ENABLING GOV 3.0 THROUGH SEMANTIC WEB, NATURAL LANGUAGE PROCESSING AND TEXT ANALYTICS

Islam A. Hassan, E-Government Group, Insight Centre for Data Analytics, National University of Ireland, Galway, Republic of Ireland
islam.hassan@insight-centre.org

Adegboyega Ojo, E-Government Group, Insight Centre for Data Analytics, National University of Ireland, Galway, Republic of Ireland
adegboyega.ojo@insight-centre.org

Abstract
The notion of “Government 3.0” (Gov3.0) is gradually emerging among policymakers as a label for next generation ICT-enabled technology innovation in government and successor to “Government 2.0” (social media or web 2.0 based) initiatives. The Gov3.0 phenomenon has been largely associated with all forms desirable attributes for future government institutions such as increased agility and innovation capacity. However, only few scholarly works have attempted to provide conceptualizations or analysis of this emerging phenomenon. This paper offers such conceptualization, by describing Government 3.0 as Semantic Web (SWEB) or Web 3.0 enabled innovation in government. While there are existing SWEB-based applications in the government domain, the diffusion of this family of applications in the government has been slow if not stagnated. We argue that the reason for this problem is largely due to the lack of tools and domain specific resources for automatic semantic annotation of existing government-related resources and contents on the web and social web. To address this challenge, we describe how text-analytics and Natural Language Processing tools could be used for information extraction and annotation of existing government resources on the web. In addition, we describe how the semantic resources generated from our web-scale automated annotation approach could be used to build two exemplar Gov 3.0 applications. Finally we discuss the challenges in developing this Gov3.0 infrastructure.

Keywords: Government 3.0, Ontology based, information extraction (IE), Natural Language Processing (NLP), Public Services, t-Government.

1 Introduction

With the adoption of the “Government 3.0” agenda by the Korean Government with the vision to create a “transparent, competent and service-oriented” government [Nam 2013], there is a growing interest in both the policy and academic arena to deconstruct what this concept may stand for in general. In the Korean framework, Gov3.0 is characterised by two high-level goals: 1) providing customized services tailored to various needs and demand, and 2) creating new jobs and re-boosting development. While the Korean Government is arguably the only country that has openly declared a “Government 3.0” agenda so far, practitioners in general have associated this label with the next stage of e-government evolution (i.e. after Government 2.0 and Web 2.0). Like Government 2.0 that is associated with Web 2.0-based innovation, Web 3.0 or the Semantic Web is expected to play a major role in enabling Gov3.0 [Nam 2013].

The emergence of the Semantic Web concept in 2001 marked an important stage in the Web’s evolution. As stated in (Tim Berners-Lee et al. 2001), Semantic Web it is “an extension of the current Web in which information is given well-defined meaning, better enabling computers
and people to work in cooperation.” This proposition has not yet been fully realized, and although many efforts have been made in this direction, much remains to be done. However, over the years, a number of semantic-web applications have emerged in the government space.

For instance in (Medjahed et al. 2003), an approach for managing the e-government services using web services and the ontologies to make e-government services machine interpretable and processable was described. In (Klieschewski 2003), two possible areas for using the SWEB in the e-government domain were presented including the markup of informational resources, in which all the informational resources to be made accessible through e-government applications are annotated using ontologies and interrelated to resources. A semantic approach enabling the publishing of statistical data gathered from different government agencies was presented in (Hoxha & Brahaj 2011) which included providing a semantic representation of the data based on an ontology, and then serialising the data as RDF triples following Linked Data principles. The work described methods for querying single or combined dataset and how to visualise the results. Similar work was reported in (Gheorghiu & Nicolescu 2011) where semantic web based mash-up application providing access to governmental data was offered as an interactive tool able to query certain SPARQL endpoints of interest using semantic web standards in order to visualise, analyse, and enhance the results.

