Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies

Kalina Bontcheva

Department of Computer Science, University of Sheffield
211 Portobello St, Sheffield, UK S1 4DP
kalina@dcs.shef.ac.uk

Abstract

This paper describes reusable, open-source tools for creation, maintenance, storage, and access of Language Resources (LR) needed for generating natural language texts from ontologies. One advantage of these tools is that they provide a user-friendly interface for NLG LR manipulation. They also provide unified models for accessing NLG lexicons and mappings between lexicons and ontologies.

1. Introduction

Natural Language Generation (NLG) techniques aim at producing natural language text, tailored to the presentational context and the target reader starting from structured data, typically held in a knowledge base (Reiter and Dale, 2000). An increasing number of applications are now using ontologies to represent and reason with formal knowledge, mainly driven by new developments in the area of the Semantic Web (Fensel et al., 2002). Therefore, a new challenge for NLG is to generate texts from ontologies and an important part of it is development of tools and infrastructures to support easy adaptability of NLG components to new ontologies.

This paper describes a set of open-source tools for creation, maintenance, and storage of Language Resources (LRs) for language generation from ontologies. Typically an NLG system uses LRs like lexicons, grammars, and ontologies, in order to generate text. This paper concentrates on reusable, user-friendly tools for NLG lexicons and ontologies, which aim at lowering the overhead of storing, maintaining, and accessing those LRs.

This work was carried out as part of the e-science project MIAKT[1], which aims at developing Grid enabled knowledge services for collaborative problem solving in medical informatics. In particular, the domain in focus is Triple Assessment in symptomatic focal breast disease.

The role of NLG in the project is to generate automatically textual descriptions from the semantic information associated with each case - patient information, medical procedures, x-rays, mammograms, etc. The majority of semantic information is encoded in the domain ontology, which is a formal description of the breast cancer domain (Hu et al., 2003) and is encoded in DAML+OIL (Horrocks and van Harmelen, 2001). In addition, each case has a case-specific, i.e., instance knowledge, which is encoded in RDF (Lassila and Swick, 1999) and specifies information about this particular case, e.g., which medical procedures were undertaken, sizes and locations of lesions, diagnosis.

The NLG tools were developed to be part of GATE[2] (a General Architecture for Text Engineering), which is a well-established infrastructure for customisation and development of NLP components. In brief, GATE (Cunningham et al., 2002; Maynard et al., 2002) is a robust and scalable infrastructure for NLP, which allows users to focus on the language processing tasks, while mundane issues like data storage, format analysis, and data visualisation are handled by GATE itself. In addition, GATE has a generic model for representing ontologies which was the corner stone of this work.

2. Overview of GATE's Ontology API

As evident from our experience with MIAKT and previous work on language generation from ontologies (e.g., (Wilcock and Jokinen, 2003; Wilcock, 2003)), NLG systems need to deal with the different formats in which ontologies can be represented - DAML+OIL, OWL, RDF. In order to avoid the cost of having to parse and represent ontologies in each of these formats in each NLG application, we used GATE's open-source tools that can parse these formats and convert them into a common object-oriented model of ontologies with a unified API (Application Programming Interface) (Bontcheva et al., 2003). GATE also provides a graphical user interface to enable browsing and editing of the ontologies, based on the common model, independent of their original format (see the rightmost pane of Figure 2).

This approach has well-proven benefits, because it enables each application to use this format-independent model when dealing with ontologies, thus making the application immune to changes in the underlining ontology formats. If a new format needs to be supported, the application can automatically start using ontologies in this format, by simply including the correct tool that converts the format into the common model. From a language engineer's perspective the advantage is that they only need to learn one API and model, rather than having to learn many different and rather idiosyncratic ontology formats.

Since OWL (Bechhofer et al., 2003), DAML-OIL (Horrocks and van Harmelen, 2001) and RDF(S) (Lassila and

[1]Project Web site: http://www.aktors.org/miakt. MIAKT is supported by the UK Engineering and Physical Sciences Research Council as part of the MIAKT project (grant GR/R85150/01), which involves the University of Southampton, University of Sheffield, the Open University, University of Oxford, and King's College London.

