Large Scale Semantic Annotation, Indexing, and Search at The National Archives

Diana Maynard and Mark A. Greenwood

Department of Computer Science
University of Sheffield
Regent Court, Sheffield, S1 4DP, UK
diana@dcs.shef.ac.uk

Abstract

This paper describes a tool developed to improve access to the enormous volume of data housed at the UK's National Archives, both for the general public and for specialist researchers. The system we have developed, TNA-Search, enables a multi-paradigm search over the entire electronic archive (42TB of data in various formats). The search functionality allows queries that arbitrarily mix any combination of full-text, structural, linguistic and semantic queries. The archive is annotated and indexed with respect to a massive semantic knowledge base containing data from the LOD cloud, data.gov.uk, related TNA projects, and a large geographical database. The semantic annotation component achieves approximately 83% F-measure, which is very reasonable considering the wide range of entities and document types and the open domain. The technologies are being adopted by real users at The National Archives and will form the core of their suite of search tools, with additional in-house interfaces.

Keywords: semantic annotation, indexing, information extraction

1. Introduction

The National Archives (TNA)[1] are the UK government's official archive, containing over 1,000 years of historical data which is made publicly available. They work with 250 government and public sector bodies, helping them to manage and use information more effectively. The archive is one of the largest in the world, comprising over 11 million historical government and public records. In general, government records that have been selected for permanent preservation are sent to The National Archives when they are 30 years old, but many are released earlier under the Freedom of Information Act. Amongst other things, the archives contain information from government, diplomacy and the armed forces (e.g. documents from all government departments), court records, census records, alien arrivals, birth, marriage and death certificates and approximately 6 million historical maps. Many of the records are currently only available in paper form, but in addition to the government and military records, the online documentation contains many digitised public records, e.g. famous historical wills, selected records from MI5 and MI6, a range of UFO-related files from the Ministry of Defence, WWI and WWII selected records, and so on.

This paper describes some of the work carried out as part of the Government Web Archive Project[2], which aims to help open up TNA's records of government websites (going back to 1997 and comprising some 700 million pages). Government funding has been allocated to publishing more and more material on government websites in open and accessible forms, but in many cases it is still very hard to find the information needed, because the search tools are quite basic and only enable a keyword-based search. Sophisticated and complex semantics can transform this archive, but the real trick is to show how simple and straightforward mechanisms can add value and increase usage in the short and medium terms.

The system we have developed aims essentially to improve access to the enormous volume of data at TNA, both for the general public and for specialist researchers, by enabling a semantic-based search for categories of things, e.g. all Cabinet Ministers, all cities in the UK. Search results can include morphological variants and synonyms of search terms, specific phrases with some unknowns (e.g. an instance of a person and a monetary amount in the same sentence), ranges (e.g. all monetary amounts greater than a million pounds), restrictions to certain date periods, domains etc., and any combination of these. The search functionality allows queries that arbitrarily mix any combination of full-text, structural, linguistic and semantic queries, and can scale to gigabytes of text. Our experience is that faceted and conceptual search over spaces such as concept hierarchies, specialist terminologies, geography or time can substantially increase the access routes into textual data and increase usage accordingly.

2. System Architecture

TNA-Search aims to import, store and index structured data relevant for the web archive in a scalable semantic repository, in an easy to manipulate form, using linked data principles and in the range of tens of billions of facts. Links are made from the web archive documents into the structured data, over hundreds of millions of documents and terabytes of plain text. The system allows browsing, search and navigation from the document space into the structured data space via semantic annotation and vice versa via a SPARQL endpoint, both as full text and as linguistic annotation structures. For example, it enables complex queries by different kinds of users, ranging from the medical student who wants to find all medical publications relating to Type 2 diabetes, to the public user who wants to search on a specific minis-

