REVEAL

FP7-610928

REVEALing hidden concepts in Social Media

Deliverable D5.3.2

Metadata driven data fusion framework

Editor(s):	Stuart E. Middleton
Responsible Partner:	University of Southampton IT Innovation Centre
Status-Version:	v1.2
Date:	28/04/2017
EC Distribution:	Public (PU)

Project Number:	FP7-610928
Project Title:	REVEAL

Title of Deliverable:	Metadata driven data fusion framework
Date of Delivery to the EC:	30/06/2015

Workpackage responsible for the Deliverable:	WP5 - Modalities Analysis Framework
Editor(s):	Stuart E. Middleton (ITINNO)
Contributor(s):	ITINNO
Reviewer(s):	Symeon Papadopoulos (CERTH)
Approved by:	All Partners

Abstract:	The metadata driven data fusion framework is collectively grouped into a component called the REVEAL situation assessment framework. The situation assessment framework incrementally aggregates real-time content from REVEAL social media crawlers and subsequent annotations from processing components in WP2/3/4. A Storm controller provides an HTTP endpoint to start and stop new situation assessments which is driven by the pilot UI's. Real-time JSON formatted social media content is streamed from crawlers via a RabbitMQ message bus with WP2/3/4/5 components adding asynchronous annotations as their processing results become available. A situation assessment Storm topology aggregates all JSON annotations and populates a set of database tables which can be later visualized using the tools from D5.4.2. We have evaluated both the throughput of data import and query speeds for typical queries we expect to run via our visualizations. The data rates are above the expected peak data throughput for our use cases and the query times are well within the speed needed for interactive queries. We also report results comparing the REVEAL spatial location analysis approaches. A software release accompanies this deliverable, installed and running on the WP6 project testbed.
Keyword List:	Situation Assessment, Aggregation, Data Fusion

DOCUMENT DESCRIPTION

Document Revision History

Version	*Date*	*Modifications Introduced*	
		Modification Reason	*Modified by*
v0.1	16/06/2016	Setting up document, initial draft	ITINNO
v1.0	17/06/2016	Release candidate for internal QA	ITINNO
v1.1	23/06/2016	Version ready for coordinator QA	CERTH, ITINNO
v1.2	28/04/2017	Public version	ITINNO

CONTENTS

LIST OF FIGURES

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Acronym	Title
DSS	Decision Support System
GeoSPARQL	Geographic Query Language for RDF Data
HCI	Human Computer Interaction
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ITINNO	University of Southampton IT Innovation Centre
JDL	Joint Defence Laboratories
JSON	JavaScript Object Notation
PM	Person Month
SPARQL	SPARQL Protocol and RDF Query Language
URI	Uniform Resource Identifier
WP	Work Package
CERTH	The Centre for Research and Technology Hellas

1 Introduction

The metadata driven data fusion framework (i.e. software from D5.3.2) is collectively grouped into a component called the REVEAL situation assessment framework. The scope of this component is to incrementally aggregate real-time content from REVEAL social media crawlers and annotations from components in WP2/3/4/5. WP5 metadata templates for all annotations are defined in D5.1, and continually updated in an internal WP5 data dictionary document as part of our WP5 agile approach to development.

The concept of a situation assessment is taken from the de-facto standard JDL data fusion approach [Lambert 2009], and in our case represents an aggregated collection of content filtered and annotated in the context of a specific news story or enterprise event. A number of parallel situation assessments can be launched since news and events do not typically happen in a nice sequential order. For each parallel situation assessment a set of Postgres[1] and PostGIS[2] tables are created for content items and person profiles. A number of additional tables are also created and indexed for the annotations on these content items and person profiles, such as locations, tags, URIs and person groups.

It should be noted that batch-style annotations (e.g. offline content indexing) are recorded by the WP6 infrastructure in a MongoDB database. The WP5 data model is based on a real-time stream of content and annotations of this content.

In order to provide a scalable solution we have implemented the situation assessment framework as an APACHE Storm[3] process. Each Storm process receives JSON annotated social media content from WP2/3/4/5 via a RabbitMQ[4] message bus, which is well known as a high performance messaging layer. We expect WP2/3/4 annotations to arrive asynchronously and therefore adopt an incremental data fusion approach, adding annotations to the content as they arrive.