Despite the above examples, the diffusion of this family of applications in the government has been slow if not stagnated. In our opinion, the reason for this problem is largely due to the lack of tools and domain specific resources for automatic semantic annotation of existing government-related resources and contents on the web and social web. Specifically, to progress towards the Semantic Web, there is a need to convert existing and new web contents to forms that can be understood by both humans and also understandable by machines. The semantic annotation (or markup) of Web documents is the first step towards adapting Web content to the Semantic Web. Providing the information elements that currently make up the Web with a well-defined meaning would, inter-alia improve its contextual search capabilities, increase interoperability between systems in ‘collaborative’ contexts and, when combined with Web services, ultimately enable near automatic composition of applications (Sheth et al. 2002) (Tse-Ming Tsai et al. 2003). Unfortunately, most Web contents remain unstructured (i.e. textual information) making this process difficult and cost of annotation high. Addressing this problem requires the availability of Natural Language and Text analytics tools to automatically extract domain-relevant information from textual data and creating domain-specific annotation resources based on the extracted information.

There are few past efforts such as (Misuraca et al. 2012) (Lin et al. 2009) that aimed to address this challenge by developing domain ontologies used along with the Information extraction and Natural Language Processing (NLP) tools to identify structured information from the raw text. Specifically, in (Alfonseca & Manandhar 2002), General Architecture for Text Engineering (GATE) components and NLP tools were used to extract crime information from police reports, newspaper articles, and victims’ and witnesses’ crime narratives automatically with the output presented as a meaningful summary for police investigators. These enabled the police investigators’ quick comprehension of crime incidents without having to read an entire report. In (Asahara & Matsumoto 2003), structural patterns expressed in terms of regular expressions combined with lexical conditions were used to detect the typologies of provisions contained in a normative document and extracted its related arguments.

A plausible explanation for the paucity of NLP applications in the government domain is the high manual engineering efforts required in developing domain-specific rules needed to meaningfully extract information of interest from text sources associated with traditional NLP methods. With the emergence of domain independent or open information extraction (IE) techniques, interest on how to exploit these tools as light-weight domain-specific IE tools are
growing. This paper describes one such effort aiming leverage a well-known “generic or open” entity and information extraction tools to identify relevant terms and relations associated with an e-government domain vocabulary related to public services.

Our proposed approach consists in using open information extraction (OIE) in combination with a generic named entity recognition (NER) tools to extract public service information from the different governmental websites. First, we extended the DBpedia Spotlight as a generic NER tool to support extraction our domain-specific terms. Second, we used domain-specific ontology to map the relations on the extracted triples from the OIE tool. Third, the extracted entities and relations were used in populating Public Service Ontology.

The paper is organized as follows: Section 2 summarizes the background and the different techniques for Information Extraction. Our ontology-driven Entity and Information Extraction approach is presented in Section 3. Section 4 presents two applications scenarios or use cases for the proposed approach. Section 5 presents the challenges associated with the implementation of our approach. Concluding remarks are presented in Section 6.

2 Background

2.1 Semantic Web

The desire to extend the capabilities of the Web for publishing of structured data is not new, and can be traced back to the earliest proposal for the World Wide Web (Tim Berners-Lee 1989) and subsequent papers on the topic (Barners-Lee 1992). Trends foreseen at the early stages of the Web’s existence included “Evolution of objects from being principally human-readable documents to contain more machine-oriented semantic information” (Berners-Lee 1992), which can be seen as the seed of an idea that became known as the Semantic Web. The vision of a Semantic Web has been interpreted in many different ways (Tim Berners-Lee et al. 2001; Marshall & Shipman 2003). However, despite this diversity in interpretation, the original goal of building a global Web of machine-readable data remains constant across the original literature on the subject. According to (Berners-Lee 1999) “The first step is putting data on the Web in a form that machines can naturally understand, or converting it to that form. This creates what I call a Semantic Web – a web of data that can be processed directly or indirectly by machines”.

The Semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. Its well-defined data semantics enable computer agents and humans to work in cooperation (Tim Berners-Lee et al. 2001). Recent efforts in the World Wide Web Consortium (W3C) to implement Semantic Web (Gruber 1995) have spurred interest in the use of ontologies for information modelling and knowledge representation. Ontologies provide shared domain models that are understandable to both humans and machines. They describe a set of concepts and relationships between them. Ontologies provide a controlled vocabulary of terms that can collectively provide an abstract view of the domain (Schreiber & Swick 2006; Uschold & Gruninger 2009). Such a shared understanding of the domain greatly facilitates querying of data and increases recall and precision. Semantic Web technologies and ontologies are being used to address data discovery, data interoperability, knowledge sharing and collaboration problems. Software agents can then be used to construct and provide dynamic services on the web.

Ontologies can be described in RDF (Resource Description Framework) [16, 17] which provides a flexible graph based model, used to describe and relate resources. An RDF document is an unordered collection of statements; each with a subject, predicate and object (triples). These statements describe properties of resources. Each resource and property can be
identified by a unique URI (Uniform Resource Identifier), which allows metadata about the resource to be merged from several sources. RDF has a formal specification and it is widely being used in a number of standards. It provides a common framework for expressing information, so it can be exchanged between applications without loss of meaning. RDF Schema (RDFS) adds taxonomies for classes and properties. It allows expressing classes and their relationships (subclass), and defining properties and associating them with classes. It facilitates inferencing on the data based on the hierarchical relationships.

**Fig.1 The Linking Open Data cloud diagram**

OWL (Web Ontology Language) [(Hayes & Ian n.d.; McGuinness & Harmelen n.d.) provides extensive vocabulary along with formal semantics and facilitates machine interpretability. OWL is much more expressive than RDF or RDFS, allowing us to build more knowledge into the ontology. For example, cardinality constraints can be imposed on the properties of an OWL class. OWL is designed as a specific language to define and describe classes and properties within ontology. It has many predefined built-in functionality. For example, an ontology can import other ontologies, committing to all of their classes, properties and constraints. There are properties for asserting or denying the equivalence of individuals and classes, providing a way to relate information expressed in one ontology to another. These features, along with many others, are important for supporting ontology reuse, mapping and interoperability. Also, OWL is a standard and is supported by the standard organization W3C. We have used the Web Ontology Language OWL to define the ontologies and RDF is used to describe the actual instances within the knowledge base. As the semantic web grows, more and more ontologies will be available in OWL. There will be a wide variety of development tools available for integrating the different OWL ontologies and doing intelligent reasoning. OWL provides 3 sub 5 languages with increasing levels of expressiveness and complexity. They are OWL Lite, OWL DL (Description Logics) and OWL Full. We have used mostly OWL Lite due to the reason that it is less complex than DL and Full. Also it suffices for our purpose. More tool support is currently available for OWL Lite than others. A comprehensive list of OWL Lite language constructs is available at (McGuinness & Harmelen n.d.).
By publishing Linked Data, numerous individuals and groups have contributed to the building of a Web of Data, which can lower the barrier to reuse, integration and application of data from multiple, distributed and heterogeneous sources. Over time, with Linked Data as a foundation, some of the more sophisticated proposals associated with the Semantic Web vision, such as intelligent agents, may become a reality (see Figure 1).

2.2 Information Extraction

To be able to manipulate vast amounts of unstructured data effectively, automated systems require efficient and accurate methods to derive information structures directly from text. The purpose of adding structure to otherwise flat text is to generate a partial representation of content in a form that can be effectively manipulated by the computer (Small & Medsker 2013). Specifically, most applications typically require representation that captures key events and the attributes of these events, including their role in analysing a corpus. Information extraction is the field that primarily deals with text.

Information Extraction addresses a variety of problems including: identifying relations from textual content (Embley et al. 1998; Buitelaar & Ramaka 2005) and automatic instantiation of ontologies and building knowledge bases tools (Alani et al. 2003). In domain specific context, information extraction could be used for obtaining information from specialized literature such as biomedical literature (Kamegai 2002).