[1] http://www.nationalarchives.gov.uk/

[2] http://www.nationalarchives.gov.uk/webarchive/

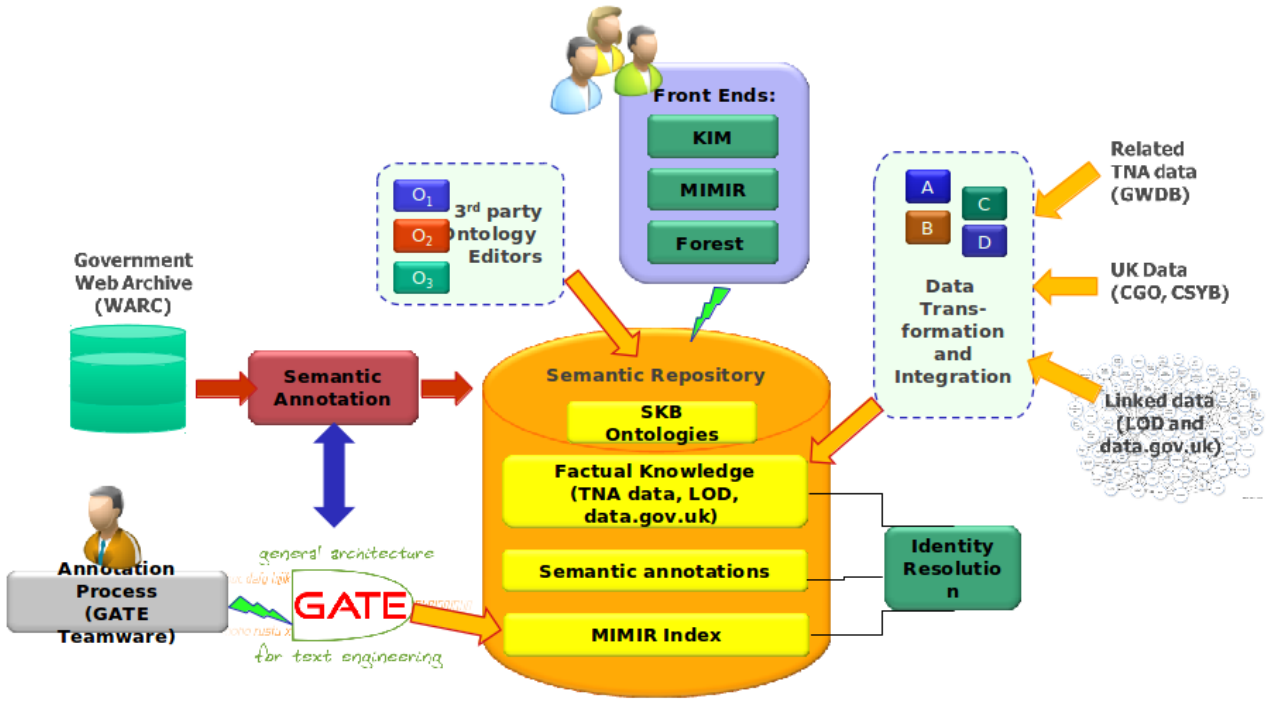

Figure 1: System Architecture

ter to see their career in government, e.g. for information about Gordon Brown's career in government, to the very specific request from a person working in the Treasury Office who wants to find a document about the Chancellor's statement on Northern Rock, which was formerly on the Treasury website but has since been removed.

TNA-Search is built on GATE (Cunningham et al., 2002) for semantic annotation and GATE Mímir (Cunningham et al., 2011) for indexing and search, and relies on a huge semantic repository combining FactForge and the SKB (Semantic Knowledge Base) ontology. Factforge[3] is a knowledge base developed by Ontotext[4] containing over 2.2 billion statements and containing datasets from DBPedia, Freebase, Geonames, UMBEL, WordNet, CIA World Factbook, Lingvoj and MusicBrainz. The SKB ontology was developed by Ontotext specifically for TNA-Search, based on the Central Government Ontology (CGO) and comprises official titles in the UK government (e.g. "Secretary of State for Health"), 8138 names of officials, and names of unambiguous UK government organizations. The CGO consists of a class hierarchy which describes governmental organizations (22 classes), governmental roles (19 classes), and concepts around the functioning of the government (2 classes). The class hierarchy is supplied with 33 properties describing relationships between the government organizations and roles and the people who hold the roles, e.g. memberOfCabinet, or between different kinds of government organizations, e.g. hasCabinet. The knowledge base connects disparate elements in the UK Government Web Archive to create a more "joined up" experience for the user, showing associations, linking between sites and allowing the archive to be segmented in different ways (e.g. everything related to DEFRA).

The basic methodology comprises the following set of steps:

1. Annotate the documents using GATE Embedded: this uses ontology-based information extraction (OBIE) tools to annotate entities in text and relate them to the ontology where appropriate.

2. Index the documents using GATE Mímir: this consists of creating an index based on the annotations produced in Step 1.

3. Search the documents using GATE Mímir: this entails querying the GATE Mímir index using full text, SPARQL or annotation-based queries.

