

Comparing the template-based approach to GF: the case of Afrikaans

Lauren Sanby

Computer Science Dept.
University of Cape Town
Cape Town, South Africa

{lauren.sanby, ion.todd}@alumni.uct.ac.za

Ion Todd

Computer Science Dept.
University of Cape Town
Cape Town, South Africa

C. Maria Keet

Computer Science Dept.
University of Cape Town
Cape Town, South Africa

mkeet@cs.uct.ac.za

Abstract

Ontologies are usually represented in OWL that is not easy to grasp by domain experts. A solution to bridge this gap is to use a controlled natural language or natural language generation (NLG), which allows the knowledge in the ontology to be rendered automatically into a natural language. Several approaches exist to realise this. We used both templates and the Grammatical Framework (GF) and examined the feasibility of each by developing NLG modules for a language that had none: Afrikaans. The template system requires manual translation of the ontology’s vocabulary into Afrikaans, if not already done so, while the GF system can translate the terms automatically. Yet, the template system is found to produce more grammatically correct sentences and verbalises the ontology slightly faster than the GF system. The template-based approach seems easier to extend for future development.

1 Introduction

The knowledge acquisition bottleneck is well known for many years, and many proposals have been made to ameliorate this problem. One such avenue is to avail of a natural language interface. This has gained traction in the Semantic Web community in the past 10 years; two recent surveys on this intersection serve to illustrate its relevance (Bouayad-Agha et al., 2014; Safwat and Davis, 2016). While a template-based approach to generate natural language from OWL files is popular (e.g., (Androustopoulos et al., 2013; Third et al., 2011)), other approaches have been proposed, from ‘patterns’ (Keet and Khumalo, 2014) to specific grammars for controlled natural languages

(Kuhn, 2013) to the comprehensive Grammatical Framework (GF) that principally serves to translate between natural languages (Gruzitis et al., 2010; Ranta, 2011). It is not clear what would be the ‘best’ approach and technology to generate sentences from OWL files, if any, which may depend more on the system requirements or on the grammar of the language. To this end, we used a fairly controlled experiment in building two NLG modules that take OWL files as input for a language that had none—Afrikaans—by two people with the similar background in computer science in the same time frame. The template-based approach included a formal specification of correctness of encoding and a proof-of concept implementation. The GF-based approach used GF and required substantial software development. Both were evaluated and compared.

The tools, source code, template specification and GF file, test data, output files, and further information on design, proofs, and analyses of the experiments are online at: http://pubs.cs.uct.ac.za/honsproj/cgi-bin/view/2015/sanby_todd.zip/.

2 Design of the verbalisers

The template-based approach followed an NLG system-oriented development process (Reiter, 1997), focussing on: surface realisation as to what should go in the templates, a formal proof of correctness of the templates with respect to OWL, and subsequent implementation. Surface realisation included design choices; e.g., ‘must be’ vs. ‘at least one’, which is illustrated here for ‘is part of’ in *Tak* \sqsubseteq *∃is_deel_van.Boom*:

- (1) Elke tak moet deel van ‘n boom wees
‘each branch must be part of a tree’
- (2) Elke tak is deel van ten minste een boom
‘each branch is part of at least one tree’

noting that the second option is easier in a template-based approach, because then the name of the OWL object property can be used directly in the template rather than requiring additional string and verb processing.

GF is a functional programming language that has an abstract grammar as an intermediate language and a concrete grammar that defines how components should be put together in a sentence (Ranta, 2011). The latter is language-dependent and thus needs to be changed when adding a new language (e.g., (Angelov and Ranta, 2009)). GF has an Afrikaans library, but it needed to be extended so as to create the specific grammar files needed for verbalising OWL 2 DL ontologies in Afrikaans. The GF system also required software development for GF \leftrightarrow OWL file interaction, resulting in the architecture depicted in Fig. 1.

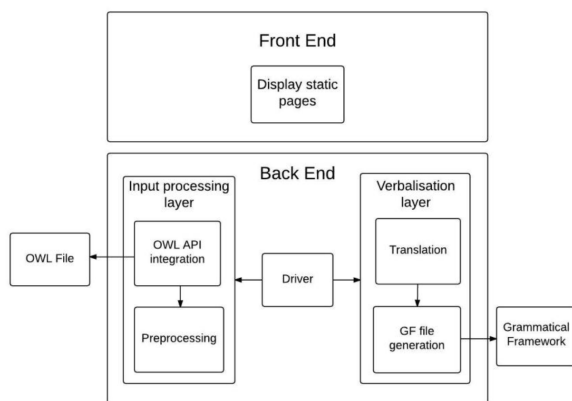


Figure 1: Architecture diagram for the GF approach. Preprocessing focuses on ‘normalising’ names from different ontologies, such as the property names *is-part-of* and *isPartOf*.