[2]GATE and its IE tools are freely available, under the GNU Library License, from http://gate.ac.uk.

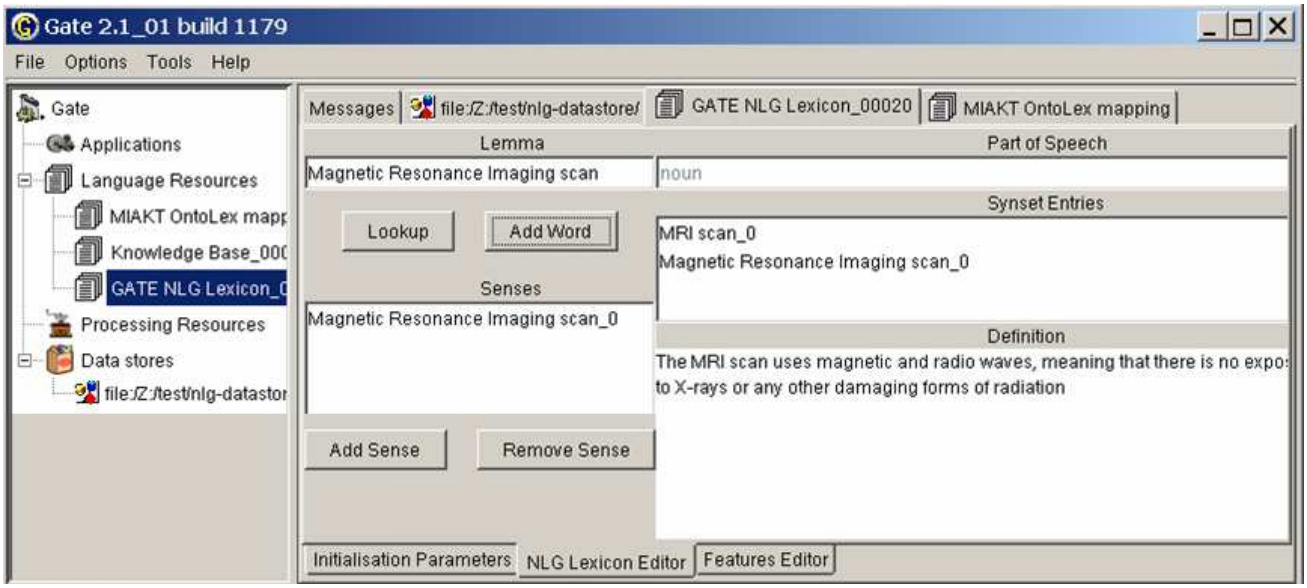

Figure 1: The MIAKT NLG Lexicon

Swick, 1999) have different expressive powers, GATE's ontology model consists of a class hierarchy with growing level of expressivity. At the top is a taxonomy class which is capable of representing taxonomies of concepts, instances, and inheritance between them. Multiple inheritance is not supported.

At the next level is an ontology class which can represent also properties, i.e., relate concepts to other concepts or instances. Properties can have cardinality restrictions and be symmetric and/or transitive. There are also methods providing access to their sub- and super-properties and inverse properties. The property model distinguishes between object (relating two concepts) and datatype properties (relating a concept and a datatype such as string or number).

The expressivity of this ontology model is aimed at being equivalent to OWL Lite. In the case of a DAML-OIL ontology, GATE uses a sub-set of Jena's API to read in the model and populate the GATE ontology classes. Any features outside the GATE model are ignored. When reading RDFS, which is less expressive than OWL Lite, GATE only instantiates the information provided by the RDFS model, i.e., classes, instances, and properties between them, but without cardinality restrictions, etc. If the API is used to access one of these unsupported features then the API returns empty values.

From MIAKT's perspective RDFS does not provide sufficiently expressive power, while both DAML-OIL and OWL Lite do. In our experiments we used Jena and DAML-OIL, because at the time OWL Lite was not finalised and there were no implemented tools for it.

