

# Translating Maps and Coordinates from the Great War

Rob Warren<sup>1</sup> and David Evans<sup>2</sup>

<sup>1</sup> Big Data Institute  
Dalhousie University  
Halifax, Canada  
[rhwarren@dal.ca](mailto:rhwarren@dal.ca)

<sup>2</sup> Department of Computing and Mathematics  
University of Derby  
Derby, UK  
[d.f.evans@derby.ac.uk](mailto:d.f.evans@derby.ac.uk)

**Abstract.** Many obsolete coordinate systems used in the past have fallen into disuse. However, the contents of historical documents still refer to these obsolete coordinates and thus translation systems are important in locating historical events. We present a specialized Linked Open Data API constructed to translate obsolete British Trench Map coordinates from the Great War into modern WGS84 coordinates. We report on the design of the API, the construction of the triple structures used to answer queries and the methods used to enrich query results while ensuring network performance. Several use cases for the API are presented and we close with a discussion of our experiences in linking to external data providers.

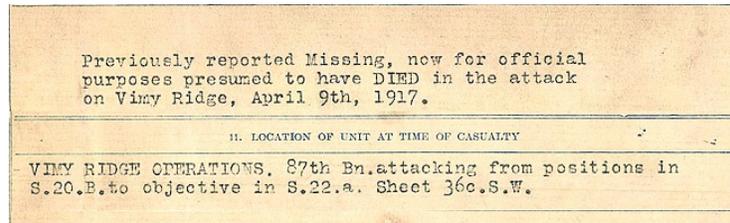
**Keywords:** Linked Geo Data, Coordinates Translation, Great War

## 1 Introduction

With the centenary of the Great War renewed interest has been shown in the archival records of this conflict. British and Commonwealth forces used a special coordinate system known as the British Trench Map Coordinate system, which was invented to support military operations on the Western Front.

An example in Figure 1a is an extract from the Circumstances Of Death register of Private John Richard Aaron who went missing during the Battle of Vimy Ridge in 1917. The document records the starting point 36C.S.20.b (the left box in Figure 1b) of the attack and the location of the intended objective at 36C.S.22.a (right box in Figure 1b). His body was never recovered and his remains are likely still there today.

For a number of years translating these coordinates into a modern location could only be done by locating a physical copy of trenchmap sheet 36C, re-projecting the map based on known landmarks and only then could the specific squares S.20.b and S.22.a be geo-located.



(a) Circumstances of death of John Aaron



(b) Jumping off position at Squares 36C.S.20.b (left) and objective square 36C.S.22.a (right).

Fig. 1: Linking locations as recorded in archival material to their current locations.

In the context of a Linked Open Data project on the Great War called Muninn<sup>3</sup> it became necessary to be able to translate these coordinates back and forth on a large scale. This paper reports on the design of a Linked Open Data API<sup>4</sup> capable of translating British Trench Map coordinates of the Great War to and from modern WGS84 coordinates.

This paper is organized as follows: we begin with a short introduction to Trench Map Coordinates and the work done to rebuild the mapping system of the time. A brief related work section is then followed by a description of the API functions, the underlying mapping ontology and the data enrichment strategies used by the API. We report on some experimental results obtained in the construction and operation of the API and close with some opportunities that were identified during the course of this work.

## 2 Problem definition

With the invasion of Belgium by the Germans in 1914, the official Belgian printing plates for the base country maps were evacuated to England where the Ordnance Survey used them as the basis for a new series of small scale maps.

<sup>3</sup> <http://www.muninn-project.org/>

<sup>4</sup> The actual API is at <http://rdf.muninn-project.org/api/TrenchCoordinates>, with a simple web application at <http://rdf.muninn-project.org/TrenchCoordinates.html>.

These were based on a Bonne projection with a Delambre ellipsoid and used the metric system [6,5,8].

These were then merged with information obtained from the French Government about their network of triangulation stations, magnified large scale maps of France, *Plan Directeur* fire control maps and some manual survey works. The set of sheets thus extended beyond the Belgian borders and into France. For reasons that are lost to history, the decision was made to overlay a grid in Imperial (yard) measurements over the metric projection meaning that in some cases duplicate trench coordinates exist and overlap with others.