This report outlines the overall design of the WP5 software and the situation assessment framework in particular. We evaluate its performance in terms of throughput and query speeds, as well as comparing some of the spatial analysis components from WP3 and WP4. The development work on the situation assessment framework component is a continuous process. This prototype deliverable outlines the progress so far within task 5.2 and represents a PM32 'snapshot' in this WP5 agile development process.

A software release accompanies this deliverable, installed and running on the WP6 project testbed.

[1] http://www.postgresql.org/

[2] http://postgis.net/

[3] http://storm.apache.org/

[4] http://www.rabbitmq.com/

2 Situation Assessment Framework

2.1 Situation Assessment Framework Design

The situation assessment framework is designed to incrementally aggregate JSON content received asynchronously from a RabbitMQ message bus. A high-level view of the information flow is shown in Figure 1.

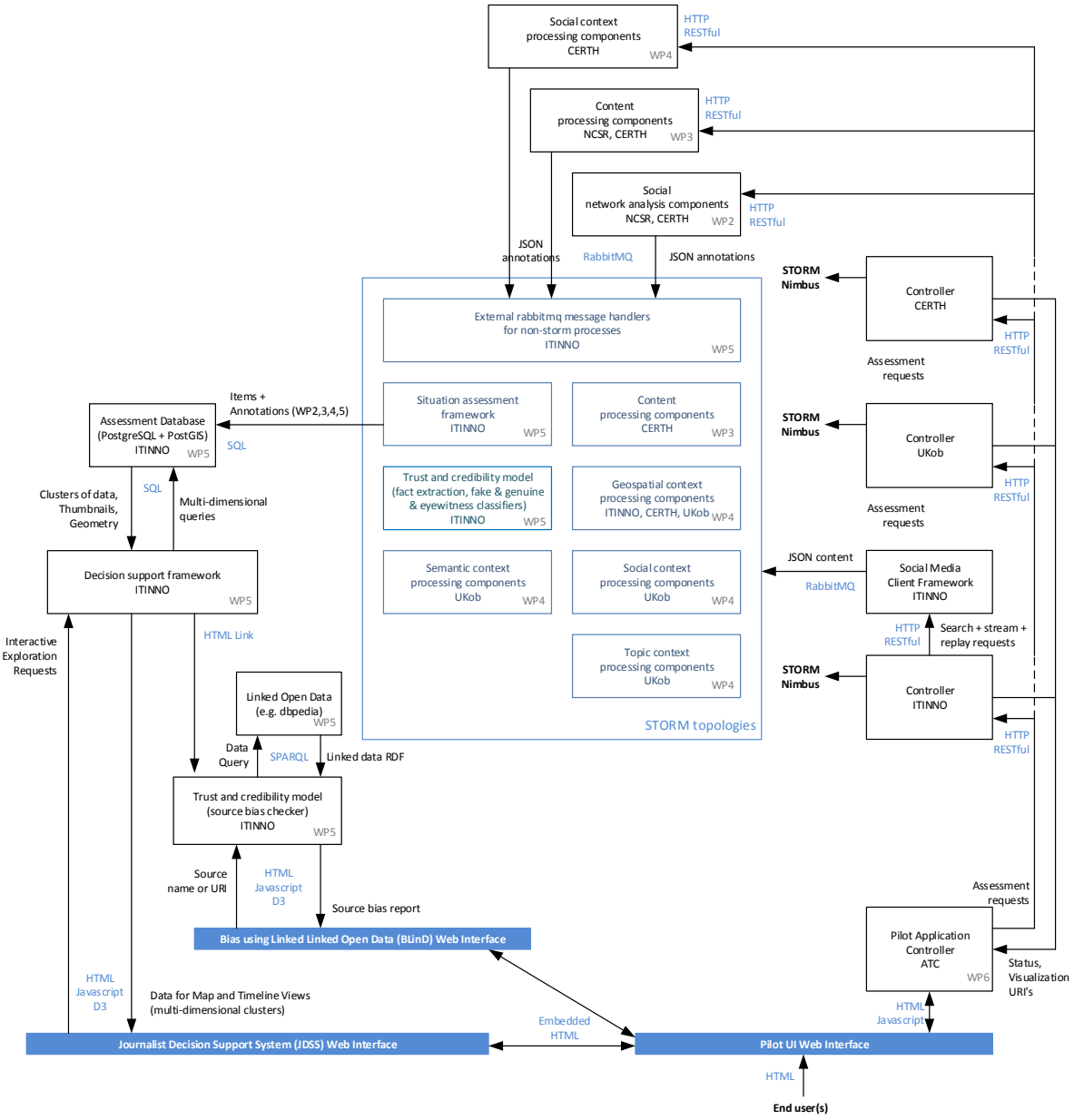

Figure 1: Information Flow for the Situation Assessment Framework

The WP5 situation assessment framework consists of (a) an ITINNO Storm controller and (b) a Python situation assessment process implemented as a Storm topology. The ITINNO Storm controller provides an HTTP interface that allows new situation assessments to be created whenever a news story or enterprise event occurs. As such there will be a unique situation assessment Storm

topology for each situation assessment. We have tested running several assessments in parallel; the memory footprint is below 64 Gbytes of RAM for 10's of assessments and the limit is more the number of CPU cores available to do the processing work (which is overcome by simply adding resources to the cluster).

The HTTP interface for the ITINNO Storm controller is shown in Table 1. The Pilot UI's will use this interface to start a new situation assessment, provide crawler keywords for it, provide focus areas and finally delete it when its work is finished.

Table 1: HTTP situation assessment framework interface

HTTP Action	Namespace	Request	Response
GET	/itinno-controller/assessment	None	JSON with list of all active assessments
GET	/itinno-controller/assessment/<assessment-id>	None	JSON containing detailed information about specified (i.e. /<assessment-id>) assessment
POST	/itinno-controller/assessment/<assessment-id>	JSON with assessment configuration (start/stop new assessment request), new focus area (add focus area request) or social media search/stream/replay request (add/delete social media request)	None
DELETE	/itinno-controller/assessment/<assessment-id>	None	None