3. The MIAKT NLG Lexicon and Tools

An important part of every NLG system is the lexicon. The need for a specialised NLG lexicon and associated editing tools comes from the fact that NLG lexicons often contain entire phrases and typically need to be extended with domain terminology, i.e., editing as well as visualisation tools are needed. More importantly, the tools need to be usable both by language engineers and knowledge engineers, i.e., have an intuitive interface and require minimal linguistic expertise.

Since NLG systems typically operate in specialised domains, only part of these lexicons can be acquired from existing general-purpose lexicons, such as Wordnet (Miller, 1995). The domain-specific terms are typically acquired from corpora, terminological dictionaries or are entered manually. In order to lower the overhead of NLG lexicon development we created graphical tools for editing, storage, and maintenance of NLG lexicons, combined with a model which connects lexical entries to concepts and instances in the ontology. These tools are open source and are part of GATE (Cunningham et al., 2002) - General Architecture for Text Engineering (see Figure 1). GATE also provides access to existing general-purpose lexicons such as WordNet and thus enables their use in NLG applications.

The structure of the NLG lexicons is similar to that of WordNet. Each lexical entry has a lemma, sense number, and syntactic information associated with it (e.g., part of speech, plural form). Each lexical entry also belongs to a *synonym set* or *synset*, which groups together all word senses which are synonymous. For example, as shown in Figure 1, the lemma "Magnetic Resonance Imaging scan" has one sense, its part of speech is noun, and it belongs to the synset containing also the first sense of the "MRI scan" lemma. Each synset also has a definition, which is shown in order to help the user when choosing the relevant synset for new word senses.

When the user adds a new lemma to the lexicon, it needs to be assigned to an existing synset. The editor also provides functionality for creating a new synset with part of speech and definition.

The advantage of a synset-based lexicon is that while there can be a one-to-one mapping between concepts and instances in the ontology and synsets, the generator can still use different lexicalisations by choosing them among those listed in the synset (e.g., MRI or Magnetic Resonance

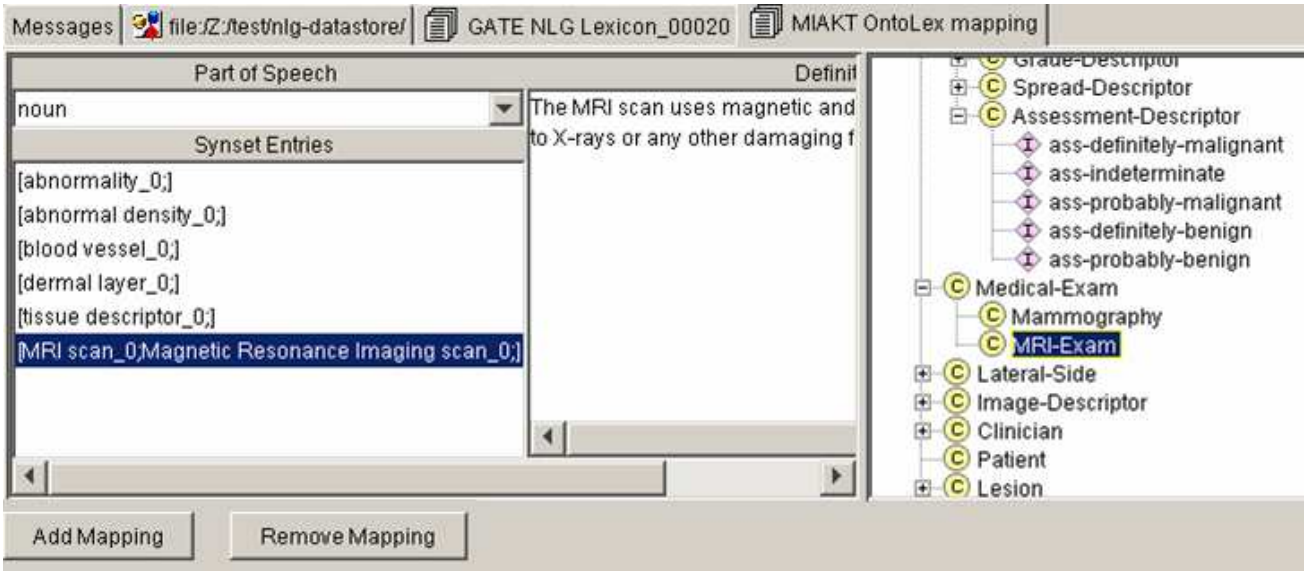

Figure 2: Mapping lexical entries to concepts and instances

Imaging). In other words, synsets effectively correspond to concepts or instances in the ontology and their entries specify possible lexicalisations of these concepts/instances in natural language.