The specifics of the coordinate systems are reviewed in other documents [2,14,6] but it consists of an alphanumeric string read left to right with increasing accuracy. Most recorded coordinates result in a 50 yard sided square, through a smaller squaring system of a 5 yard sided square was also used<sup>5</sup>. As an example: the location of a trench coordinate such as 27.L.22.d.6.3 would be a 50 yard sided box with centroid 50.8300, 2.7005.

The origin of the original Belgian projection is important because it is used to calculate a conversion between the Bonne projection and the WGS84 datum. It is purported to be the old Brussels observatory, which moved several times and the exact coordinates of the origin remains a point of contention. Positions officially recorded by Mugnier [10] as  $50^{\circ}25'0.0006''$ ,  $4^{\circ}22'12.6978''$  and Winterbotham [14] (see also Close [6]) as  $51^{\circ}10'06.895''$ ,  $4^{\circ}22'05.89''$  are both several kilometres off. A recalculation from the original Belgium Triangulation (1867, [4]) by the Belgium Geographical Institute yielded an origin of  $50^{\circ}24'$ ,  $4^{\circ}22'5.89''$  and after adjustment using several referenced church steeples,  $50^{\circ}23'57.2418''$ ,  $4^{\circ}22'10.0518''$  currently yields results with an average positional error of less than 0.0001 degrees of latitude and longitude.

One of the interesting elements that has caused not a little amount of frustration on the part of the authors is the uneven precision of the maps and the difficulty in obtaining precise location information for referenced land marks within the maps. The angle of observation of the overhead imagery tends to induce errors when trying to locate church steeples precisely and makes the resolution of the origin difficult. In any event, it is unclear that a local surveyor would have been of help: the French trigonometric points originally used by the ordnance survey referenced some churches that had since been moved before the war and others that were destroyed and rebuilt after the war.

These inaccuracies and problems reflect both the expected “fog of war” as well as some less-than-comprehensible events such as the faulty transcriptions of the locations of French trigonometric stations by cartographers and carelessness in print shops. Peter Chasseaud [5] reviews some of these events in details which

---

<sup>5</sup> Owing to the particularities of the British Trench Map Coordinates system, a grid reference can be of a number of different rectangular or square shapes. We use the grid square for convenience in the text.

are at times comical and tragic. Some maps have a grid offset by several thousand yards while others have a grid that is inexplicably printed backwards<sup>6</sup>.

All of the different uncertainties with the trench coordinates make for a conversion process that can at times return mathematically and geographically sound transformations with historically inconsistent results, *caveat emptor*.

At maximum accuracy, this system's limit was a square with sides of 5 yards through the 50 yard square is much more prevalent. Depending on the sources used to create new map sheets or update the original Belgian ones, the accuracy of the maps would vary dramatically. A complete update of a map by a ordnance survey team would be expected to be accurate within 20 yards [14].

### 3 Previous Work

Coordinate translation is a common problem that has been tackled comprehensively by Gerald I. Evenden [7] with his Proj4 package (used within our API). Most recently, Troncy et al. [13] published a Java servlet API to translate automatically across different projection in an automated fashion. From a markup and ontological perspective, the (modern) Ordnance Survey has been publishing Linked Open Data using its own coordinate ontology which has inspired our own. In terms of a representation of geometries, Claus et al. [12] used their own RDF constructs, along with the OGC GeoSPARQL [11] and NeoGeo vocabulary<sup>7</sup> to create the Linked Geo Data representation of OpenStreetMap data.

## 4 Translation API design and implementation

In this section, we review the API and its uses. We begin by describing the operating modes of the API and then review the LOD structures used in its operations and reporting. We report on some analytical results on the opportunistic enrichment of API results in cases where it has not been requested and discuss our experiences in using different enrichment strategies when explicitly requested by the client.

