Controlled Natural Language for Semantic Annotation

Brian Davis and Pradeep Varma and Siegfried Handschuh and Laura Dragan[1] and Hamish Cunningham[2]

[1] Digital Enterprise Research Institute, National University of Ireland, Galway
{brian.davis, pradeep.varma, siegfried.handschuh, laura.dragan }@deri.org
[2] Sheffield NLP Group, University of Sheffield
hamish@dcs.shef.ac.uk

Abstract. *Sovereign* is a novel collection of resources for authoring(1), annotating(2) and accessing(3) knowledge on the Social Semantic Desktop. With respect to (2), the Sovereign Semantic Annotator allows the non-expert user in a novel way to semi-automatically author and annotate meeting minutes and status reports *simultaneously* using Controlled Natural Language(CNL). The metadata is captured as knowledge on the Social Semantic Desktop for later aggregation and access. The annotator is based on Controlled Language for Information Extraction (CLIE) technology. Furthermore, it is available for as a plugin for a semantic note-taking application for the Social Semantic Desktop. We intend to present a working prototype of the Sovereign annotator for the Social Semantic Desktop at ESWC.

1 Introduction and Background

Semantic technologies are difficult to access for non-expert users wishing to author(1), annotate(2) and/or access(3) knowledge. Our research investigates how Human Language Technology (HLT) interfaces; specifically Controlled Natural Languages(CNL)[3] and applied Natural Language Generation(NLG) can provide a user friendly means for non-expert users and organisations seeking to exploit Semantic Web technologies. *Sovereign*[4] is a novel collection of resources for authoring(1), annotating(2) and accessing(3) knowledge on the Social Semantic Desktop. This demonstration focuses on the annotation resource(2) in Sovereign, however we will also discuss briefly resources (1) and (3) since all three are interrelated.

With respect to **authoring(1)** - specifically ontology authoring, few ontology editing tools are aimed at non expert users, in particular users wishing to create simple structures without delving into the intricacies

[3] CNLs are "subsets of natural language whose grammars and dictionaries have been restricted in order to reduce or eliminate both ambiguity and complexity". See http://www.ics.mq.edu.au/~rolfs/controlled-natural-languages/

[4] Semantic annOtation VErbal REsources for ExtractIon and Generation of kNowledge

of knowledge representation languages. The CLIE (Controlled Language for Information Extraction) approach however allows users to create and edit ontologies quite simply by using a restricted version of the English language. This 'controlled natural language' is based on an open vocabulary and a restricted set of grammatical constructs. Sentences written in this language unambiguously map into a number of knowledge representation formats including OWL and RDF-S). Previous work has used CLIE to generate ontologies from the input CNL called CLOnE (Controlled Language for Ontology Engineering)[1]. The reverse of the process involves the generation of CLOnE from an existing ontology using NLG, specifically shallow NLG. The NL generator and the authoring process both combine to form a RoundTrip Ontology Authoring (ROA) environment: one can start with an existing or empty ontology, create an ontology using CLIE, reproduce CLOnE from the newly using the NL generator, modify or edit the text as required and subsequently parse the text back into the ontology using the CLIE environment. The process can be repeated as necessary until the required result is obtained.
With regard to **knowledge access(3)**, NLG can act as a human-readable window into the otherwise formally structured database. Rather than summarising the contents of an ontology for the purposes of modification or quality assessment as is specified above in the context of ROA(1), the use case of NLG is to provide a user friendly means of presenting semantically annotated knowledge captured within the Social Semantic Desktop in a human readable form and furthermore at the appropriate level of detail to the non-expert user, based on his/her specific queries.