At present, the MIAKT NLG lexicon encodes only synonymy, while other non-lexical relations present in WordNet like hypernymy and hyponymy (i.e., superclass and subclass relations) are instead derived from the ontology, using the mapping between the synsets and concepts/instances. The reason behind this architectural choice comes from the fact that ontology-based generators ultimately need to use the ontology as the knowledge source. In this framework, the role of the lexicon is to provide lexicalisations for the ontology classes and instances.

The mapping between synsets in the lexicon and concepts and instances in the ontology is done using a model, called *ontolex mapping*. This model supports polysemy and synonymy by allowing the same lexical entry to be mapped to different concepts/instances (polysemy) and many lexical entries to be mapped to the same concept/instance (synonymy). The model also has the corresponding user interface where the mappings can be edited, stored, and browsed (see Figure 2).

In the MIAKT project we used the NLG lexicon tools to create a lexicon of 320 terms in the domain of breast cancer and map them to the 76 concepts and 153 instances in the MIAKT ontology. These terms were collected manually from the BIRADS lexicon of mammography terms[3] and NHS documents[4], then verified and enriched them manually with synonyms from online papers, Medline abstracts, and the UMLS thesaurus (NLM,).

4. Complexity and Generality

The lexicon model was kept as generic as possible by making it incorporate only minimal lexical information. Additional, generator-specific information can be stored in a hash table, where values can be retrieved by their name. Since these are generator specific, the current lexicon user interface does not support editing of this information, although it can be accessed and modified programmatically.

On the other hand, the NLG lexicon is based on synonym sets, so generators which subscribe to a different model of synonymy might be able to access GATE-based NLG lexicons only via a wrapper mapping between the two models.

Given that the lexicon structure follows the WordNet synset model, such a lexicon can potentially be used for language analysis, if the application only requires synonymy. Our NLG lexicon model does not support yet the richer set of relations in WordNet such as hypernymy, although it is possible to extend the current model with richer relations. Since we used the lexicon in conjunction with the ontology, such non-linguistic relations were instead taken from the ontology.

The NLG lexicon itself is also independent from the generator's input knowledge and its format, i.e., is not restricted only to Semantic Web ontologies. The ontology-specific component is the ontolex mapping and its editor, because it relies explicitly on GATE's ontology model. In principle, any knowledge representation formalism with a similar expressive power as OWL Lite can be mapped to it and thus a generator using this KR formalism can benefit from the ontolex mapping tools, as well as the lexicon ones.

The need for a lexicon separate from the ontology and connected to it by a mapping model arises because most ontologies are not lexicalised, i.e., do not provide lexical information for their concepts and instances. For lexicalised ontologies like TAP (`http://tap.stanford.edu`) part of the NLG lexicon and the ontolex mapping can be derived automatically, although the remaining missing information (e.g., part of speech) will need to be added manually or from another lexicon.

[3] Available at http://www.acr.org/departments/stand_accred/birads/

[4] http://www.cancerscreening.nhs.uk/breastscreen/index.html

5. Related Work

Our NLG lexicon editor is most similar to the editor presented in (Callaway, 2002). The main difference is that the GATE-based tools provide a strong connection to ontologies, encoded in standard formats like RDF and OWL, whereas Callaway's authoring tool uses its own idiosyncratic format. A promising avenue to be explored in future work is the integration of these editors by combining the text-centered orientation of Callaway's tool with the extensive ontology support of the GATE-based ones.