Traditional methods on IE have focused on the use of supervised learning techniques such as hidden Markov models (Freitag & McCallum 1999; Skounakis et al. 2003), self-supervised methods (Etzioni et al. 2005) and rule learning (Soderland 1999). These techniques learn a language model or a set of rules from a set of hand-tagged training documents and then apply the model or rules to new texts. Models learned in this manner are effective on documents similar to the set of training documents, but extract quite poorly when applied to documents with a different genre or style. This process was expected to be simpler than manually creating patterns and rules by hand and therefore faster and more accurate than the pattern matching technique. BBN’s statistical language model used this approach for their MUC system which performed extremely well (Miller et al. 1998) at MUC and is currently at the core of their leading IE system (Identifinder). However, the effort to annotate training data for each new domain was found to be more challenging than expected (Small & Medsker 2013).

The semi-supervised learning (SSL) methods use smaller sets of annotated data than what is used in supervised learning (SL) methods, and they are augmented with large amounts of unannotated data. The typical semi-supervised learning (SSL) process is that annotations of some examples, named entities, events, and relations can be used to find more examples and thus more patterns from unannotated text. Unsupervised learning (UL) methods attempt to glean information automatically from the texts themselves, also called Open Information Extraction (Dalvi et al. 2012; Etzioni et al. 2008). As the name suggests, this approach to IE does not require a pre-specified vocabulary or ontology (Fader et al. 2011), nor does it necessarily need training data or rules. Usually, this approach facilitates domain independent extraction of assertions. This paradigm is often considered to be liberal in a sense that essentially any texts between two entities’ mentions are considered as a relation. Obviously, this implicitly promotes the recall while accepting a level of noise in the extraction results (Lin et al. 2009).

2.3 Named entity recognition and disambiguation

The Named Entity (NE) recognition and disambiguation problem has been addressed in different research fields and they agreed on the definition of a named entity, which is an information unit described by the name of a person or an organization, a location, a brand, a
product, a numeric expression including time, date, money and percent found in a sentence (Grishman 1996). The Supervised Learning (SL) techniques used a large dataset manually labelled. In the SL field, a human being usually trains positive and negative examples so that the algorithm computes classification patterns. SL techniques exploit Hidden Markov Models (HMM) (Bikel et al. 1997), Decision Trees (Borthwick & Sterling 1998), Maximum Entropy Models (Hoffart et al. 2011), Support Vector Machines (SVM) (Asahara & Matsumoto 2003) and Conditional Random Fields (CRF) (McCallum & Li 2003). The common goal of these approaches is to recognize relevant key-phrases and to classify them in a fixed taxonomy. The challenges with SL approaches are the unavailability of such labelled resources and the prohibitive cost of creating examples. Semi-Supervised Learning (SSL) and Unsupervised Learning (UL) approaches attempt to solve this problem by either providing a small initial set of labelled data to train and seed the system (Ji & Grishman 2006), or by resolving the extraction problem as a clustering one. Other unsupervised methods may rely on lexical resources (e.g. WordNet), lexical patterns and statistics computed on large annotated corpus (Alfonseca & Manandhar 2002).

The NER task is strongly dependent on the knowledge base used to train the NE extraction algorithm, for example by leveraging on the use of DBpedia (Bizer et al. 2009), Freebase3 and YAGO (Suchanek et al. 2007)ontologies. In addition to detect a NE and its type, efforts have been spent to develop methods for disambiguating information unit with a URI. Disambiguation is one of the key challenges in this scenario and its foundation stands on the fact that terms taken in isolation are naturally ambiguous. These methods generally try to find in the surrounding text some clues for contextualizing the ambiguous term and refine its intended meaning. Therefore, a NE extraction workflow consists in analysing some input content for detecting named entities, assigning them a type weighted by a confidence score and by providing a list of URIs for disambiguation. Initially, the Web mining community has harnessed Wikipedia as the linking hub where entities were mapped (Hoffart et al. 2011; Kulkarni et al. 2009). A natural evolution of this approach, mainly driven by the Semantic Web community, consists in disambiguating named entities with data from the LOD cloud. In (Nadeau & Sekine 2007), the authors proposed an approach to avoid named entity ambiguity using the DBpedia dataset.