1.1 Semantic Annotation

Concerning **annotation(2)**, richly interlinked, machine-understandable data constitute the basis for the Semantic Web, and by extension the Social Semantic Desktop[2]. Manual semantic annotation is a complex and arduous task both time-consuming and costly often requiring specialist annotators. (Semi)-automatic annotation tools attempt to ease this process by detecting instances of classes within text and relationships between classes, however their usage often requires knowledge of Natural Language Processing(NLP) and/or formal ontological descriptions. This challenges researchers to develop user-friendly annotation environments within the knowledge acquisition process. CNLs offer an incentive to the novice user to annotate, while simultaneously authoring, his/her respective documents in a user-friendly manner, but simultaneously shielding him/her from the underlying complex knowledge representation formalisms. A natural overlap exists between tools used for both ontology creation and semantic annotation. However, there is a subtle difference between both processes. "Semantic annotation is described as a process, as well as the outcome of the process. Hence it describes i) the process of addition of semantic data or metadata to the content given an agreed ontology and ii) it describes the semantic data or metadata itself as a result of this process"[3]. Of particular importance here is the notion of the addition or association of semantic data or metadata to *content*.

2 A Use Case for Controlled Natural Language for Semantic Annotation

In our scenario Nepomuk-KDE[5] - the KDE instance of the Social Semantic Desktop serves as the platform on which to build our tools and conduct our experiments. Firstly, however with respect to **annotation(2)**, CNLs cannot offer a panacea for semi-automatic annotation since it is unrealistic to expect users to annotate every textual resource using CNL, however there are certain use-cases where CNLs can offer an attractive alternative as a means for semi-automatic semantic annotation, particular in contexts, where controlled vocabulary or terminology is implicit such as health care patient records or business vocabulary. Our use case focuses on administrative tasks such taking minutes during a project team meeting and weekly status reports. Very often such note taking tasks can be repetitive and boring. In our scenario the user is a member of a research group which is part of an integrated EU research project. Based on pre-defined templates, the user *simultaneously authors and annotates* his/her meeting minutes or status reports in CNL, using a semantic note taking tool - SemNotes[6], which is an application available for Nepomuk-KDE. The newly created metadata is thus available for immediate use, inclusive of both querying and aggregation, whereby the retrieved RDF triples can be passed to a **Natural Language Generator(3)** to produce tailored textual reports and summaries. In addition, wrt **authoring(1)**, ROA supports the ontology authoring process to create a common vocabulary on which to base annotation and knowledge capture and subsequent text generation. The authoring, annotation and NLG functionalities, are made available as plugins to SemNotes. These three plugins are collectively called **Sovereign** - *Semantic annOtation VErbal REsources for ExtractIon and Generation of kNowledge.*

3 Implementation

As mentioned earlier, the Sovereign CNL annotator is available as a plugin to SemNotes. Furthermore, the CNL is realised within a semantic note. The CNL is anchored to existing semi-structured data such as a *AgendaTitle*, *Scribe* or *ActionItem* based on a predefined meeting minutes or status report template. The annotator is based on CLIE. The CNL itself is very similar to the CLOnE language, with some modifications. The annotator architecture contains a standard GATE pipeline[7](see Figure 1) consisting of the following language and processing resources: The GATE English tokeniser, the Hepple part-of-speech tagger, a morphological analyser, a gazetteer list component for recognising useful key-phrases, such as structured elements from the templates and reserved phrases in the controlled language. Any sentences preceded by a `Comment:` element are considered candidates for controlled language

[5] http://nepomuk.kde.org/
[6] http://smile.deri.ie/projects/semn
[7] General Architecture for Text Engineering, See http://gate.ac.uk/

parsing. Any remaining tokens from the CNL sentence which are not recognised as reserved CNL key-phrases are used as names to generate ontological objects(See Figure 2). This is followed by a standard Named Entity(NE) transducer in order to recognise useful NEs, a `preprocessing` JAPE[8] finite state transducer(FST) for identifying quoted strings, chunking Noun Phrases(NPs) and additional preprocessing. A second gazetteer list look up is applied to identify trigger phrases associated with NEs which intersect with quoted and unquoted NP annotation spans. Additional feature values are then added to the NP chunks to indicate the appropriate class to link an NP chunk as an instance to. The last FST parses the CNL from the text and generates the metadata. The current tool is bootstrapped via the Nepomuk Core Ontologies[9] and currently the application creates/populates a meeting minutes/status report ontology, which references the users Personal Information Model Ontology(PIMO) [10] via the GATE Ontology API. We have also modified the code to write directly to Nepomuk KDE RDF store.