The situation assessment ID is used as a well-known unique identifier for the (a) storm topologies associated with the situation, (b) RabbitMQ exchange names for a dedicated situation communication channel, (c) DSS web pages URI's and (d) database table names for aggregated content. This can be seen in Table 2. Use of a naming convention means components do not need to notify each other about where to locate resources, simplifying the control flow exchanges.

Table 2: Resource naming for situation assessments

Resource	Namespace
Storm Topologies	<assessment_id>_<topology_name> e.g. ukraine_2015_situation_assessment_aggregator
RabbitMQ exchange	<assessment_id>_<exchange_name> e.g. ukraine_2015_raw_json_exchange
DSS webpages	http://reveal-eu/dss/<assessment-id>/<view_type>.html where <view_type> = map | timeline e.g. http://reveal-eu/dss/ukraine_2015/map.html
Database tables	<assessment-id>_db_items <assessment-id>_db_people <assessment-id>_db_<annotation_type> <assessment-id>_db_<annotation_type>_index e.g. ukraine_2015_db_loc_index

The data fusion algorithm inside the situation assessment Storm topology follows a number of principles to ensure scalability and reduce the amount of data that must be transferred across the network. These principles are important to allow real-time aggregation able to handle the throughputs expected from the social media crawlers.

Typical throughputs [Middleton 2014] [Middleton 2016] are in the region of 18,000 content items per hour for a keyword filtered crawl using the Twitter streaming API. In REVEAL a typical scenario might be handling four keyword filtered streaming API crawls - making 4*18000 = 72,000 content items per hour to aggregate (i.e. 20 item per second). When using the Twitter search API to capture unsampled tweet sets (i.e. not the sampled Twitter streaming API) we see a typical result [Wiegand 2016] of 60,000 content items for each hour of data retrieved (i.e. 16 items per second). The search API is rate limited so this throughput is close to the maximum achievable without a commercial account with a Twitter data provider (current Twitter rate limits are 180 queries per 15 minutes with 100 results per query = 20 items per second).

The first design principle is to maximize the use of relational database indexing and the associated relational database query caching. We do this by using relational PostgreSQL and PostGIS and not triple-store based GeoSPARQL for any large scale geometry processing since (from our own analysis) PostgreSQL and PostGIS SQL OpenGIS functions are up to 10 times quicker that uSeekM GeoSPARQL operators. We do a lot of pre-processing of geometry information (e.g. super region calculations using ST_Contains, ST_Intersects, ST_Distance etc.) when focus areas are added in SQL. This avoids large scale geometry calculations on the fly (e.g. GeoSPARQL or in-memory shape analysis using Python libraries such as Shapely). It should be noted that recent MongoDB solutions have integrated PostGIS but we need the relational database technology for interactive query performance so opted for PostgreSQL.