### 4.1 Query formation

The primary use case is transforming either a trench coordinate or a WGS84 point to the other representation. This transformation is not completely seamless since we move from a continuous coordinate system to a grid based coordinate system. The query is sent to the API through an inline parameter `q` that can contain either a WGS84 location or partial trench map coordinate.

---

<sup>6</sup> This was not only a Commonwealth experience, the Central powers used a number of competing reference systems, including one grid that was shared between the 4th and 6th German Armies (See [1]), but referenced differently.

<sup>7</sup> <http://geovocab.org/doc/neogeo.html>

In designing this API, we tried to deal with the most common use cases without making decisions that could affect accuracy. Currently, WGS84 is one of the more widespread coordinate system and fits in well with the expectation of an end-user to be able to input the coordinates into their consumer GPS unit. In future work we will integrate the translation API described by Troncy et al. [13] into the API in order for other coordinate systems to be supported.

While queries are submitted through URL parameters, answers are provided in one of RDF, raw text or XML formats with additional LOD formats supported through the parameter `fmt` or content negotiations headers.

Precision and accuracy were important consideration in the handling of the conversion. A trench map is made from several different sources of mapping information: on site surveys, larger scale maps (1:100,000), fire direction maps and the original 1:40,000 Belgium grid plates. The precision one can expect of the map varies wildly depending on the sources used to create it. The re-projection of large scale maps (1:100,000) down to smaller (1:40,000) scale was performed often at the beginning of the war and in these cases one could expect errors of about 200 yards.

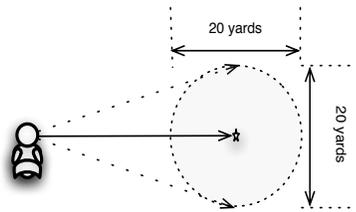
A survey unit making maps from sightings or aerial photographs would achieve a precision of about 20 yards as in Figure 2a. Hence any feature on a map should be expected to be within the area of a circle of a 10 yard radius. Similarly, the estimation of the origin confers an error to the corner points of any trench square. We explain in Section 4.2 in detail the LOD methods used to deal with this, including a separation between the terms that represent a trench coordinate and the corresponding WGS84 feature.

Dealing with the precision of a WGS84 query point remains problematic: A longitude of 5.03 cannot be distinguished from a longitude of  $5.0\bar{3}$  or 5.03000000 within most GIS systems. Yet in some cases, the location precision would be a valid means of identifying what size of trench square to return. Currently we transform the point “as-is” and return a square with sides of 5 yards. This is not always ideal and we are experimenting with the use of polygons or precision properties within the query to remedy this.

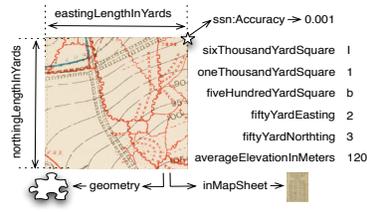
## 4.2 Converting coordinates with Linked Open Data

The API is fully enabled to use LOD approaches in reporting conversions to queries and LOD provides a number of tools with which we can avoid precision and accuracy errors from penalizing researchers and users.

Specifically, the conversion is known to introduce errors into the position and there are situations where the grid indicated on an Officer’s map was inaccurate. The irony is that this problem is brought upon by the use of technology; officers using these maps during the war would normally not have a problem since map series would be the same across organizational units and registration errors would thus be ignored. Recomputing the actual location that they were referencing depends as much on what feature was drawn on the map where as it does on the mathematical transformation.



(a) Feature accuracy was expected to be within 20 yards. Anecdotally, variations range from 5 yards to more than a 100.



(b) An abridged description of the classes and properties

Fig. 2: A representation of the precision and accuracy problems with coordinates translation

The use of RDF/OWL and of the paired ontology<sup>8</sup> was meant to support the following requirements: 1) The location as a British Trench Map Coordinate had to be a separate entity from its location in WGS84. This is both to deal with inaccuracy in the translation and to isolate the location as reported in the documents with the actual location where it occurred. 2) Any ontological statements had to have a minimum of expectation of truthfulness under the previously described problems of precision and accuracy.