2.4 Automatic Annotation

Annotation is the process of adding additional information to any other information such as information in a book, document, online record, video etc. In linguistics, annotation is the process of adding additional linguistics information such as morphological, syntactic, semantic etc. to available linguistics forms to make these forms more descriptive. Annotation of textual document is to identify the concept with the help of domain ontologies (domain because semantic or concept analysis is mostly domain specific i.e. one thing defined for one domain may have difference sense or concept in another). For this purpose, a robust text analysis technique is required which will be composed of identification of object in a text, identification of relationship between these objects and analysis on how these objects and their relationships combine to form a concept. A lot of work has been done in the field of semantic annotation using ontology as main guide, but there is still no complete automatic semantic annotation tool available that has a good accuracy due to inherent ambiguity in Natural Languages. There are many tools available for semantic annotation of textual document like GATE (Cunningham et al. 2001), KIM (Alani et al. 2003), Melita (Ciravegna et al. 2002), MnM (Vargas-Vera & Motta 2002), Magpie (Domigue & Dzbor 2004) etc. but none of the tools are totally automatic. Furthermore, these systems perform annotation on words and terminologies to identify real world objects and their relationship in the text. None of them provide annotation above word level (Ahmed 2009).
3 Approach

As shown in Section 2, most of the current NER and IE tools provide only limited, generic capabilities for information extraction and do not support domain specific content processing. We describe here how we approach the problem, (see Figure 2). First we present the abstract Framework Architecture and with focus on the semantic annotation and enrichment component. In our work, we propose to use the Open Information Extraction (OIE) tools to extract generic triples (statements) out of the raw text. Second, these triples will be processed by DBpedia Spotlight, which will serve as a named entity recognition (NER) tool to extract and annotate the public services documents or corpus (see figure [3]). Finally we populate an ontology of public services, by mapping both the entity and the relations to the ontology to produce the RDF that will be used as domain specific semantic resources to be harnessed in our two Gov 3.0 applications examples.

![Figure 2: The Framework architecture.](image)

The main components of the proposed solution architecture include:

**The Domain specific language resource** – We started with building the domain specific language resource, this language resource will be used in extending the NER tool to recognize the public service terms. We built a seed corpus and worked on annotating it and semi-automatically transforming it from unstructured to structured form.

**Domain Specific Ontology** - This ontology is needed for building the graph by mapping the correct relations to the extracted entities. We have used a conceptual model for the public service, based on the public service ontology, being developed under the auspices of the ISA – the Interoperability Solutions for European Public Administrations programme Figure [4].

**Named Entity Recognition (NER)** - As described in the background section, the NER tool is the component that is responsible for recognizing and disambiguating the extracted entity by the open information extraction tuples, we tried to use DBpedia spotlight for doing this task. DBpedia Spotlight is a tool for automatically annotating mentions of DBpedia resources in text. Thus, it provides a generic domain independent NER tool that needs to be extended by domain specific language resource in order to recognize the public service terms, there are two possible ways to extend the DBpedia spotlight described below:

- Using Wikipedia Infoboxes and DBpedia extraction engine.
- Using N-Triples files for building indexes and Spotters training.
Open Information Extraction (OIE) - The open information extraction is mainly used for extracting the tuples automatically out of the raw text. It is a generic and domain independent. We explored three of the famous state of the art OIE tools ReVerb, OLLIE, and ClausIE during our literature review.

Tuples – This constitute the major outputs from the extracted information produced by the OIE tool. The tuples are represented in the traditional Subject, Predicate Object format (S, P, O) format.

Entities, Relations mapping and ontology population - This component is responsible for populating the ontology (public service ontology in our case) with the recognized entities by the NER, mapping the tuples into domain-specific relations based on the ontology, and producing the Extracted information in RDF/XML format.

Fig.3 The Information Extraction System architecture.