4. Browse the results: the search results point back to the texts in the original archive.

Figure 1 depicts the overall architecture of TNA-Search. The source data is a set of ARC files representing the Government Web Archive[5]. Semantic annotation is performed by GATE, with respect to the large semantic repository comprising the SKB and factual knowledge from Factforge (see Section 4.4). Once the GATE Mímir index has been created, any one of the three front-ends can be used to query the data. In this paper, we concentrate only on GATE

[3] http://factforge.net
[4] http://www.ontotext.com

[5] These files are generated by the Internet Memory Foundation as they crawl the government websites.

Mímir, but KIM (Popov et al., 2004) and Forest[6] can be used as alternatives. The main difference between them is that KIM does not provide access to the data via SPARQL queries, but on the other hand it allows faceted browsing. Forest only allows access to the data via SPARQL, while GATE Mímir allows access via any combination of full-text, structural, linguistic and SPARQL queries.

3. Related Work

Semantic annotation is the process of attaching metadata tags and/or ontology classes to text segments, as an enabler for knowledge access and retrieval tools. Automatic annotation is carried out by employing Information Extraction (IE) (Cunningham, 2005) techniques, which automatically recognise instances of a given set of events, entities or relationships. From an algorithmic perspective, IE approaches fall in to two broad categories: manually engineered ones (frequently based on pattern-matching rules, e.g. (Maynard et al., 2001)) and machine learning ones (e.g. (Bikel et al., 1999; Li et al., 2005)). From an operational perspective, IE tools can be deployed in both fully and semi-automatic applications (where users can inspect and, if needed, correct the automatically created metadata). In general, fully automatic methods are preferred when the volume of data is too large to make human post-annotation practicable, as is the case with our scenario. Ontology-based Information Extraction is not new (Müller et al., 2004; Maynard et al., 2007) but previous techniques do not allow for complex annotation to be performed in this way with respect to huge ontologies and to the Linked Open Data Cloud, as described in Section 4.4.

A previous version of GATE Mímir has been used for patent annotation and searching (Cunningham et al., 2011). The current version has a number of structural changes and improvements, including the ability to create a federated index and to handle more than one index per instance. The patent annotation tool differs in other ways: it was based on a relatively narrow domain, was on a much smaller scale, and did not permit the use of SPARQL queries.

GATE Mímir is based on ANNIC (Aswani et al., 2005), a tool designed to support the development of finite state transduction patterns in GATE's JAPE language (Cunningham et al., 2000). ANNIC is used to search corpora that have been annotated and then indexed using Lucene[7]. Users make searches based on a query language very similar to JAPE and are presented with a results summary similar in form to KWIC (Key-Words In Context) tools; unlike GATE Mímir, queries using SPARQL are not possible, and it is not scalable.

While there exist a variety of tools which enable more complex search than a standard keyword-based search, there are no tools that we know of based on a single semantic index which are capable of combining structured semantic queries, annotation pattern-based queries, full-text search and faceted search in a single query. Hybrid search tools effectively combine keywords and semantic search (e.g. (Bhagdev et al., 2008; Rocha et al., 2004)), but do not have the functionality provided by an annotation-based search and do not enable the kind of complex queries offered by GATE Mímir (for example, they cannot reliably return sentences in a particular section of a document from a particular domain where a cabinet minster talks about a London hospital spending between 1 and 10 million pounds at a date between July 2009 and January 2010).

4. GATE Application

The GATE application consists of a set of processing resources (PRs) executed sequentially in a conditional pipeline over a corpus of documents. The conditional pipeline enables us to run some PRs only if certain conditions about the document are true: this is particularly useful when dealing with a heterogeneous dataset such as the government archives. The pipeline consists of 6 main parts:

- Linguistic pre-processing
- Gazetteer lookup
- Rule-based Annotation
- Semantic Annotation
- Co-reference
- Final output creation

4.1. Linguistic pre-processing

The linguistic pre-processing phase contains standard GATE components such as tokenisation, part-of-speech tagging, morphological analysis and so on. Details of these standard components can be found in the GATE User Guide[8]. It also contains specialised components for content detection (Boilerpipe) and for number recognition (Numbers Tagger). These components have now been contributed to GATE and are included in the standard distribution as plugins.

The **Content Detection** PR uses the Boilerpipe Java library[9] to detect and remove the surplus "clutter" (boilerplate, templates) around the main textual content of a web page. A Content annotation is created on the meaningful content of the document, meaning that this can be incorporated in a search query. For example, one can search for an instance of a Person only if it appears inside the meaningful content of the document.

The **Numbers Tagger** is a special PR which finds numbers written using words in the document, e.g. "three million" as well as numerous different number formats, including exponential numbers. This is used in conjunction with the Measurement and Date Normalisation PRs described in Section 4.3.