The paired ontology contains the different instances of all map sheets used within the coordinates system, the relationships that bind them and the underlying organization of the coordinate system. The ontological structures borrow heavily from the modern British Ordnance Ontologies, Linked Geo Data Ontologies, GeoSPARQL and NeoGeo ontologies. The ontology also provides a convenient repository for the storage of known instances of trench maps so that an imaged version can be quickly located. Representing the geometry is done using a mixture of different style of representation, previously looked at by Auguste et al. [3].

A trench coordinate is a Feature that contains all the information about a trench map location but that is separated from geometry information. The geometry information is added through a property between query and response that always places the `geo:Point` `ogc:sfWithin` (or `ogc:sfContains`) a coordinate square. This ensures the statement is always logically true: because of accuracy issues, we cannot state what the `geom:geometry` actually is. The benefit of publishing an ontology capable of handling native trench map coordinates is that the locations can be referenced without committing to a specific longitude/latitude translation. This allows authors of semantic web data-sets to use coordinates as a means of locating additional information at that exact location, nearby or within a greater geographic areas. Enforcing the separation of the different co-

<sup>8</sup> <http://rdf.muninn-project.org/ontologies/btmaps>

ordinate system using LOD is what allows people to interchange location data while allowing for uncertainty issues in the translations of the coordinates.

The position of a grid square is communicated using a series of geo:Point and OpenGIS “well known text” instances, one for each of the vertices of the shape. An additional point at the centroid is provided as a convenience for placing labels. The concurrent use of both Point and WKT terms allows for native access from both naive and GeoSPARQL-enabled SPARQL servers.

Calculating the theoretical precision of a transformation from a longitude, latitude point to a grid square is straightforward because the error can be determined from both significant figures and the physical size of the grid square. Documenting the precision is still problematic; there currently exists no standardized way of reporting precision information beyond the terms provided by the Semantic Sensor Ontology<sup>9</sup>. Currently precision information is reported through it and the Provenance Ontology<sup>10</sup>.

A method that is used to resolve this issue is the reuse of the reference points used to derive the origin of the Bonne projection. By tracking the accuracy of the computed coordinate transformation against the actual position of the reference points we can get an estimate of the map registration error in the area of a trench coordinate. As with precision information above, reporting this information to the end user is still not standardized from a linked-geo perspective.

Currently, this information is reported using the `ssn:Accuracy` term from the Semantic Sensor ontology which is still in the incubator stage. In some cases, there is sufficient information about the coordinate systems and maps series that a heat-map of the different probability areas can be reported. This style of data reporting is useful in risk analysis applications, such as located forgotten ammunition depots and an appropriate means of reporting it using linked data is still an open question.

An additional issue in the Trench Map System is that the use of yards for defining squares on a metric map means that the top and bottom of the grid spills into the next map sheet and some coordinates squares overlap. These confusing cases are returned with an additional `ogc:overlaps` to the alternate square in an attempt to signal corner cases.

### 4.3 API latency and response enrichment

The API has two use cases: it may be used with applications that are user-facing or it may be used for the batch geo-referencing of an archival collection of maps. These represent extremes of response time requirements. In the former case, the client will want as much enrichment as possible without increasing response time since this saves it from making further requests; in the latter, speed is less of a concern as the process is likely automated. Furthermore, in this case enrichment is likely already contained within the archive catalogue.

<sup>9</sup> <http://www.w3.org/2005/Incubator/ssn/wiki/SSN>

<sup>10</sup> <http://www.w3.org/TR/prov-o/>

```
GET /api/TrenchCoordinates?q=57c.i.11.d.5.6 HTTP/1.1
User-Agent: curl/7.22.0 (x86_64-pc-linux-gnu) libcurl/7.22.0
    OpenSSL/1.0.1 zlib/1.2.3.4 libidn/1.23 librtmp/2.3
Host: rdf.muninn-project.org
Accept: application/rdf+xml
```