The process for applying the framework is as follows:
1. Governmental websites and document are used as input for the OIE tool.
2. The OIE tool automatically extracts the information from the text in a form of tuples.
3. Transform tuples into domain specific entities and relations:
   a. Use the extended NER to recognize and disambiguate entities out of the tuples.
   b. Map the tuples independent predicates into domain-specific relations based on the ontology (i.e. the public service ontology).
   c. Populate the ontology with the entities (3.a) and the (3.b).

The output of this process is the domain-specific extracted information in RDF/XML format.

4 Use case – Automatic Annotation of Public Service Descriptions

The goal in this section is to present two concrete applications of our solution. Before presenting these applications, we discuss in more details how the domain-specific language resource for public services domain annotation was created.
We used the Core Public Service Ontology for documenting services in Europe as the basis for creating our language. Using this language resource used for the information extraction will help in populating the public service ontology using the information extracted from other governmental public service documents like PDF documents or web documents.

![UML diagram for the Core Public Service Vocabulary. All classes and properties are in the CPSV namespace unless otherwise indicated.](image)

Extracting information and representing them in a linked structured data form will open up opportunities for useful applications that can leverage these information, for instance improving citizens experience in exploration of public service information on the web. Specifically, Citizens can enjoy new applications, which are built on the populated ontology to get better access for the public services. At the same time, governments could use it for carrying out sentiment analysis or public opinions analysis on different public services mentioned on the social media or web. These kinds of applications enhance the delivery of the public services and enable better understanding the citizen’s needs. As illustration, we show in Figure 5 the extraction of public service information from a governmental website. The figure shows important attributes annotated in the text like the public service name, the different inputs and we can use the page URL as the channel of delivering this service. In Appendix B we can see the RDF/XML that produced by the purposed framework as an output of the information extraction processes and after populating the result on the public service vocabulary “ontology”.

Islam A. Hassan and Adegoke Ojo

Enabling Gov 3.0 through Semantic Web, Natural Language Processing and Text Analytics
Next we describe the two applications involving: 1) the intelligent navigation and exploration of public services on government portals and 2) monitoring of public services mentions on social media, described in details in Sections 4.1 and 4.2.

4.1 Public Service intelligent navigator

As indicated above, the semantic markup of Web documents is the first step towards adapting Web content to the Semantic Web. The embedded semantics approaches like Microformats and RDFa create a link between the Semantic Web and (social) Web containing the governmental pages (in our scenario). This link helps in consuming the semantic data from third party applications such as crawlers, or search engines. However, until now the human-readable presentation of these semantic meta-information has been set aside. From the citizen’s point of view, the actual interpretation of a rendered document relies on:

1. The contextual knowledge provided by the hyperlinks contained in a document;
2. The background knowledge of the citizen that helps him to bind unlinked information to known concepts. Citizens need to use the document’s internal and external semantics to lead them to a better understanding of the public service documents content. For making it possible, the document's representation must be augmented with making explicit this external knowledge in a human readable way.

The public service intelligent navigator is conceived as a plugin for the Internet browsers that enables citizens to consume the Public Service Linked Data generated from automatic annotation of public service pages. To demonstrate the use of this tool, consider a citizen searching for information about the driving test service in Ireland. After locating the government webpage describing the sought service, the public service intelligent navigator plugin in the browser will provide notification telling the user other complementary information about the service like driving test service (see Figure 6). Clicking over the plugin icon will popup three tabs containing information related to this service like, the related services, required for and notes.

![Fig.5 Information Extraction of a public service information from it’s website.](image-url)
In the first tab the user will find information about the related services as in Figure [6] services like “Apply for Driver Theory Test”, “Find a Driving Instructor”, etc. Under the “required for” tab the user will find all the services that require the output of the current service for as an input to get it. In the last tab “notes” users will find the some advice regarding taking the service that could be extracted from the social media as citizens now share their experience of using the public services frequently “sharing knowledge”, you also might find notes from the original website about the service under this tab too. Note that these tabs are generated based on the relations described in the public service ontology presented in Figure 4.