4.2. Gazetteer Lookup

The gazetteer lookup phase comprises a combination of default gazetteer lists from ANNIE (GATE's vanilla information extraction system), some newly developed gazetteer

[6] http://www.ontotext.com/sites/default/files/downloads/IntroducingForest_2010.pdf

[7] http://lucene.apache.org/java/

[8] http://gate.ac.uk/sale/tao

[9] http://code.google.com/p/boilerpipe/

lists relevant to the government domain (e.g. government departments, positions, agencies, abbreviations and so on), and the LKB[10], a gazetteer of relevant entities complete with URIs, generated on the fly and based on the semantic repository (SKB + Factforge). Section 4.4 explains in more detail how this is used.

4.3. Rule-based Annotation

Rule-based annotation is performed by a number of taggers: in addition to the default ANNIE resources, we have a Document tagger for finding the body, title and domain of the document, a Government tagger for finding government-related entities not covered by ANNIE (e.g. cabinet ministers, different kinds of reports and official documentation, military conflicts, projects and so on), and taggers for finding and normalising dates and measurements. This normalisation is a crucial step for the indexing and search, because it enables search to be carried out over dates and measurements expressed in many different forms. Similarly, finding the various subcomponents of the document (title, body and domain) enables us to search over any of these independently, if we so wish. The grammars also modify some of the annotations produced by ANNIE: for example, some organizations can be considered as both agencies and departments (e.g. The National Archives), and so we want these to be matched when a search query mentions either term.

The **Measurement Tagger** builds upon the number annotations created by previous resources in the application, using them as cues as to the most likely locations of measurements within the documents. One of the main challenges in recognising measurements comes from the large number of measurement units in existence. Another challenge is that some units have single letter abbreviations, which introduce ambiguities in many cases: for example when we encounter "1C", we need to distinguish temperature from other cases, such as references to figures, examples, tables, etc. (as in "see Figure 1C"). Most measurements comprise a scalar value followed by a unit, e.g. $2x10^{-7}$ metres, while two scalar values with or without a unit can be contained in an interval. Sometimes there are also accompanying words, such as "less than" or "between" which can be important for searching, e.g., "less than about 0.0015 mm". The Measurement Tagger is based on an open-source Java port by Roman Redziejowski[11] of the GNU Units package[12].

The **Date Normalizer** is a special PR which attempts to determine for each date instance in the document the fully specified date to which it refers, using an open-source date parser[13]. Documents are always written or published within a specific context: one of the more common reasons for requiring this context is understanding the timeline of a document. For example, if a document refers to any date that is either relative (e.g. today, yesterday, last Tuesday) or not fully specified (e.g. 14th February), then the date on which the document was written or published is needed in order to determine the date being discussed.

The approach we have taken to finding the date of the document is to employ a back-off strategy through the following date sources:

- DocumentDate: these annotations are usually the most accurate, and are extracted from the body of the document by the Government Tagger.
- http_header_Last-Modified: this is a document feature which, if available, states the time at which the web server that served the page thought the document was last changed.
- http_header_Date: this is also a document feature, and is the time (including the date) at which the page was served during the crawl.
- arc_header_creation-date: this is the time at which the ARC file in which the page is stored was created, and is likely to be the least accurate source of the document date.

Once normalized, the date is stored as an integer to enable easy range searches in GATE Mímir – the numeric format can be read as yyyymmdd.

4.4. Semantic Annotation

In order to enable semantic queries on the data, the relevant entities in the documents need to be linked to the various ontologies. This means that one can then search for e.g. mentions of all UK cities, or all Persons who are Cabinet Ministers, even when these facts are not expressed in the documents themselves.

The **Large Knowledge Base (LKB) gazetteer** enables us to annotate certain concepts directly from the semantic repository, rather than from a predetermined and flat set of gazetteer lists. In theory, this should lead to greater coverage and better precision; more importantly, however, it means that certain annotated entities are linked to specific instances in the semantic repository. The LKB is part of the GATE distribution and provides efficient representation of very large vocabularies, as well as query-based selective loading from RDF databases. The instance we use in this application is loaded from `http://skb.ontotext.com`.