Fig. 3: Sample API request

```
HTTP/1.1 200 OK
Date: Thu, 1 Aug 2014 14:10:48 GMT
Server: Apache/2.2.22 (Ubuntu)
Pragma: no-cache
Cache-control: no-cache
Access-Control-Allow-Origin: *
Content-Encoding: gzip
Content-Type: application/x-gzip
Expires: Thu, 1 Aug 2014 14:10:48 GMT
```

Fig. 4: Sample response HTTP headers

Anecdotally, a typical document of this period such as a Regimental War Diary contains about 4 coordinate references per handwritten page. If we assume that the maximum permissible page load time is a generous 5s, requesting a single coordinate should not take longer than 1s. Besides careful provisioning and administration of the server, the network remains the most important factor in meeting this constraint.

If a request and its response each comprise only one network packet, there is no lag from packet reordering or fragmentation and reassembly which is the best scenario. Two questions arise: (i) is an average network packet big enough to hold typical responses, including protocol overhead? and (ii) how much space is left in this packet to enrich the responses with additional practical URIs? In this section we provide answers to these two questions.

**A response in one packet** As a first step we estimate the expected maximum size of one network packet. Any given path through the Internet is composed of one or more network links. Each of these links has a *maximum transmission unit* (MTU) size which reflects the largest packet that the link can carry. Messages larger than this must be broken into multiple packets. When data traverses several network links en route from sender to receiver, the smallest MTU along the path determines the size of message that may be sent without being fragmented along the way. The *maximum segment size* for a transport protocol like TCP will be this MTU minus protocol overhead. HTTP is used by the API and it runs over TCP, so this MSS value is what interests us. Luckie and Stasiewicz found an MSS of 1460 bytes for about 86% of IPv4 paths (and 1440 for 85% of IPv6 paths) [9].

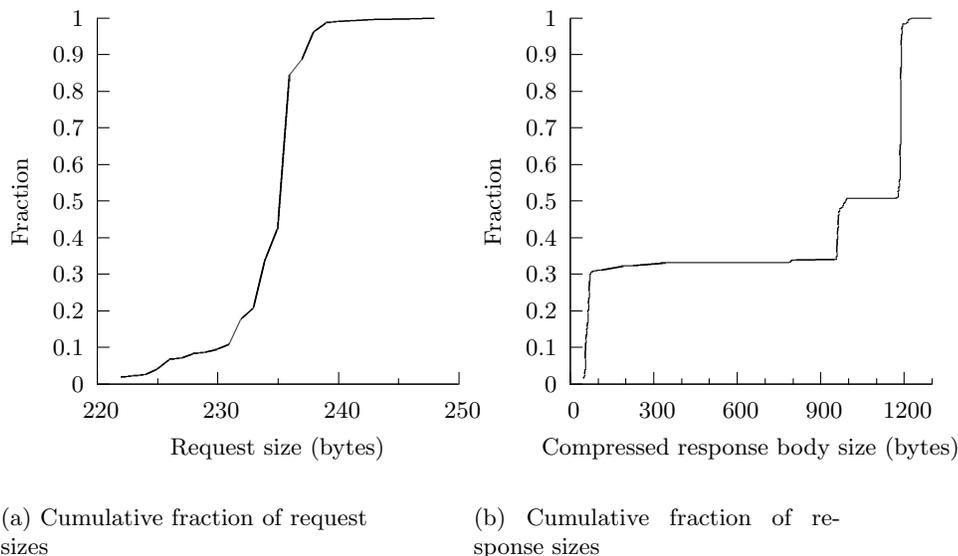


Fig. 5: Cumulative fraction of request and response sizes

Figure 3 shows a sample API request. 6111 representative API requests were recorded from the public-facing server over one month and Figure 5a is a plot of cumulative sizes, i.e., the fraction of requests that are at or below a given size (for example, about 40% of requests are at most 235 bytes). All requests were under 300 bytes, so each request that was observed fits into one packet.

