AN ONTOLOGY BASED APPROACH FOR GEOSPATIAL DATA INTEGRATION OF AUTHORITATIVE AND CROWD SOURCED DATASETS

(1) University of Nottingham, NOTTINGHAM, UNITED KINGDOM ; (2) Ordnance Survey, SOUTHAMPTON, UNITED KINGDOM

BACKGROUND AND OBJECTIVES

The progress of national and international spatial data infrastructures such as the UK Location Programme and European Commission INSPIRE SDI, contrasted against crowd-sourced geospatial databases such as OpenStreetMap, creates a promising opportunity for exploring data integration between crowd sourced information and authoritative data. A further aim of this research was to look into the mid-term and long-term effects of crowd sourcing technologies on the change intelligence operations of national mapping agencies (NMAs). This paper explains an ontology-based approach for geospatial data integration between crowd sourced and authoritative data. An algorithm for feature matching based on ontology matching concepts is presented. An implementation of the algorithm is also presented, together with initial experimental results.

This research has been carried out to understand the issues of data integration between crowd-sourced information and authoritative data. Ordnance Survey (OS), as the national mapping agency of Great Britain, provides authoritative datasets with published data specifications, driven by a combination of user need and the history of national mapping with a remit to ensure real-world feature changes are reflected in the OS large-scale data within 6 months. OpenStreetMap (OSM), in contrast, relies on the availability of local mapping enthusiasts to capture changes, but through its more informal structure, can capture a broader range of features of interest to different sub-communities, such as cyclists or horse riders (Anand et al, 2010). Geospatial data integration (GDI) in this context refers to combining geographic data, including spatial and non-spatial data, from disparate sources, with differing conceptual, contextual and topographical representations. The paper investigates feature matching using a geosemantic algorithm for position and high level ontological description. The algorithm can be qualified as fuzzy as it combines probabilistic answers obtained from conceptual matching functions. This research used OS and OSM datasets as a case study for exploring techniques for integrating the authoritative and crowd sourced data.

An open source software application was developed to integrate geospatial data from disparate sources. The methodology, implementation and experimentation details of this research are described in this paper.

CASE STUDY

Ordnance Survey’s Integrated Transport Network (ITN) data and OpenStreetMap (OSM) road data for Portsmouth, UK were used as a case study to explore ontology based methodologies for integrating the two heterogeneous data sources. The two input data sets used are shown in Figure 1 and Figure 2. The OSM data has been filtered to select only the road features for the area. The OSM data set is larger than that of OS ITN to ensure all features in OS ITN can find their corresponding feature, if there is one, in OSM. In the OS ITN data set, there are 112 named roads and 4 Department for Transport (DfT) classified roads, while there are 145 named roads in OSM road data set.
The trust placed in geographic data and products is an important issue. These technologies will only prove useful if they are fit for their intended purpose, and uncertainty regarding the quality of user-generated content is often cited as a major obstruction to its wider use. Critics argue that amateur contributions may be highly erroneous and as such essentially invalid for serious academic or industrial uses, with Goodchild (2009) arguing that a crowdsourcing project should publish additional documentation and assessments reviewing quality issues. (Anand et al, 2010). Ontologies have been acknowledged to be the core methodology for capturing and sharing semantics of geospatial information. Ontologies, specifically domain-specific ontologies, are at the heart of most semantic approaches to interoperability (Klien E and Probst F 2005).

There has been previous research on ontologies which allow the use of probabilistic representation of categories (Costa and Laskey 2006). Reasoning mechanisms using such probabilistic information, which allow not only equivalent concepts but also the ‘most similar’ or the ‘least similar’ concepts to be inferred, are best suited for practical use of ontologies. However, current work in geospatial ontologies does not provide sufficient insight into the use of probabilistic knowledge. (Sen, 2008). Sen (2008) also points out that ontologies of geospatial entities need to be extended with probabilistic frameworks in order to enable rich and practical inferences such as concept similarity and concept overlaps.
For geospatial data integration between authoritative and crowd sourced data using ontologies a feature matching concept is considered. The main research question is how do we define data from different sources that are corresponding, i.e., referring to the same feature. The following basic relationships for the features in the crowd sourced and authoritative data are considered:

- **SamePlace**(featureA, featureB): featureA and featureB are in the same place (e.g., Portsmouth, UK).
- **Near**(featureA, featureB, m): featureB is within m metre buffer of featureA.
- **SameName**(featureA, featureB): featureA and featureB have a same name. Firstly a list with necessary definitions, such as “ST = Saint”, is kept. In addition, for comparing the name strings, an edit distance is defined.
- **SameCategory**(featureA, featureB): featureA and featureB are of a shared category.
- **Neighbour**(featureA, featureB, m): featureA and featureB have at least one point in common, given m metre fuzzy tolerance.
- **SameNeighbour**(featureA, featureB): featureA and featureB have at least one neighbour with a same name.