The LKB makes use of a number of configuration files such as the set of SPARQL queries to be used on the ontology. For example, a query to find Persons in the SKB is shown below:

```
SELECT ?inst, ?cls WHERE {
?inst rdf:type ptop:Person .
?inst ff:preferredLabel ?label .
?cls a owl:Class .
FILTER regex(?label, "") .
FILTER(?cls = ptop:Person) }
```

The linking between the annotated entities and the instances in the SKB is done in two complementary ways. First, class and instance information from the SKB is added to relevant entities in the text, where a match is found via the LKB

[10] http://nmwiki.ontotext.com/lkb_gazetteer/

[11] http://units-in-java.sourceforge.net/

[12] http://www.gnu.org/software/units/

[13] http://greenwoodma.servehttp.com/jenkins/job/Date\%20Parser/

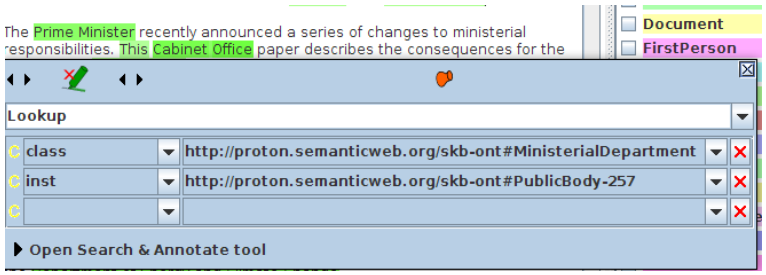

Figure 2: Lookup annotation for "Cabinet Office"

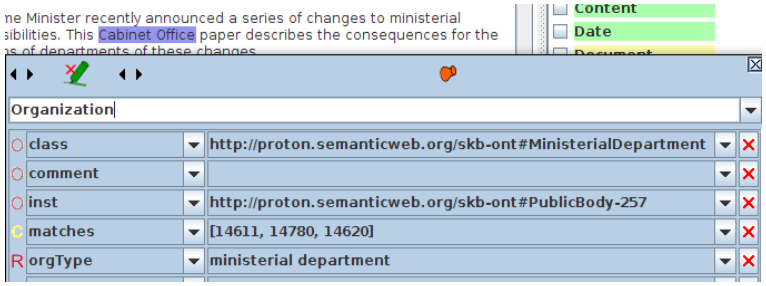

Figure 3: Organization annotation created for "Cabinet Office"

gazetteer. Second, entities in the text which do not have a direct link to an instance or class in the SKB may have this information inferred by means of co-reference. If a mention in the text has been linked to the SKB by means of the above process, all co-referring mentions of that same entity can automatically have the same class and instance information added to them, by means of the TNA Instance Generator (see below).

In order to make use of the LKB lookups, a JAPE grammar is required to match these lookups and to create for each one an annotation of the appropriate type. It also copies the class and instance features onto the new annotation. For example, a Lookup from the SKB ontology with class `http://proton.semanticweb.org/skb-ont#MinisterialDepartment` and instance `http://proton.semanticweb.org/skb-ont#PublicBody-257` (as depicted in Figure 2) will be given an Organization annotation, an "orgType" feature whose value is "ministerial department", and the relevant class and instance features (as depicted in Figure 3).

4.5. Co-reference

The **Orthomatcher** links different orthographic variants of the same entity within a document, e.g. "D. Cameron" and "David Cameron". It is set here to operate over Person, Location and Organization annotation types. More specifically, the Orthomatcher adds identity relations between annotations found by the semantic tagger, in order to perform co-reference. The matching rules are only invoked if the names are of the same type, e.g. Organization. This prevents a previously classified name from being re-categorised. The Orthomatcher is set to run in the pipeline only if certain conditions are met (less than 500 annotations of the same type are present) in order to avoid huge delays in processing caused by long lists of names and addresses.

The **TNA Instance Generator** assigns class and instance URIs from SKB to annotations in the document, based on co-reference information. The Instance Generator generates URIs for entities, taking into account in-document co-reference and the lookups that the gazetteer found in the document. For example:

1. if one document mentions Person annotations "David Cameron", "D Cameron" and "Mr. Cameron";
2. and the orthographic co-reference has linked those three together using a "matches" feature;
3. and the LKB gazetteer has created a Lookup over "D Cameron";

then the Instance Generator will copy the "inst" and "class" features, containing URIs, from the Lookup over "D Cameron" to all three Person annotations. If only 1 and 2 are satisfied, the Instance Generator will generate a URI, using the label of the longest Person annotation (either the annotated text or the originalName feature if present). In the example above, the URI would be `http://proton.semanticweb.org/skb-ont#Person_David_Cameron`. The Instance Generator will not generate URIs for newly recognised entities where the name extracted from the text is too ambiguous. Specifically, URIs will not be generated for a set of co-referenced Person annotations, where the longest label is a single name like "David" or "Cameron".