Figure 4 shows the headers from a typical API response. These headers amount to 259 bytes (each line must be terminated by the two-byte sequence `0x0d 0x0a`) and with the required blank line leaves 1199 bytes left in an IPv4 packet (1179 for IPv6). The response body is be about 3500 bytes for a basic grid reference but the variant of Lempel-Ziv compression [15] used by the `gzip` tool can reduce this. Figure 5b is a plot of cumulative sizes of 4813 of these compressed responses. (Note that some of the requests used in figure 5a were invalid and so led to no response.) About 70% of these will fit, alongside the protocol overhead of headers, within one packet.

**Enriched responses in one packet** The primary form of enrichment we have in mind is the inclusion of URIs of “relevant” information, where relevancy is defined by the server based on context. In other words, whenever possible the API will attempt to “round out” a response packet with opportunistic linkages. A number of these are present within the ontology used to track map instances which reference geonames entities and battle locations. In these specific circumstances, the cost of adding extra triples to the response packet has already been amortized while the data may be of benefit to the requesting client.

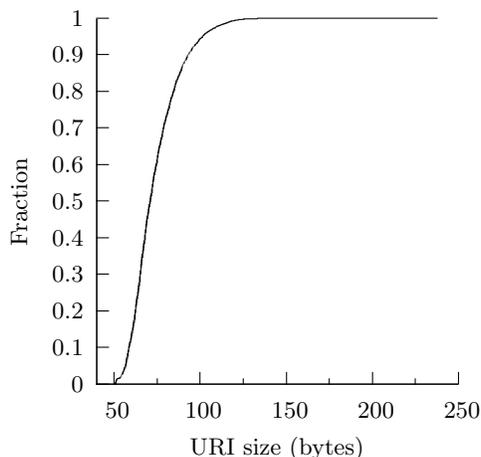


Fig. 6: Cumulative fraction of URI sizes in bytes

To evaluate the extent of enrichment that is feasible whilst retaining the advantages of a single-packet response, we use a set of URIs linking to DBpedia; we believe that this forms a representative set as DBpedia stands at the centre of the Linked Open Data Cloud and is arguably one of the better known data-sets. We obtained a list of 1,566,746 such URIs<sup>11</sup>. Figure 6 shows the cumulative fraction of URIs that are under a certain size; more than 90% of URIs are smaller than 100 bytes. In conjunction with figure 5b, this suggests that about half the time at least one URI can be expected to fit in a single packet alongside the API's response. A nontrivial fraction of requests (about 30%) will leave room for two or three enrichment URIs.

#### 4.4 Explicitly enriching answer terms

In the previous section, we looked at opportunistically enriching the query answer because the cost of doing so was negligible. This is the default behaviour of the API that can be turned off with the `enrich=none` parameter.

In other cases a request for enrichment of the query results can be made through the enrichment parameter `enrich=full`. This data can be derived from the trenchmap ontology or external data sources, including geonames for country information or larger data-sets.

The French National Library publishes extensive cataloguing data as part its experimental data portal<sup>12</sup>. We considered querying it as part of our enrichment strategy in that the library contains a number of map holdings of the periods

<sup>11</sup> [http://downloads.dbpedia.org/3.9/en/iri\\_same\\_as\\_uri\\_en.nt.bz2](http://downloads.dbpedia.org/3.9/en/iri_same_as_uri_en.nt.bz2), retrieved 2014-07-07

<sup>12</sup> <http://data.bnf.fr/>

before and after the conflict. The problem is that most of the geo-spatial data is still locked within human readable strings, making the geo-location of a map difficult.

Similarly, the German National Library<sup>13</sup> does provide OGC triples for some of its holdings and the API reports them on an opportunistic basis. However, the library lacks a SPARQL server with which to query the data which requires the loading of data into a local SPARQL server for use. Besides data-set size, in this case about the size of the average hard drive in a desktop, there is no means of keeping the local copy synchronized for updates. Given that the average query is likely not to find a document relevant to the locality of interest, the investment in storage and bandwidth is hard to justify.

A promising area of research is in the retrieval of a “working set” of triples that are relevant to the researchers or study’s area of interest. Currently, this is exemplified by the non-LOD APIs such as the OSM Overpass<sup>14</sup> that allows clients to retrieve all features within a bounding box.