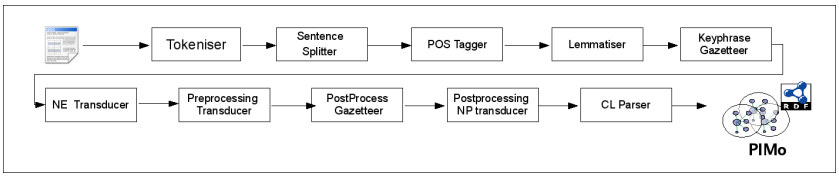

Fig. 1. The CNL Semantic Annotator pipeline

4 The Demo

Our demonstration will provide the audience with an opportunity to obtain live, hands-on experience using the **Sovereign CNL Annotator**. The demo will consist of a running Nepomuk KDE installation as a fictitious user (one of the personas from the NEPOMUK project[11]). The demo will be supported by a dataset consisting of prepopulated PIMO instances of the fictitious user amongst others as well as a dataset based on a corpus of **real-world** status-reports and meeting-minutes which has been collected over the three year duration of the Nepomuk Project. A demonstration of the user authoring/annotating meeting minutes and a status report based on the screenshot in Figure 2 will be provided in order to showcase what can achieved with the CNL annotator. We will

[8] Java Annotations Pattern Engine
[9] http://www.semanticdesktop.org/ontologies/
[10] http://www.semanticdesktop.org/ontologies/2007/11/01/pimo/
[11] http://nepomuk.semanticdesktop.org/

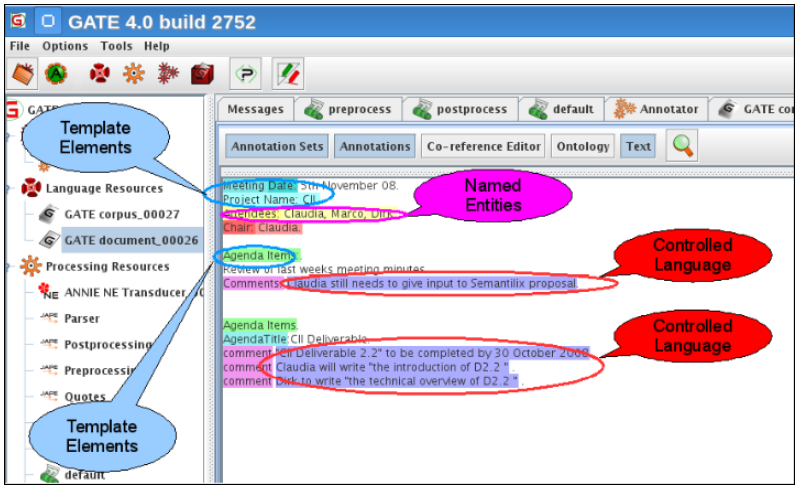

Fig. 2. CNL Annotator visualised in GATE

also provide a small CNL reference guide. Hence visitors will be invited to write CNL as well as experiment with and test the annotator as they see fit.

Acknowledgements

The work presented in this paper was supported (in part) by the Líon project supported by Science Foundation Ireland under Grant No. SFI /02/CE1/I131 and (in part) by the European project NEPOMUK No FP6-027705.

References

1. Brian Davis, Ahmad Ali Iqbal, Adam Funk, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, and Siegfried Handschuh. Roundtrip ontology authoring. In Amit P. Sheth, Steffen Staab, Mike Dean, Massimo Paolucci, Diana Maynard, Timothy W. Finin, and Krishnaprasad Thirunarayan, editors, *International Semantic Web Conference*, volume 5318 of *Lecture Notes in Computer Science*, pages 50–65. Springer, 2008.
2. S. Decker. The social semantic desktop: Next generation collaboration infrastructure. *Information Services and Use*, 26(2), 2006.
3. Siegfried Handschuh. *Creating Ontology-based Metadata by Annotation for the Semantic Web*. PhD thesis, 2005.