TRIDEC and REVEAL projects

Stuart E. Middleton

University of Southampton IT Innovation Centre

sem@it-innovation.soton.ac.uk

www.it-innovation.soton.ac.uk

Overview

- Overview of project use cases [TRIDEC, REVEAL]
- Architecture and use of standards
- Geoparsing and geosemantic annotation
- Geosemantic knowledge model for trust and credibility analysis
- Conclusions

Overview of project use cases

- FP7 TRIDEC project [Sept 2010 - Oct 2013]

Tsunami Early Warning Centres

Natural Disasters

Tide gauges

Landslides

Flooding

In-Situ Sensors

Seismic sensors

Alerting

GPS sensors

Social Media

Twitter

Decision Support

Pattern Matching

Command and Control

Data Fusion

Geoparsing

Crisis Mapping

Modelling

Overview of project use cases

- FP7 TRIDEC project [Sept 2010 - Oct 2013]

Tsunami Early Warning Centres
Natural Disasters
Tide gauges
Landslides
Flooding
In-Situ Sensors
Seismic sensors
Alerting
GPS sensors
Social Media
Twitter
Decision Support
Pattern Matching
Command and Control
Data Fusion
Geoparsing
Crisis Mapping
Modelling

Overview of project use cases

- FP7 REVEAL project [Nov 2013 – Oct 2016]

Breaking News

News Events Journalists User forums

Geoparsing Social Media Twitter You Tube

Geo Semantics Four Square Instagram

Linked Data Signature detection Data Fusion Provenance

Crisis Mapping Natural Language Processing

Decision Support

Verification

Overview of project use cases

- FP7 REVEAL project [Nov 2014 – Oct 2017]

Breaking News
News Events
Journalists
User forums
Geoparsing
Social Media
Twitter
You Tube
Geo Semantics
Four Square
Instagram
Signature detection
Linked Data
Data Fusion
Provenance
Crisis Mapping
Natural Language Processing
Decision Support
Credibility
Trust
Verification

Architecture and use of standards

- W3C Standards
  - Sensor data ➔ Semantic Sensor Network Ontology (SSNO)
  - Communities ➔ Semantically-Interlinked Online Communities (SIOC)
  - Semantic Web, Linked Data ➔ RDF, OWL
- OGC Standards
  - Sensors ➔ SensorML, SOS, SPS
  - Geometry ➔ OpenGIS, GeoSPARQL
- Other
  - Open Street Map (OSM) ➔ OSM entities and tags including OpenGIS

Architecture and use of standards

- Scalable architecture for geosemantic processing
  - Storm framework – scalable clustered deployment of processing bolts
  - HTTP endpoints – control storm topologies in real-time
  - RabbitMQ – communication backbone
- Database layer
  - PostgreSQL + PostGIS database + planet OSM tables
  - OWLIM SE (under evaluation)

Geoparsing and geosemantic annotation

- Exploiting OSM entity model as a knowledge-base
  - planet_osm_line ➔ roads, rivers ...
  - planet_osm_point ➔ bus stops, small buildings, stations ...
  - planet_osm_polygon ➔ admin areas, large buildings, parks ...
  - All with OSM tags and OpenGIS geometry
  - But ...
    - 2+ trillion nodes impossible to hold in memory !!
    - Use spatial filtering based on dynamically declared focus areas
    - Compile inverted indexes and use information retrieval (IR) techniques