5. Annotation Evaluation

The quality of the annotations generated by the Information Extraction system was evaluated using the metrics of *Precision*, *Recall*, and *F-Measure*. A gold standard corpus was produced by 4 manual annotators (domain ex-

Annotation Type	Frequency	P	R	F_1
Address	33	86	88	87
Conflict	1	100	100	100
Date	206	79	95	86
Location	168	86	01	89
Military Operation	1	100	100	100
Money	48	93	97	95
Official Document	35	63	74	68
Organization	326	77	85	81
Person	37	87	76	81
Post	48	92	65	76
Project	21	92	64	75
Total	924	81	85	83

Table 1: Evaluation of IE system

perts) at The National Archives, consisting of 13 documents selected from the archive. The documents were selected to cover a range of different domains and were annotated using GATE Teamware. Once the documents had been double-annotated, they were curated by the system developers to create a single gold standard version. Inter-Annotator agreement was found to be reasonable, with an F-measure of 79%: this is an acceptable level, but shows also that the task is hard, even for humans. The prototype system was then run and the results compared with the gold standard, using the Corpus Quality Assurance Tool available as part of GATE Developer.

The results are shown in Table 1, in terms of frequency (i.e. total number of entities of this type in the gold standard corpus), Precision (P), Recall (R) and F-measure (F1). We note that, compared with the archive as a whole, the relatively small frequency of occurrence of some annotation types could skew evaluation results slightly. However, if we remove the two annotation types Military Operation and Conflict from the evaluation, the total scores are identical, precisely because the occurrence of these types is so small as to barely affect the results, and since we use a Micro Summary rather than a Macro Summary for the totals. A more comprehensive evaluation involving a much larger number of documents (and thus annotations) could give a more accurate indication of results; however, based on the iterative development cycle involving the training data, we do not predict that such an evaluation would give widely differing results unless the format or content of the data used were significantly different.

Inspection of the inter-annotator agreement for the manual annotation round produces also some interesting observations. Most annotation types had agreement in the 90th percentile, but some annotation types clearly were more problematic. Military Operations and Conflicts were rather sparse in the corpus, so their results are a bit misleading, but it was also not entirely clear to the annotators what constituted a military operation. Similarly, Projects and Official Documents appeared to be hard for the annotators to agree on, and these were also the types which got the lowest scores in the system evaluation.

For this type of task, we consider the results to be very satisfactory. As mentioned earlier, the task is hard even for humans, and the difficulty is increased by the fact that it involves both a wide range of entities and a rather open domain. There are a number of trade-offs between complexity of task, openness of domain, and accuracy of results, that are typically associated with an information extraction task (Cunningham et al., 2005). Essentially, a task becomes more difficult and therefore tends to elicit lower performance when the domain becomes more open and/or the nature of the recognition becomes more complex (e.g. moving from a few basic named entities to a more complex set of entities and/or relations). It is important to note also the trade-off between Precision and Recall. The system has been developed in a fairly neutral way with an equal balance towards the two. If necessary, the system could be tuned in future to favour one over the other, by tightening or relaxing the rules.

6. GATE Cloud Parallelizer

One of the main challenges in the task of annotating and indexing millions of documents from The National Archives is the sheer size of the data: at the time of processing, the archive contained 42TB (around 700 million documents), of which approximately 150 million documents were unique. Processing this data was achieved using the GATE Cloud Parallelizer (GCP)(Tablan et al., 2011), installed on the Amazon Cloud. The GCP is a platform for parallel semantic annotation of text documents, designed as a parallel version of the execution engine found in GATE. It takes a language processing pipeline created using the GATE Developer environment (in this case, the TNA-Search annotation pipeline detailed in Section 4, and executes it using a set of parallel threads. The job control is performed through document batches, which are XML files describing outstanding tasks.

7. Indexing and searching with GATE Mímir

In order to query the archive, we create a GATE Mímir index from the annotations produced by the annotation pipeline. GATE Mímir (Cunningham et al., 2011) is a multi-paradigm information management index and repository which can be used to index and search over text, annotations, semantic schemas (ontologies), and semantic metadata (instance data). It allows queries that arbitrarily mix full-text, structural, linguistic and semantic components, and can scale to gigabytes of text. The multi-paradigm aspect of GATE Mímir refers to the accessing and linking together of multiple information sources, such as the textual content of the documents, the semantic metadata associated with the documents, and data in knowlege bases (such as the linked data cloud). Documents are annotated and indexed in GATE Mímir on the fly, using a federated index for efficiency.