Based on the definition above, **SameFeature** is defined as following:

\[
\text{SameFeature}(\text{featureA} \text{ and } \text{featureB}) = \frac{\text{SamePlace} + \text{Near} + \text{SameName} + \text{SameCategory} + \text{SameNeighbour}}{5}
\]

One of the key problems for implementing a matching algorithm based on the above concept, is that there is no information on neighbours stored in one of the datasets used (OSM data). To solve this problem, a Network Building algorithm (Figure 3) is implemented. The pseudocode of the same is shown below.

```java
/** To generate topological connectivity, given a fuzzy tolerance */

NetworkBuilding (inputLineLayer, fuzzy):
   net ← NetGraph(fuzzy)
   for feature in inputLineLayer:
      name ← feature.getName()
      net.addFeature(name, feature)
   for featuresA in net.features.values():
      for featuresB in net.features.values():
         if featuresA != featuresB:
            // if within the fuzzy tolerance, they are neighbours
            featuresA.neighbour(featuresB)
            featuresB.neighbour(featuresA)

Class NetGraph(
   double fuzzy;
   Map<String name, ArrayList<Feature>> features;
)
```

**Figure 3: Pseudocode for Network Building**

Also there are key integration issues with incomplete dataset (in OSM data some required information is incomplete; for example some fields show name=NULL, OSM: highway=unclassified or NULL). To handle this problem, a probability based approach is taken. For example, for a named road, **SameFeature=SamePlace/Near/SameName/SameCategory/SameNeighbour**. The formula used for calculating the probability of two features being the same is defined as following:

\[
\text{Probability (SameFeature)} = \frac{w1 \times \text{SamePlace} + w2 \times \text{Near} + w3 \times \text{SameName} + w4 \times \text{SameCategory} + w5 \times \text{SameNeighbour}}{w1 + w2 + w3 + w4 + w5}
\]

A user-friendly interface is designed to allow users specify the buffer size and the fuzzy tolerance, as well as weights (w1, w2, and w3) to put on these constraints.
The feature matching algorithm designed is illustrated below as flow chart (Figure 4) and pseudocode of the feature matching algorithm is presented in Figure 5.

![Flowchart for Feature Matching](image)

**Figure 4: Flowchart for Feature Matching**
RESULTS
This research was focused on developing Open Source based software tools. The Open Source Geospatial Foundation (OSGeo) is an excellent example of a community initiative to support and promote the collaborative development of open geospatial technologies. OSGeo’s key mission is to promote the use of open source software in the geospatial industry and to encourage the implementation of open standards and standards based interoperability in its projects. The software development uses Quantum GIS (QGIS), which is an Open Source GIS application providing data visualization, editing, and analysis capabilities. QGIS is written in C++, and its GUI uses Qt library. QGIS is a volunteer driven project allowing the development of plug-ins using C++ or Python (qgis.org, 2010).

Software based on the methodology described above, was developed as a plug-in in QGIS. Its graphic user interface is shown below (Figure 6). It requires users to specify two input line layers, and some corresponding fields, such as name fields in both data sets. It enables users to conduct different experiments easily, by specifying fuzzy concepts, such as buffer size, and assigning different weights in score function variables.

```c
/** An ontology based approach to match two input features */

FeatureMatching (featureA, featureB):

    passPlaceCheck ← samePlaceCheck(featureA, featureB)
    samePlace ← calculate_%_place()

    if passPlaceCheck && near(featureA, featureB, bufferSize):
        near ← calculate_%_near(bufferSize)
        passName_CategoryTest ← False

    if bothHaveName(featureA, featureB):
        if sameName(featureA, featureB):
            sameName_Category ← calculate_%_name()
            passName_CategoryTest ← True
        else:
            if sameCategory(featureA, featureB):
                sameName_Category ← calculate_%_category()
                passName_CategoryTest ← True

    if passName_CategoryTest:
        passNeighbourTest ← False
        if sameNeighbour(featureA, featureB):
            sameNeighbour ← calculate_%_neighbour()
            passNeighbourTest ← True

    if passNeighbourTest:
        probability ← calculate(samePlace, near, sameName_Category, sameNeighbour)

        output(new DataModel)
```

**Figure 5: Pseudocode for Feature Matching**
Figure 6: GDI tool developed

With the inputs specified in Figure 6, outputs are shown below. Firstly, for geometric information (Figure 7), the more accurate one was maintained. According to studies done by Al-Bakri and Fairbairn (2010), OS data is more accurate than OSM data, compared to reference field survey (FS) data sets. In this case study, it seems reasonable to assume that the geometric information of authoritative OS ITN data is more accurate than that of OSM data. Thus, the geometry of matched roads in OS ITN was maintained. In addition, for attribute information (Figure 8), important information, such as name in both data sets, and interesting (tags of OSM) and unique information (e.g. TOID), was kept. Finally, newly calculated information, such as neighbours, score card and probability of matching, were added.
Figure 7: Output Geometry