The second design principle is to publish raw content on one RabbitMQ exchange only. All WP2/3/4 annotations subscribe to this raw JSON content exchange and then publish to separate annotation exchanges a JSON message with the newly created annotations and a simple reference back to the

original JSON message. This avoids sending the original 5-10 Kbytes of raw JSON each time a new annotation is added. The situation assessment Storm topology re-assembles these JSON annotations with the original raw JSON during content aggregation. This significantly reduces the RabbitMQ traffic and the RabbitMQ queue sizes associated with each RabbitMQ exchange.

The third design principle is to use a single local database for storing aggregated datasets, eliminating any need for remote database access and cross-database lookup. We do not distribute the final aggregated database and instead host it on the node running the decision support system framework to optimize query speed. The decision support system framework client will need to execute remote queries but is designed to ensure all statistical & clustering work is performed in SQL (i.e. server side) to avoid large data transfers. This is important as large data transfers would be too slow for interactive use where response times of < 1 second are needed.

The final design principle is to support incremental updates and parallel situation assessments. Incremental updates are achieved using SQL update statements where conditional updates are performed within a single commit. Relational database locking ensures the aggregated data remains consistent (e.g. index tables). To support parallel situation assessments we create a separate SQL table set for each assessment and rely on the relational database to handle multi-client access.

2.2 Database Schema

The situation assessment database has a number of tables for each requested situation assessment. There is item table, a set of annotation tables and a set of index tables. We expect at least 15 annotation tables for many annotations resulting from parsing JSON metadata (e.g. hashtags, mentioned URI's) and WP2/3/4/5 processing work (e.g. geoparsed locations, damage reports, attributed entities etc.).

The item table has a row for each social media post received and contains columns for attributes such as post timestamp, text and author. There is an annotation table for every annotation type and is the principle way of storing content annotations from WP2/3/4/5 (e.g. hashtags, user mentions, URI mentions, attributed entities, damage report facts, topic etc.). Finally there is an index table for every annotation table, and it always has a two column structure for foreign keys linking the item table to the annotation table.

The use of relational index tables allows very fast querying of the situation assessment datasets and statistics / clusters to be computed in real-time. This fast query capability is important as the visualization in D5.4.2 is interactive and long delays on database queries would be unacceptable with regards to interface performance.

The table specifications are outlined below in Figure 2. We have developed a data layer which is driven by a table specification configuration, allowing new annotation types to be added with only changes to the configuration needed (i.e. no need to recompile the code). This is an important feature to ensure integration with the many WP2/3/4/5 annotations does not take too much effort.

Items table schema

```
CREATE TABLE reveal.paris_2015_db_item
(
  item_key serial NOT NULL,
  source_uri text,
  created_at timestamp with time zone,
  text text DEFAULT ''::text,
  lang text DEFAULT 'en'::text,
  geotag geometry(Geometry,4326) DEFAULT NULL::geometry,
  eyewitness_class text,
  fake_class text,
  original boolean DEFAULT false,
  updated_time timestamp with time zone DEFAULT now(),
  CONSTRAINT paris_2015_db_item_pkey PRIMARY KEY (item_key),
  CONSTRAINT paris_2015_db_item_source_uri_key UNIQUE (source_uri)
)
WITH (
  OIDS=FALSE
);

CREATE INDEX paris_2015_db_item_gist_index
  ON reveal.paris_2015_db_item
  USING gist
  (geotag);
```

Annotation table schema example : Damage report

```
CREATE TABLE reveal.paris_2015_db_damage
(
  damage_key serial NOT NULL,
  damage text,
  CONSTRAINT paris_2015_db_damage_pkey PRIMARY KEY (damage_key),
  CONSTRAINT paris_2015_db_damage_damage_key UNIQUE (damage)
)
WITH (
  OIDS=FALSE
);
```

Index table schema example : Damage report index

```
CREATE TABLE reveal.paris_2015_db_damage_index
(
  item_key bigint NOT NULL,
  damage_key bigint NOT NULL,
  CONSTRAINT paris_2015_db_damage_index_damage_key_fkey FOREIGN KEY (damage_key)
      REFERENCES reveal.paris_2015_db_damage (damage_key) MATCH SIMPLE
      ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paris_2015_db_damage_index_item_key_fkey FOREIGN KEY (item_key)
      REFERENCES reveal.paris_2015_db_item (item_key) MATCH SIMPLE
      ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT paris_2015_db_damage_index_item_key_damage_key_key UNIQUE (item_key, damage_key)
)
WITH (
  OIDS=FALSE
);
```