Information about any particular service could be retrieved from a triple data store or by using embedded semantics, with the information published in the website. In the first case, we crawl the governmental public service websites then use the extracted obtained information in populating the ontology and stored in as linked data in the data store. There are also many methods for embedding semantics as mentioned above. In the case of embedded semantic the public service intelligent browser will extract the embedded semantic content from a page by checking the document’s RDFa statements and citizens will be able to request more information about these statements by selecting any of them. When doing so, inferences from inside and outside the page are executed for that statement, a contextualized view of the obtained knowledge is presented and the related information within the document (statements with the same predicate) displayed too (see Appendix A).

4.2 Monitoring Social media

Social media produce high velocity stream of information on the web. Twitter for example produces millions of messages per day and hundreds tweets per second. The simplicity of using it motivates people around the world to produce textual contents at a rapid rate. This large volume calls for an automated interpretation to flexibly and quickly respond to shifts in sentiment or rising trends. This way one can almost directly respond on phenomena arising in society than traditionally and that will open for the citizens “unimagined opportunities to do more for themselves (Johnston & Hansen 2011).
Driven by rising citizen expectations and the need for government innovation, social media has become “a central component of e-government in a very short period of time” (Bertot et al. 2012). As social media exists almost in the entire world, the sentiment expressed by users of social media is written in different languages. The social media connect the entire world and thus people can much more easily influence each other and that could help in the interoperability.

The social media analytics using text analytics such as Information Extraction, named entity recognition tools to extract the public services information, combined with sentiment analysis tools allow us to determine citizens feedback about the different services (see figure 9). For example, tweets mentioning the “driving test public service” in Ireland are shown in the figure. The output of the social media analysis could be used again to help the other citizens in getting the service and also will help the government in understanding the citizen needs. Mentions or entities of interests could be automatically identified in tweets or from any social media sources such as Facebook pages, using the approach described in Section 3 with the extracted information property semantically annotated based on some ontology or vocabulary and stored in a knowledge base like an RDF store.

5 Challenges

We summarize here the two categories of challenges we confronted in the implementing this solution. The first relates to the difficulties in developing the domain-specific language resource and the second is linked to the choice of the Open Entity Extraction tool – the popular DBpedia Spotlight.

5.1 Language Resource Challenges:

Manual effort in creating the domain specific language resource - Creating a domain specific language resource for extending the named entity recognition still takes manual effort, albeit less than the effort spent in the other approaches where traditional information extraction systems is used. In our case we searched for the proper corpus that could be utilized as a valuable seed for the language resource, while working on the data transformation and mapping the documents to the proper ontology with the help of a domain expert was challenging.

Semantic Challenges - Identifying synonyms names, sentence relations, across different agencies in the same government, across government levels, across borders; In the case of the cross-border synonyms, there is a need to deal with the multi-lingual issues.
5.2 DBpedia Spotlight Challenges:

Data Quality - DBpedia is highly dependent on the of cross-ontology mappings. This affects the data quality of the extracted information and consequently the performance of DBpedia spotlight. The quality of the extracted data is increased by providing quality assurance tools for the Wikipedia community and to notify the authors about contradictions between the content of infoboxes contained in the different language versions of the article. There are also efforts on interlinking DBpedia with other knowledge bases such as Cyc and YAGO to apply semi-automatic consistency checks for Wikipedia content. Crowdsourcing quality assessment is also used over DBpedia to solve this challenge (Medjahed et al. 2003).

Resources Demand - Creating a spotlight model locally requires setting up Apache Hadoop and Apache Pig, for hadoop-based indexing process. This requires an expensive computational resource to be able to handle the English Wikipedia dump. These resources are not available in most government IT ecosystem currently.

Poor Documentation - The third challenge is about the completeness of the documentation. The documentation does not cover the different components of the DBpedia spotlight that. Consequently, time is wasted in figuring out a solution for most of the problems faced while working with DBpedia spotlight. However, the provided mailing list is a valuable source for seeking solutions.