In the case of the trench map API, it can serve as a retrieval engine by enumerating all known features from the Great War era that are within the area of interest, such as all known features (`enrich=full`) of Sheet 36C. The advantage of this API feature is that all of the required data is retrieved in one transaction. Usually when a client is faced with a server whose query engine is limited a strategy of flooding the server with multiple queries is used which can result in an overload of the server. If the client interleaves the requests, then the time delay cost due to latency makes the query impractical.

## 5 Future work and Conclusion

In this paper we have presented an API used for retrieving geo-spatial information reckoned in obsolete military coordinates. The principal job of servers implementing this API is to translate from these coordinates, dating from the Great War, into a modern coordinate system. The issues encountered in processing requests using a Linked Open Data approach were reported on.

There is nothing in the API’s design that ties it to the British Trench Map Coordinate System. The Central powers also had their own coordinate systems for the Western Front, including one that was used in two different ways by two different German armies. Given the obvious overlap in operational records, it would be useful to be able to use the API to link the same locations referenced by the different belligerents. We expect such generalizations to be straightforward and valuable to historians.

Finally, the use of LOD provides opportunities to link the geo-spatial information relevant to the API with other information. For example, established social network vocabularies such as Foaf include terms for interests, such as

---

<sup>13</sup> <http://www.zeitschriftendatenbank.de/de/services/schnittstellen/linked-data/>

<sup>14</sup> [http://wiki.openstreetmap.org/wiki/Overpass\\_API](http://wiki.openstreetmap.org/wiki/Overpass_API)

foaf:interest. Given the pointed nature of our geo-spatial data, it may be possible to leverage Linked Open Data to match different historians with interests in specific locations (and contexts) for future collaborations.

## Acknowledgements

The authors wish to thank Pierre Voet for his help with the Belgian triangulation and Bill Frost for access to his database of church steeples.

## References

1. Imperial War Museum, WW1 Map M-025838.
2. Howard Anderson. *How to find a point on a Great War trench map*. The Western Front Association.
3. Ghislain Atemezing and Raphaël Troncy. Comparing vocabularies for representing geographical features and their geometry. In *11th International Semantic Web Conference, Terra Cognita Workshop*, volume 901, 2012.
4. Institut cartographique militaire. *Triangulation du Royaume de Belgique*. Number v. 2-3 in *Triangulation du Royaume de Belgique*. Cnophs, 1867.
5. Peter Chasseaud. *Artillery's astrologers : a history of british survey and mapping on the western front 1914-1918*. Naval and Military Press Ltd, March 1999.
6. Charles Close, Lieut.-Colonel Jack Keeling, and et al. Lieut.-Colonel Salmon. British survey on the western front: Discussion. *The Geographical Journal*, 53(4):pp. 271–276, 1919.
7. Gerald I. Evenden. *Cartographic Projection Procedures for the UNIX Environment - A User's Manual*, September 1995.
8. E. M. Jack. *Report on survey on the Western Front 1914-1918*. His Majesty's Stationery Office, London, 1920.
9. Matthew Luckie and Ben Stasiewicz. Measuring path mtu discovery behaviour. In *10th Conference on Internet Measurement*, pages 102–108, 2010.
10. Clifford J. Mugnier. The kingdom of belgium. *Photogrammetric Engineering & Remote Sensing*, pages 956–957, October 1998.
11. OGC. OGC GeoSPARQL - a geographic query language for RDF data. Technical Report OGC 11-052r4, Open Geospatial Consortium, 2012.
12. Claus Stadler, Jens Lehmann, and et al. Linkedgeodata: A core for a web of spatial open data. *Semantic Web Journal*, 3(4):333–354, 2012.
13. Raphaël Troncy, Ghislain Atemezing, and et al. Modeling geometry and reference systems on the web of data. In *W3C Workshop on Linking Geospatial Data*, 2014.
14. H. S. L. Winterbotham. British survey on the western front. *The Geographical Journal*, 53(4):pp. 253–271, 1919.
15. J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theor.*, 23(3):337–343, May 1977.