Annotation table schema example : Location

```
CREATE TABLE reveal.paris_2015_db_loc
(
  loc_key serial NOT NULL,
  osm_id bigint[],
  osm_tag hstore,
  parent_osm_id bigint[],
  shape geometry(Geometry,4326),
  multi_name text[],
  osm_uri text,
  loc_name text,
  CONSTRAINT paris_2015_db_loc_pkey PRIMARY KEY (loc_key)
)
```

```
WITH (
  OIDS=FALSE
);

CREATE INDEX paris_2015_db_loc_gist_index
  ON reveal.paris_2015_db_loc
  USING gist
  (shape);

Annotation table schema example : Hashtag

CREATE TABLE reveal.paris_2015_db_tag
(
  tag_key serial NOT NULL,
  tag text,
  CONSTRAINT paris_2015_db_tag_pkey PRIMARY KEY (tag_key),
  CONSTRAINT paris_2015_db_tag_tag_key UNIQUE (tag)
)
WITH (
  OIDS=FALSE
);
```

Figure 2: Database schema for situation assessment framework

3 Evaluation

We have performed two evaluations within the context of D5.3.2. The first looks at comparing geoparsing techniques for spatial location annotations which is important for the WP5 map view. The second looks at the overall performance in terms of aggregation throughput.

3.1 Situation assessment framework performance

We performed an evaluation on the situation assessment framework software to measure its real-time data import and interactive query performance. This is important to know as we expect to work in real-time. Our peak throughput target is 16 items per second based on the analysis from section 2.1.

The test reported in this section was performed using a dataset crawled during the first 6 hours of the Paris shootings November 2015 containing content items from Twitter (2,000,000+), Facebook (100+), You Tube (1000+) and Instagram (200,000+). The Facebook post numbers are small as we only crawled a couple of news sites for a proof of concept for Facebook in the Paris dataset. A single 2GHz CPU core was used and there was 64 GBytes RAM available (although a small fraction of this was needed). Using a single CPU core represents a 'worst case' setup as we usually run on one or more multi-core servers; we wanted in this section to show that the worst case was able to handle our target throughput easily.

These content item sizes are well above what we would expect for a breaking news story. The first 1 hour of the Paris dataset filtered by English and French is about 63,000 content items, so this 2,000,000+ dataset is about 31 times larger than we would usually expect to process. The test results are shown in Table 3 and Table 4 and represent a worst case performance level to allow us assess the scalability of the framework.

The query types are representative of the type of multi-dimensional queries that the WP5 visualization generates. Users can select a choice of 1 or 2 dimensions (e.g. location, damage report, hashtags etc.) and will display either the top N items ranked my item mention count or the first N items ordered by timestamp with 0,1 or 2 dimensional values specified (e.g. first 10 items containing a hashtag value of *'#stadedefrance'*).

Table 3: Data import throughput for 32k item chunks. Throughput varies by social media type mostly due to post text and metadata size. Throughput did not decrease at all as prior data volume increased (i.e. excellent import scalability).

Social media type	Mean Import Throughput for 32k Item Chunks [stdev for data if multiple 32k chunks were imported] 2M dataset
Twitter posts	43 items / sec [stdev 3.43]
Facebook posts	46 items / sec [stdev n/a]
You Tube posts	31 items / sec [stdev n/a]
Instagram posts	23 items / sec [stdev 2.37]
Target throughput	16 items / sec

Table 4: Data query speed (worst case)

Query type	Result type	Query speed 2M dataset
Temporal view get sample count, items sampled by 1 hour	List of all time samples and frequency counts	200ms
Temporal view get top 5 authors for a specific 1 hour sample	List of author names	720ms
Temporal view get top 5 items (earliest first), location 'Stade de France', hashtag '#stadedefrance'	List of source URIs	30ms
Map view get location cluster count	List of all locations and frequency counts	1,000ms
Map view get top 10 damage reports , location 'Stade de France'	List of damage report text	33ms
Map view get top 5 items (earliest first), location 'Stade de France', hashtag '#stadedefrance'	List of source URIs	30ms
Target query speed	Any	1,000ms

We have found that the query response times are very fast and suitable for supporting an interactive UI. The response times are limited mostly by the network I/O delays sending data for the request and response as opposed to being limited by the database query planning and execution. This is expected as we have made heavy use of SQL indexing and structured the tables to allow maximum use of the relational capabilities of PostgreSQL and PostGIS.