6 Conclusion

The work conceptualizes the notion of Government 3.0 as a Web3.0-enabled innovation in the government arena. Although the usage of Gov3.0 transcends technological issues (like in the Korea Government case), we have sought to focus on the technical aspect of Gov3.0 to highlight the enabling role that NLP and text analytics plays in the generation of semantic contents on the semantic web. In fact, we attribute the apparent stagnation in semantic web adoption in government and arguably other domains, to the lack of NLP tools that could automatically generate semantic annotations guided by relevant ontologies. However, we have also shown through our approach that successful application of NLP tools is contingent on the availability of domain specific language resources and domain adaptation (or specific) frameworks. We have also shown that an effective Gov 3.0 application should enable intelligent interactions between citizens and government online resources. In addition to addressing the challenges outlined in Section5, our future work will focus on developing government-specific language resources and relevant vocabularies to enable the processing and annotation of different types of government information on the web. Furthermore, while we have employed DBpedia spotlight as our NER tool, our recent experiments show that, a detailed comparative analysis of domain-specific NER approaches may be beneficial to improve both the accuracy and recall of the extracted information. A major implication of this is work is the availability of a concrete language resource for government public services domain that could be freely exploited by other researchers to build Gov 3.0 applications or solutions. Another major side effect of the work is the availability of templates for describing government services in Wikipedia. Consequently, the open community could define and co-create public services descriptions on Wikipedia. This will significantly impact on the quality of annotations obtained through DBpedia Spotlight.

7 References


Buitelaar, P. & Ramaka, S., 2005. Unsupervised ontology-based semantic tagging for knowledge markup. Workshop on learning in web search at 22nd .... Available at: http://cosco.hiit.fi/search/learninginsearch05/ICMLW4-LWS.pdf#page=34.


McGuinness, D.L. & Harmelen, F. van, OWL Web Ontology Language Overview. Available at: http://www.w3.org/TR/owl-features/.


Skounakis, M., Craven, M. & Ray, S., 2003. Hierarchical hidden markov models for information extraction. IJCAI. Available at: http://papercut.googlecode.com/hg-history/98464ac0efb47c55159b313c89b0b305ba1d83f9/PaperCutTesting/targetPDF/success/hhmm.pdf.


Appendices

Appendix A, A Public Service RDFa snippet.

<!--snip-->

Name Surname (First Author) or First Author and Second Authors or First Author et al.

Title of the Paper

<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
    <rdf:Description rdf:about="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest">
        <rdf:type rdf:resource="http://purl.org/vocab/cpsv#PublicService"/>
        <dct:title xml:lang="en">Apply for Driving Test</dct:title>
        <dct:description xml:lang="en">Driver testing in Ireland is carried out directly by the Road Safety Authority to a standard that complies with the EU Directive on Driving Licences. You can now apply and pay for your driving test online. You will need a credit card to do this (Visa or Mastercard) or a debit card (Laser) plus a valid email address. Alternatively, you can download a driving test application form, or obtain a copy from any motor tax office.</dct:description>
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/LearnerPermit"/>
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/MandatoryLessons"/>
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/DriverNumber"/>
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/PPSNumber"/>
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/PaymentCard"/>
        <cpsv:hasChannel rdf:resource="http://www.rsa.ie/RSA/Learner-Drivers/The-Driving-Test/Apply-online/"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://www.rsa.ie/org">
        <cpsv:provides rdf:resource="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest"/>
        <cpsv:playsRole rdf:resource="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest">
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/LearnerPermit"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest">
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/DriverNumber"/>
        <dct:title xml:lang="en">Driver Number</dct:title>
    </rdf:Description>
    <rdf:Description rdf:about="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest">
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/PPSNumber"/>
        <dct:title xml:lang="en">PPS Number</dct:title>
    </rdf:Description>
    <rdf:Description rdf:about="http://cpsv.testproject.eu/id/irl/PublicService/DrivingTest">
        <cpsv:hasInput rdf:resource="http://cpsv.testproject.eu/id/irl/Input/PaymentCard"/>
    </rdf:Description>
</rdf:RDF>