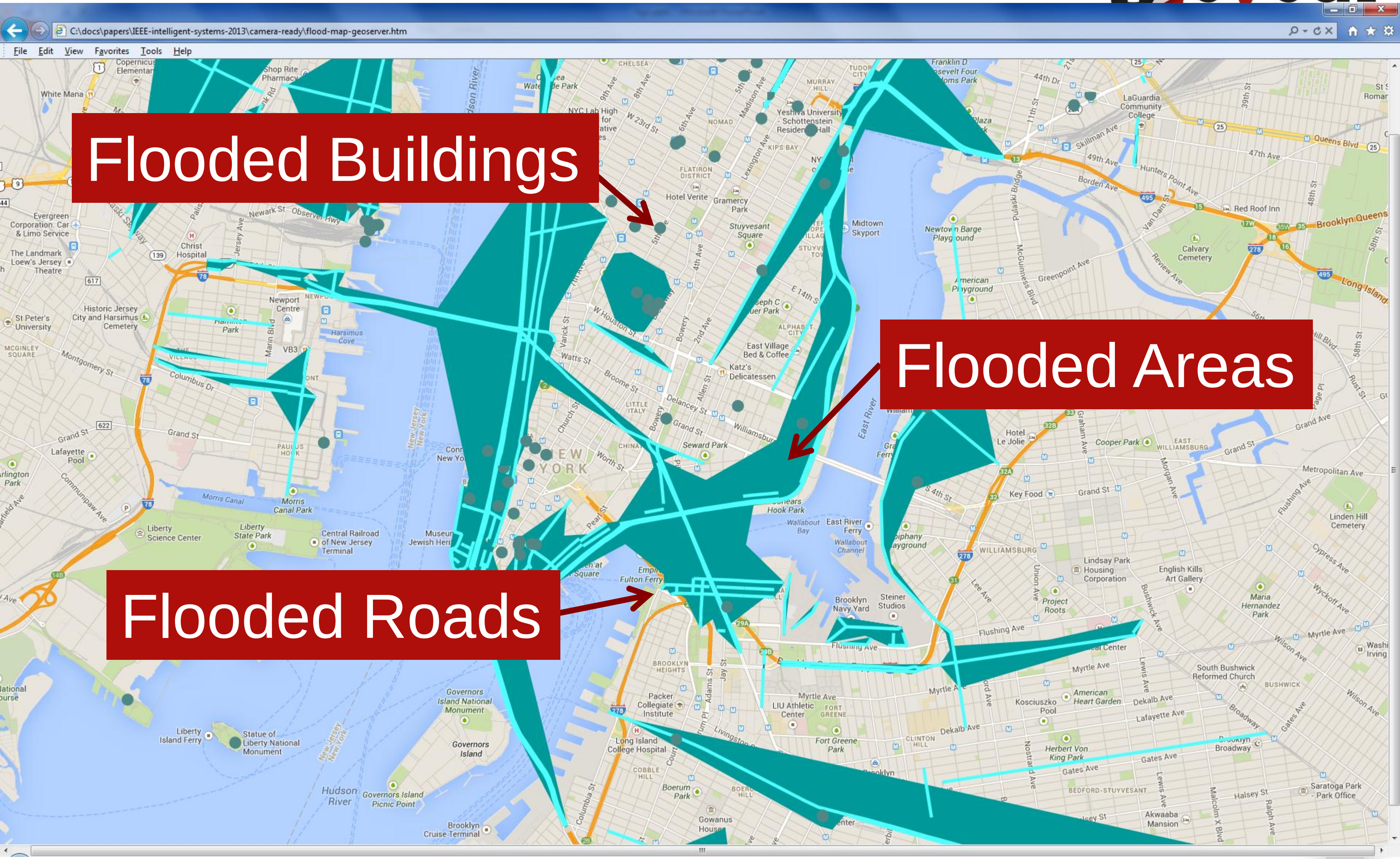
Geoparsing and geosemantic annotation

- Real-time geoparsing of social media streams
  - PostGIS spatial filtered lookups for focus area (e.g. a city boundary)
  - Compile an in-memory cache of location data
  - Loop in real-time
    - Match location name tokens to social media JSON content stream (e.g. JSON tweet)
    - Add location annotations to JSON content
    - Send annotated JSON content to aggregator and situation assessment picture building
  - Map reduce type architecture for distributed geoparsing

Geoparsing and geosemantic annotation

- Exploiting linked open geospatial data
  - OSM location matches ➔ OSM ID set ➔ linkedgeodata.org URI's
  - linkedgeodata.org URI ➔ linked dbpedia.org URI ➔ Semantic context
  - Semantic context ➔ Reports for decision support visualization
- GeoServer WMS visualization

New York Hurricane Sandy 2012 Storm Surge Flooding



Middleton, S. Middleton, L. Modafferi, S. (2014) Real-time Crisis Mapping of Natural Disasters using Social Media. IEEE Intelligent Systems, 02 Jan. 2014. IEEE computer Society Digital Library. IEEE Computer Society

Geosemantic knowledge model for trust and credibility analysis

- Evidential approach to situation assessment and trust modelling
- Geospatial knowledge representation
  - In real-time as situation develops incrementally build knowledge-base
  - Assert JSON location annotations as OWL individuals + OSM tag metadata as OWL properties
  - Subclass GeoSPARQL geo:Feature to encode geometry
  - Results include linkedgeodata.org URI's for OSM locations so we can later add Dbpedia linked data entries for end user reports