We have implemented a geometry object cache in our decision support system to avoid the need for interactive queries containing large text encoded geometry objects. In this way geometry objects are rendered quickly (e.g. for the map view) from our cache at interactive speeds. Loading large geometry objects (e.g. the outline of Russian) on the fly is not possible at interactive speeds with the hardware we are using.

3.2 Spatial placing accuracy evaluation

Location annotation is important to WP5, and the map view in particular, so we performed a benchmark comparison of these complementary location annotation approaches to understand the strengths and weaknesses of each.

Within REVEAL we have developed two very different approaches to geoparsing and extracting spatial locations from posts. The first (from ITINNO) is described in D4.1 [Middleton 2014] [Middleton 2016] and is based on named entity matching to location tokens taken from a planet deployment of OpenStreetMap. This approach tries to associate a disambiguated OpenStreetMap location to mentions of locations in post text. The second (from CERTH) is described in D4.3 and is based on a supervised learning approach using tags from geotagged Flickr posts in the Yahoo Flickr CC 100M dataset. This approach tries to associate a likely bounding box, also referred to as the most likely cell, of configurable resolution to mentions of specific tags and phrases.

We used the MediaEval 2015 Placing task dataset (which is a subset of the Yahoo Flickr CC 100M dataset) as a benchmark. The tag-based supervised learning approach has already been applied to this dataset and the early results are published in [Kordopatis-Zilos 2015a] [Kordopatis-Zilos 2015b]. This datasets consists of a training set (4.7M posts) and a testset (950k posts). The idea of the placing task is to estimate a (long,lat) coordinate for each post and then calculate its distance from the ground truth Flickr post geotag. The results are broken down into several distance ranges (1km, 10km, 100km) and an overall mean distance error.

Since the entity matching OpenStreetMap approach returns the full geometry for a location (i.e. not a simple point) we used the OpenStreetMap admin centre node (if available) or polygon centroid as a point result.

Results

Table 5: Geoparse comparison results

	Recall	Precision @ 0.1km	Precision @ 1km	Precision @ 10km	Mean Distance Error
CERTH high confidence	0.28	0.15	0.54	0.88	0.8 km
CERTH any confidence	1.00	0.06	0.24	0.43	69 km
ITINNO high confidence	0.25	0.01	0.22	0.63	4 km
ITINNO any confidence	0.68	0.01	0.12	0.35	172 km

From the results in Table 5 it is clear that the tag-based supervised learning approach is the strongest on the benchmark dataset. It can pick up from the training data much more than the location names alone, and is able to make use of statistical patterns around landmarks, event names, colloquial terminology etc. This is mostly a strength and produces good results but can also be a weakness as discussed later.

Some sources of error for the entity matching OpenStreetMap approach are (a) the calculation of the spatial point (long,lat) for each location and (b) the limitation of only working at a regional level of granularity. Unlike the tag-based approach, which is able to learn the exact points people are taking Flickr images from, the OpenStreetMap approach uses a set of admin region polygons and thus must guess a point based on the admin centre or centroid of the region being specified. The entity matching OpenStreetMap approach also can only load regional data (due to memory constraints) as the benchmark dataset is global in nature so does not have access to any street or building information. This makes precision below 10km difficult.

It should be noted that in many REVEAL use cases the focus areas are known in advance (e.g. location of a breaking news story) and are not anywhere on the planet. This means we can load focus areas from OpenStreetMap (e.g. a whole city) into memory and geoparse regions, streets and buildings. However this was not possible for this global-scope benchmark evaluation. The global regions dataset of about 900,000 locations takes about 12 Gbytes of RAM to load into memory but only has to be done once.

Another point to note is that supervised learning approaches will probably work well only for locations where there is training data available. For example if a news story breaks in an area of the world where there are no Flickr image posts then it is unlikely that any classification will be able to be made. The OpenStreetMap database has a clear advantage here, since it has detailed information on every location on the planet (although coverage of third world countries or countries that strictly control mapping data can be sparse).