Accessing the data from a knowledge base allows GATE Mímir to understand generalisations, making it capable of answering more complex information needs, such as identifying documents that refer to “capital cities in Western Europe”. At the same time, the explicit semantics associated with the indexed documents ensures that references to any of the many places called London (other than the one in the

```
{Person semanticConstraint="?inst <http://proton.semanticweb.org/skb-ont#hasPosition>
?pos .  ?pos <http://proton.semanticweb.org/skb-ont#hasTitle>
<http://proton.semanticweb.org/skb-ont#OfficialTitle-Minister_of_State>"} root:say
```

Figure 4: Example GATE Mímir Query

UK) are not seen as relevant results to such a query. Additionally, support for sophisticated text-based queries allows GATE Mímir to filter the set of results to, for example, only those mentions that occur in the same sentence with the name of a particular government agency, or only within the "Contact" page of a given web site. Finally, GATE Mímir can use the semantic annotation of documents to perform some simple reasoning over the meaning of entities. For example, monetary amounts can be normalised according to the date of the document (to account for inflation), while measurements using different unit systems (such as inches and millimetres) can be matched against each other.

7.1. Examples of queries

The documents in the archive can be queried using search terms composed of combinations of actual words (strings), the annotation types and features mentioned previously, plus some additional annotations such as Token, Sentence, Document etc and their respective features, plus a variety of operators on these. GATE Mímir will return from the index all documents which contain the relevant matches. More general information about GATE Mímir, as well as a live demo on the TNA dataset, can be found at `http://demos.gate.ac.uk/mimir/`.

The simplest kind of query is simply matching against a string of text: this will perform an exact match. For example, searching for `Harriet Harman` will return every document that mentions Harriet Harman explicitly (but not, "Mrs Harman"). We can combine this with morphological analysis, e.g. searching for `Harriet Harman root:say` will match the exact string "Harriet Harman" followed by any morphological variant of the word "say", e.g. "Harriet Harman said". We can extend this to any person's name, rather than specifically to mentions of Harriet Harman, with the query `{Person} root:say`, which would match any person's name (that has been recognised by the system) followed by any morphological variant of the word "say", e.g. "John said", "John Smith says". Finally, we can incorporate semantic information, using a query such as that shown in Figure 4. This is the same as the previous query but with an additional semantic constraint. The SKB is used to limit the Person annotations that the query matches: this specific constraint limits the query to only matching those people that are listed as Ministers of State in the SKB.

7.2. Search Quality Evaluation

The prototype version of the TNA-Search tool was evaluated by users at TNA in order to compare it with the existing search facilities in place. The users compared the three different front-ends (GATE Mímir, KIM and Forest) with the European Archive Full Text Search[14]. 26 different search queries were tested on the 4 search techniques, and scoring was performed on the basis of ease of use and quality of results. Additional comments and feedback were also provided by the users for each query. Examples of queries are: *"As a researcher I'd like to find all the annual reports for DEFRA"* and *"As a journalist I'm looking for all the speeches made by Tony Blair about foot and mouth while Prime Minister."*

One of the major benefits of the TNA-Search tool (which was apparent in all 3 front-ends based on this technology) in this evaluation was the ability to perform multi-faceted queries. Overall, TNA-Search performed very well, the main drawback being that it was harder to use than the existing European Archive tool. Given further training, this would be less of a problem. Furthermore, the TNA-Search tool was designed primarily with function rather than interface in mind: the idea is that appropriate (and different) front-ends could be built on top of the technology in order to make the same tool appropriate for in-house researchers on the one hand and the general public on the other hand.

Essentially, custom GATE Mímir interfaces could be designed to target one specific type of search (e.g. people, government departments) and would consist of a form-based structure that removes the need for an end user to know (or ever see) the GATE Mímir search syntax. As an example of how this would work, an example interface called PiN (People in the News) has been developed (Greenwood et al., 2011) for finding people mentioned in news articles[15]. PiN enables users to search a corpus of news articles for people and allows the search to be restricted based on their name, where they were born, what they are famous for (e.g. sport, politician), as well as the date the article was originally published.