Geosemantic knowledge model for trust and credibility analysis

- Inference model for trust and credibility analysis
  - Using image features and location OSM features
    - Instagram (text) ➔ Geoparse ➔ owl_loc + owl_feature_props
    - Instagram (image) ➔ Image classifier ➔ owl_feature_props
    - features(text) == features(image) ➔ loc_credibility++
  - Using location geography
    - Tweet (text) ➔ Geoparse ➔ owl_loc + GeoSPARQL_geom
    - Instagram (text) ➔ Geoparse ➔ owl_loc + GeoSPARQL_geom
    - owl_loc ST_Dwithin(500m) owl_loc ➔ loc credibility++

Geosemantic knowledge model for trust and credibility analysis

- Inference model for trust and credibility analysis
  - Using trusted lists of users
    - Tweet (text) ➔ Geoparse ➔ owl_loc + owl_author_prop
    - Trusted list (end user) ➔ owl_trusted_user
    - owl_loc_author IN [owl_trusted_user] ➔ trusted++
  - (Geo)SPARQL queries to filter relevant / credible / trusted results
    - SPARQL to classify owl_loc individuals that match known facts about the breaking news story
    - GeoSPARQL to classify owl_loc individuals based on OpenGIS geometry [nearby, connected_to, within_admin_area]

Geosemantic knowledge model for trust and credibility analysis

- Experiences so far
  - Storm supports scalable incremental processing really well
  - Spatial filtering is key when dealing with Planet OSM database
  - owl:restriction’s allows incremental inference via 'class-individual mirror' inference pattern
    - e.g. evidence arrives, end user axioms asserted such as trusted users
- Open questions regarding geosemantic inference
  - Which triple store is best for GeoSPARQL? OWLIM-SE?
  - We have access to geometry in PostGIS and OWLIM-SE. Trade off between efficiency of SQL and inference of GeoSPARQL?

Conclusions

- Future work
  - 1st year prototype storm deployment due Nov 2014
    - 4 social media sources
    - Geoparsing locations in parallel on a storm cluster
    - Real-time situation assessment map visualization
    - Incremental geosemantic reasoning
  - Scalability testing – cluster deployment testing on location data from several complete cities
  - Scientific evaluation - we will gather ground truth from journalists in a real news room for several target breaking news stories in 2015

Many thanks for your attention!

Any questions?

Stuart E. Middleton

University of Southampton IT Innovation Centre

email: sem@it-innovation.soton.ac.uk

web: www.it-innovation.soton.ac.uk

twitter:@stuart_e_middle

Project partners & where to find us

INTRASOFT INTERNATIONAL

Alcatel·Lucent

software AG

ATC ATHENS TECHNOLOGY CENTER

WeST People and Knowledge Networks

INFORMATICS & TELEMATICS INSTITUTE

iT innovation

ΔΗΜΟΚΡΙΤΟΣ

SINTEF

DW Deutsche Welle

icri Interdisciplinary Centre for Law & ICT

SEVENTH FRAMEWORK PROGRAMME

Visit our website!

"The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement #610928. The positions expressed are those of the authors and do not necessarily reflect the views of the European Commission"

Axillary slide: Class-individual mirror inference pattern

```
<itinno:River> rdfs:subClassOf <geo:Feature>
<ns:river1> rdf:type <itinno:River>
<ns:river1> itinno:name "River Itchen"
<ns:river1> geo:hasGeometry <ns:geom4>
<ns:geom4> rdf:type <geo:Geometry>
<ns:geom4> geo:asWKT "POLYGON((...))"^^<http://www.opengis.net/def/sf/wktLiteral>
<itinno:NearbyRiverItchen> owl:equivalentClass [ a owl:Restriction;
  owl:onProperty itinno:nearby;
  owl:hasValue <river1>]
SELECT ?loc
WHERE {
  ?loc rdf:type <geo:Feature> .
  ?loc rdf:type <itinno:NearbyRiverItchen> .
  ?loc geo:hasGeometry ?g .
  ?g geo:asWKT ?wkt .
  FILTER( geof:sfContains( ?wkt, "POLYGON((...))"^^<http://www.opengis.net/def/sf/wktLiteral> ) )
}
```