| ROAD          | ROADNAME          | Town         | Neighbour          | Neighbour          | OSM_tag       | name_category | name | name | place | neighbour | probability |
|---------------|-------------------|--------------|--------------------|--------------------|---------------|---------------|------------|-------|-------|--------|----------|-------------|
| 1             | RAILWAY ארֶנֶס | Hamrun Road  | Whitfield Road     | Whitfield Road     | PLOTLE-ROAD.   | name_category | name | name | place | neighbour | probability |
| 2             | RAILWAY ארֶנֶס | PLOTLE-ROAD  | Whitfield Road     | Whitfield Road     | PLOTLE-ROAD.   | name_category | name | name | place | neighbour | probability |
| 3             | RAILWAY ארֶנֶס | Whitfield Road| Whitfield Road     | Whitfield Road     | PLOTLE-ROAD.   | name_category | name | name | place | neighbour | probability |
| 4             | RAILWAY ארֶנֶס | Whitfield Road| Whitfield Road     | Whitfield Road     | PLOTLE-ROAD.   | name_category | name | name | place | neighbour | probability |
| 5             | RAILWAY ארֶנֶס | Whitfield Road| Whitfield Road     | Whitfield Road     | PLOTLE-ROAD.   | name_category | name | name | place | neighbour | probability |

Figure 8: Output Attributes

EXPERIMENTATION RESULTS

111 out of 116 roads in OS ITN can find a corresponding road in OSM. These matched roads shown in Figure 7 are categorized according to the calculated probability of matching (Table 1).
<table>
<thead>
<tr>
<th>Probability of Matching</th>
<th>Number of Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.98</td>
<td>100</td>
</tr>
<tr>
<td>0.92</td>
<td>4 (Larkhill Road, Benham Drive, Honeywood Close, Mark Close)</td>
</tr>
<tr>
<td>0.86</td>
<td>3 (Station Road, Stroudley Avenue, Walton Road)</td>
</tr>
<tr>
<td>0.83</td>
<td>1 (Green Farm Gardens)</td>
</tr>
<tr>
<td>0.8</td>
<td>1 (Ackworth Road)</td>
</tr>
<tr>
<td>0.77</td>
<td>1 (A27)</td>
</tr>
<tr>
<td>0.65</td>
<td>1 (M27)</td>
</tr>
</tbody>
</table>

**Table 1: Matched Roads**

Examples of matched roads are shown in the figures below. (OS ITN: red OSM: green)

![Figure 9: Marazan Road](image)
For the 5 missing roads, Table 2 summarizes relevant information and missing reasons below.

<table>
<thead>
<tr>
<th>Name in OS ITN</th>
<th>Name in OSM</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>A2030</td>
<td>Eastern Road</td>
<td>Name conflict</td>
</tr>
<tr>
<td>A288</td>
<td>Common Road</td>
<td>Name conflict</td>
</tr>
<tr>
<td>Kirtley Close</td>
<td>NULL</td>
<td>NULL in OSM</td>
</tr>
<tr>
<td>Breech Close</td>
<td>Breech Close</td>
<td>Neighbour conflict</td>
</tr>
<tr>
<td>Dundas Close</td>
<td>Dundas Close</td>
<td>Error in OSM attributes</td>
</tr>
</tbody>
</table>

Examine A2030 further, one can find it represented as the same line as Eastern Road, which has been matched, in OS ITN data set. Similar situation applies to A288. Kirtley Close is a named road in OS ITN, and fails to match to NULL in OSM. In Figure 12, the neighbour of Breech Close (a) in OS ITN is Green Farm Gardens (c), while its neighbour is Benham Drive (b), who has a neighbour Green Farm Gardens (c), in OSM. Because of different topological connection, Breech Close fails to match. This also suggests the necessity of geometric assessments for matched roads (e.g. Benham Drive).
Compared to pure geometric matching approach, this ontology based approach is more efficient. However, the effectiveness of this approach relies on the information completeness of input data sets. The methodology shows promising experimental results when applied to this case study, with more than 90% of roads matched in experiments.

CONCLUSION
This paper looks into developing techniques for geospatial data integration (GDI) using ontology based matching techniques. A prototype ontology matching technique has been developed to derive ontology based matching between an OS ITN dataset and an OSM road dataset. Compared to geometric matching, it requires less computation time, and seems more efficient and effective, especially when the completeness of data is high. Future work will concentrate on refining the methodology and assessing the matched data geometrically, providing a good foundation for geospatial data updates.

ACKNOWLEDGEMENTS
The authors express thanks for the Ordnance Survey, UK and OSM for the data used in this work. All figures in this text using OS data are ©Crown Copyright/database right 2010. An Ordnance Survey/EDINA supplied service.

REFERENCES


Goodchild, M. (2009), NeoGeography and the nature of geographic expertise. Journal of Location Based Services, 3 (2), pp. pages 82 - 96 Available online at:
http://www.informaworld.com/smpp/content~db=all~content=a911734343


Quantum GIS 2010, Welcome to the Quantum GIS Project [online] Available at:
www.qgis.org