Another popular development environment for NLG systems is KPML (Bateman, 1997), which is based on systemic linguistics. The main difference between the KPML tools and the GATE NLG tools is that the latter are independent from the linguistic theory used by the generator.

6. Conclusion

To summarise, this paper described reusable, open-source tools for creation, maintenance, storage, and access of language resources needed for generating natural language texts from ontologies. The advantages of having such tools are that they provide a user-friendly interface for NLG LR manipulation and unified models for accessing ontologies, NLG lexicons, and mappings between the two. At present we are working towards extending this tool set with other reusable NLG components.

Future work is also aimed at evaluating the robustness and scalability of the NLG lexicon tools. So far they have been used with relatively small lexicons (several hundred entries), therefore they need to be tested on bigger data sets, e.g., importing and editing of WordNet synsets and lemmas.

Another useful extension would be to provide input and output format for the NLG lexicon, compatible with emerging XML and RDF-based standards for lexical resources (Ide and Romary, 2002).

Acknowledgements

MIAKT is supported by the UK Engineering and Physical Sciences Research Council as part of the MIAKT project (grant GR/R85150/01. The author wishes to thank Borislav Popov and Atanas Kiryakov from Ontotext Lab, Sirma AI. for implementing GATE's ontology API and ontology visualisation tool; Hamish Cunningham, Valentin Tablan, and the rest of the GATE team for their help and support and for implementing the core GATE system; Yorick Wilks for his helpful comments and support. This paper also benefitted from the comments and suggestions of the three anonymous reviewers.

7. References

John A. Bateman. 1997. Enabling technology for multilingual natural language generation: the KPML development environment. *Natural Language Engineering*, 3(1):15—55.

S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. 2003. OWL Web Ontology Language Reference. Technical report, W3C Proposed Recommendation 15 December 2003, http://www.w3.org/TR/2003/PR-owl-ref-20031215/.

K. Bontcheva, A. Kiryakov, H. Cunningham, B. Popov, and M. Dimitrov. 2003. Semantic web enabled, open source language technology. In *EACL workshop on Language Technology and the Semantic Web: NLP and XML*, Budapest, Hungary.

C. B. Callaway. 2002. Tools for large-scale generation. In *Proceedings of the ACL-02 Demonstrations Session*, pages 118–119, Philadelphia, July.

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)*.

D. Fensel, W. Wahlster, and H. Lieberman, editors. 2002. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press.

I. Horrocks and F. van Harmelen. 2001. Reference Description of the DAML+OIL (March 2001) Ontology Markup Language. Technical report. `http://www.daml.org/2001/03/reference.html`.

B. Hu, S. Dasmahapatra, and N. Shadbolt. 2003. From Lexicon To Mammographic Ontology: Experiences and Lessons. In D. Calvanese, G. De Giacomo, and E. Franconi, editors, *Proceedings of the International Workshop on Description Logics (DL'2003)*, pages 229–233.

N. Ide and L. Romary. 2002. Standards for language resources. In *Proceedings of 3rd Language Resources and Evaluation Conference (LREC'2002)*, Gran Canaria, Spain.

O. Lassila and R.R. Swick. 1999. Resource Description Framework (RDF) Model and Syntax Specification. Technical Report 19990222, W3C Consortium, `http://www.w3.org/TR/REC-rdf-syntax/`.

D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. 2002. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274.

G. Miller. 1995. WordNet: a Lexical Database for English. *Communications of the ACM*, Volume 38(Number 11), November.

NLM. Unified Medical Language System (UMLS). Technical report, National Library of Medicine, http://www.nlm.nih.gov/research/umls/umlsmain.html.

E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge.

G. Wilcock and K. Jokinen. 2003. Generating Responses and Explanations from RDF/XML and DAML+OIL. In *Knowledge and Reasoning in Practical Dialogue Systems, IJCAI-2003*, pages 58–63, Acapulco.

G. Wilcock. 2003. Talking OWLs: Towards an Ontology Verbalizer. In *Human Language Technology for the Semantic Web and Web Services, ISWC'03*, pages 109–112, Sanibel Island, Florida.