8. Conclusions

In this paper, we have shown how the use of simple and straightforward mechanisms can add value and increase usage in the short and medium terms to a huge archive of information. Our solution, based on GATE (text mining), OWLIM (semantic repository) and GATE Mímir (multi-paradigm search front-end components), provides search paradigms over semantic annotation that relates archival content to Linked Data and other structured sources. We import Linked Data into the semantic repository which provides a SPARQL endpoint (and also full text and annotation structure indices), and annotate the archive text relative to the repository. The knowledge base, which cross-references the UK Government Web Archive with a number of government-specific and more general ontologies, will continue to be progressively enhanced and developed, as will the complementary tools to mine both explicit knowledge within the archive and implicit knowledge which can

[14] `http://collections.europarchive.org/tna/adv_search/?lang=en&where=text&y=17&x=30`

[15] A live demo of PiN is available at `http://demos.gate.ac.uk/pin/`

be inferred from it. Text mining enables the extraction of facts from content and thus better interfaces, while the integration of the Semantic Knowledge Base makes it possible to combine a massive body of external knowledge with facts extracted from texts. The technology developed in the TNA-Search Tool is designed primarily as a technology source rather than as an end-product: we foresee the creation of more visual and interactive (suggestive) interfaces by experts at TNA.

Aside from the issue of improving the search interface, further work may involve better management of temporal information, such as dealing with changes of roles, department name/function and so on over time. This is partially supported via the semantic knowledge base, but enabling relevant search functionality is still not straightforward. For example, annotations such as Post (which represents a ministerial position, for example), have a timeline feature associated with them to represent former, current or future roles, but this stems from information valid at the time, e.g. mentions of "the former Prime Minister" in the document. This is different from mentions of someone who was Prime Minister at the time of writing, but is no longer Prime Minister. For search purposes, we may want to distinguish between the two things. Information about changes to the structure of government over time also need to be inferred from e.g. changes to the location of data within the structure of the web archive.

9. References

N. Aswani, V. Tablan, K. Bontcheva, and H. Cunningham. 2005. Indexing and Querying Linguistic Metadata and Document Content. In *Proceedings of Fifth International Conference on Recent Advances in Natural Language Processing (RANLP2005)*, Borovets, Bulgaria.

Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, and Daniela Petrelli. 2008. Hybrid Search: Effectively Combining Keywords and Semantic Searches. In *Proceedings of the 5th European Semantic Web Conference*.

D. Bikel, R. Schwartz, and R.M. Weischedel. 1999. An Algorithm that Learns What's in a Name. *Machine Learning, Special Issue on Natural Language Learning*, 34(1-3), Feb.

H. Cunningham, D. Maynard, and V. Tablan. 2000. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November.

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

H. Cunningham, K. Bontcheva, and Y. Li. 2005. Knowledge Management and Human Language: Crossing the Chasm. *Journal of Knowledge Management*, 9(5):108–131.

Hamish Cunningham, Valentin Tablan, Ian Roberts, Mark A. Greenwood, and Niraj Aswani. 2011. Information Extraction and Semantic Annotation for Multi-Paradigm Information Management. In Mihai Lupu, Katja Mayer, John Tait, and Anthony J. Trippe, editors, *Current Challenges in Patent Information Retrieval*, volume 29 of *The Information Retrieval Series*. Springer.

Hamish Cunningham. 2005. Information Extraction, Automatic. *Encyclopedia of Language and Linguistics, 2nd Edition*, pages 665–677.

Mark A. Greenwood, Valentin Tablan, and Diana Maynard. 2011. GATE Mímir: Answering Questions Google Can't. In *Proceedings of the 10th International Semantic Web Conference (ISWC2011)*, October.

Y. Li, K. Bontcheva, and H. Cunningham. 2005. SVM Based Learning System For Information Extraction. In M. Niranjan J. Winkler and N. Lawerence, editors, *Deterministic and Statistical Methods in Machine Learning*, LNAI 3635, pages 319–339. Springer Verlag.

D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. 2001. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria.

D. Maynard, H. Saggion, M. Yankova, K. Bontcheva, and W. Peters. 2007. Natural language technology for information integration in business intelligence. In W. Abramowicz, editor, *10th International Conference on Business Information Systems*, Poland, 25-27 April. `http://gate.ac.uk/sale/bis07/musing-bis07-final.pdf`.

Hans-Michael Müller, Eimear E Kenny, and Paul W Sternberg. 2004. Textpresso: An ontology-based information retrieval and extraction system for biological literature. *PLoS Biol*, 2(11):e309, 09.

Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, and Angel Kirilov. 2004. KIM – A Semantic Platform for Information Extraction and Retrieval. *Natural Language Engineering*, 10:375–392.

C. Rocha, D. Schwabe, and M. P. Aragao. 2004. A hybrid approach for searching in the semantic web. In *Proceedings of the World Wide Web Conference*.

Valentin Tablan, Ian Roberts, Hamish Cunningham, and Kalina Bontcheva. 2011. GATECloud.net: Cloud Infrastructure for Large-Scale, Open-Source Text Processing. In *UK e-Science All hands Meeting 2011*, York, UK, September.