
<td>Name-Starbucks</td>
</tr>
<tr>
<td>TARGET FEATURE CLASS</td>
<td>POLYGON</td>
</tr>
<tr>
<td>TARGET SELECTION STRING</td>
<td>Name-Vienna</td>
</tr>
</tbody>
</table>
Search Engine

- Executes Search/Selection of Features based upon:
  - Parameters provided in Query Object
  - For spatial searches, which topological relationship is expressed by the user?
    - Ambiguity in language → What is really meant by IN?
    - True for both point/polygon and polygon/polygon relationships
    - For the “Starbucks in Vienna” example, which figure could it be? Does it matter?
    - If B represents a Starbucks on the outskirts of Vienna, does the user want to see it?

- There is a difference in ArcGIS Desktop spatial relationships for the graphic:
  - INTERSECT
  - WITHIN
  - COMPLETELY WITHIN
  - HAVE_THEIR_CENTER_IN
Example - Starbucks in Vienna

- Linguistic Analyzer passes the Query Object to Search Engine

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUERY TYPE</td>
<td>SPATIAL</td>
</tr>
<tr>
<td>SPATIAL PREPOSITION</td>
<td>IN</td>
</tr>
<tr>
<td>SOURCE FEATURE CLASS</td>
<td>POINT</td>
</tr>
<tr>
<td>SOURCE SELECTION STRING</td>
<td>Name-Starbucks</td>
</tr>
<tr>
<td>TARGET FEATURE CLASS</td>
<td>POLYGON</td>
</tr>
<tr>
<td>TARGET SELECTION STRING</td>
<td>Name-Vienna</td>
</tr>
</tbody>
</table>

- All Starbucks are selected from the POINTS layer
- Polygon Representing Vienna selected from POLYGON layer

Map data (c) OpenStreetMap contributors, CC-BY-SA
Example - Starbucks in Vienna

- Topological Relationship represented by Query String with clickable results - Only 2 Starbucks are inside the Boundary polygon for Vienna

<table>
<thead>
<tr>
<th>No</th>
<th>Feature</th>
<th>OSM Key Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Starbucks</td>
<td>amenity:cafe</td>
</tr>
<tr>
<td>2</td>
<td>Starbucks</td>
<td>amenity:cafe</td>
</tr>
</tbody>
</table>
Next Steps and Follow-on Work

- Build the system!
- Determine where I can present my work
- Expand support for additional Spatial Prepositions and more complex query strings
  - Near - need to resolve ambiguity in Near (scale dependency)
  - “Starbucks in Vienna near the airport”
- Expand Query Terms using other NLP Tools and Ontologies
  - Wordnet
- Train NLP Tools on Geographic-term oriented corpora
- Generalize Tool to work with non-OSM data
- Determine how to release code based upon Stanford and OSM Licenses
References


Questions?