These results represent our analysis to date. We intend to run further tests, looking at different segments of the MediaEval dataset and assessing the strengths and weaknesses of each approach on these and also any bias that exists within the dataset itself (e.g. to popular landmark locations that tourists visit). We will also benchmark our approaches on the ITINNO open geoparse dataset[5] which contains for several news events labelled location data at the region, street and building levels.

[5] http://web-001.ecs.soton.ac.uk/wo/dataset#566800b4fe0bc6f34a92e203

4 Modality Innovation Description

Below is a module innovation description for components referred to in this deliverable.

Module Name	Situation Assessment Framework	**Delivery date**	PM32
Module Overview			
The situation assessment framework incrementally aggregates real-time content from REVEAL social media crawlers and annotations from components in WP2/3/4. It implements a scalable incremental aggregation approach using relational database technology.			
Based on existing work? (e.g. from other project or open source code)			
Based on data fusion know-how from the FP7 TRIDEC and ENVIROFI projects.			
Based on implementation of specific algorithms? (which? why?)			
N/A			
Innovation introduced			
The situation assessment framework provides efficient real-time aggregation and cross-indexing of large volumes (1,000,000+ items) of social media content. The innovation it provides is supporting the spatial-temporal-semantic grounding of real-time evidence with relation to end user verification tasks.			
Is this considered a core innovation for the project? Why?			
No - This component supports many components but is not a key innovation in itself.			
What benchmarks will be used to evaluate the module performance?			
Throughput (target 16 items per second)			
Partners Involved and related WP/Task(s)			
ITINNO (T5.2 lead - development)			

5 Conclusions

This deliverable describes the metadata driven data fusion framework (i.e. software from D5.3.2) which is collectively grouped into a component called the REVEAL situation assessment framework.

The situation assessment framework incrementally aggregates real-time content from REVEAL social media crawlers and subsequent annotations from processing components in WP2/3/4. An ITINNO Storm controller provides an HTTP endpoint to start and stop new situation assessments which is driven by the pilot UI's. Real-time JSON formatted social media content is streamed from crawlers via a RabbitMQ message bus with WP2/3/4 components adding asynchronous annotations as their processing results become available. A situation assessment Storm topology aggregates all JSON annotations and populates a set of database tables which can be visualized using the tools from D5.4.2.

We have evaluated both the throughput of data import and query speeds for typical queries we expect to run via out visualizations. The data rates are above the expected peak data throughput for our use cases and the query times are well within the speed needed for interactive queries. We also report results comparing the REVEAL spatial location analysis approaches.

A software release accompanies this deliverable, installed and running on the WP6 project testbed. The work in WP5 follows a continuous agile development process with feedback from end users and as such this prototype represents a PM32 snapshot in a continuous development process.

6 References

Kordopatis-Zilos, G., Papadopoulos, S., & Kompatsiaris, Y. (2015). Geotagging Social Media Content with a Refined Language Modelling Approach. In Intelligence and Security Informatics (pp. 21-40). Ho Chi Minh City, Vietnam: Springer International Publishing.

Kordopatis-Zilos, G., Popescu, A., Papadopoulos, S., & Kompatsiaris, Y. (2015). CERTH/CEA LIST at MediaEval Placing Task 2015. In Proceedings of the MediaEval 2015 Workshop. Wurzen, Germany.

Lambert, D. A.: A blueprint for higher-level fusion systems. Information Fusion, 10 (1), 6-24 (2009)

Middleton, S.E. Krivcovs, V. "Geoparsing and Geosemantics for Social Media: Spatio-Temporal Grounding of Content Propagating Rumours to support Trust and Veracity Analysis during Breaking News", ACM Transactions on Information Systems (TOIS), 34, 3, Article 16 (April 2016), 26 pages. DOI=10.1145/2842604 (2016)

Middleton, S.E. Middleton, L. Modafferi, S. "Real-Time Crisis Mapping of Natural Disasters Using Social Media", Intelligent Systems, IEEE, vol.29, no.2, 9-17, (2014)

Wiegand, S. Middleton, S.E. “Veracity and velocity of social media content during breaking news: Analysis of november 2015 paris shootings”, Third Workshop on Social News On the Web (SNOW-2016), WWW 2016 Companion of the 25th International World Wide Web Conference (2016)