Let us illustrate the comparison of the declarative components with OWL’s disjoint classes axiom. The template is specified in XML as follows:

```
<Constraint type="Disjoint">
  <Text>'n</Text>
  <Object index="0"/>
  <Text>is nie 'n</Text>
  <Object index="1"/>
  <Text>nie</Text>
</Constraint>
```

and the corresponding GF concrete grammar as:

```
DisjointClasses x y =
  {s="'n "++x.s++"is nie 'n"
  ++y.s++"nie"};
```

Then, with the first Object index, or x , being, say, *Dier* ‘animal’ and the second one (y) *Plant* ‘plant’, then the sentence ‘n dier is nie ‘n plant nie ‘an animal is not a plant’ is generated. These tem-

plates and GF concrete grammar have been specified for most OWL 2 DL language features (see online material).

3 Verbalisations compared

Six ontologies were verbalised in Afrikaans. Due to space limitations, we only include here a selection of a range of types of axioms to illustrate the output and compare the two; the observations hold equally for the other axioms of the same type and across ontologies. The wine ontology is used, as that is an important subject domain in the Western Cape where Afrikaans is spoken widely.

Taxonomic subsumption of named classes for, e.g., $Chianti \sqsubseteq ItalianWine$ ‘Chianti is an Italian wine’ is verbalised as:

T: Elke Chianti is 'n ItalianWyn.

G: elke [Chianti] is [ItalianWyn]

where G (GF-based) misses the indeterminate article ‘n that T (template-based) correctly has. Equivalent classes for ‘Sweet wine is wine and has sugar with value sweet’ are verbalised as:

T: Elke SoetWyn (is Wyn en het suiker ten minste een Soet)

G: 'n [SoetWyn] is [Wyn] en [HetSuiker] [Soet]

The ontology is ambiguous in this regard due to the use of “value”, and therewith making it unclear whether *ten minste een* ‘at least one’ is the the best. A class expression with a disjunction or a one-of construct uses of ‘or’ in both cases, illustrated here with a one-of in $Zinfandel \sqsubseteq \forall hasFlavor.\{Strong, Moderate\}$ ‘Each Zinfandel has flavour only strong or moderate’:

T: Elke Zinfandel het net geur Gematig of Sterk

G: elke [Zinfandel] is iets wat [Gematig] of [Sterk] [HetGeur]

noting that T’s *net* ‘only’ is more precise with respect to the universal quantification in the axiom it verbalises and G’s *hetGeur* ‘has flavour’ is in the wrong place. Disjointness is correct in both cases, as in, e.g., $LateHarvest \sqsubseteq \neg EarlyHarvest$ ‘Late harvest is not an early harvest’:

T: 'n Laatoes is nie 'n VroeeOes nie.

G: 'n [Laatoes] is nie 'n [VroeeOes] nie

Object property range has valid alternatives; e.g., ‘has sugar has as range wine sugar’:

T: Iets het net suiker WynSuiker

G: Elke ding wat is [HetSuiker] , is
[WynSuiker]

Inverse object properties also has two valid alternatives, although T is easier to read thanks to the shorthand *teenoorgestelde* ‘opposite/inverse’, as in, e.g., ‘made from grape is inverse of made into wine’:

T: "gemaak in wyn" is die
teenoorgestelde van "is gemaak van
druive" (As X gemaak in wyn Y; Y is
gemaak van druive X.).

G: As X [GemaakInWyn] Y dan Y
[IsGemaakVanDruive] X. As X
[IsGemaakVanDruive] Y dan Y
[GemaakInWyn] X.

The verbalisation of a functional property is correct and understandable, respectively, although they ignores whether a domain was declared, which was ‘wine’ in ‘Each thing has color at most one wine color’:

T: Elke objek kan net een kleur hê.

G: Elke ding het een [HetKleur]

The templates cover object property characteristics better; e.g., with *oorganklike* ‘transitive’ in T, where G misses the implication, and G does not have a verbalisation for symmetric.

ABox assertions are correct in both approaches as well, although the template-based one is again more readable also for different individuals, like in ‘OffDry, dry, and sweet are all different’, for *almal* ‘all [of them]’ fits with a sequence of more than two items compared:

T: AfDroe en Droe en Soet is almal
verskillend.

G: [AfDroe] , [Droe] en [Soet] is al
verskillende

The verbalisation of object subproperty was incorrect in both cases. G verbalised it as symmetric, whereas T stringed the two verbalised parts of the axiom together in the wrong order. Notwithstanding, we include it here, because it demonstrates that referring expressions were incorporated correctly, indicated with *dit* ‘it’:

T: As iets het wyn descriptor, dit het
suiker.

(which ought to have read ‘if something has sugar then it has a wine descriptor’). Data properties were not included in the GF-based approach, but were in the template-based approach, which contributes to the difference in number of axioms verbalised (see Table 1). As the template-based ap-

proach took less time to develop, this freed up time to make the sentence more natural language-like, such as addressing the capitalisation and changing the object property names from, e.g., *IsGemaakVanDruive* ‘madeFromGrape’ (as in the wine ontology) to *is gemaak van druive*.

4 Evaluation

We conducted two experiments with the proof-of-concept software to compare the two approaches.

4.1 Experiment set-up

The template and GF-based programs were tested using six OWL ontologies. Measures such a number of axioms verbalised and time taken were collected. In addition, a general comprehensibility of the verbalisation evaluation was conducted with a human domain expert who is an Afrikaans mother tongue speaker. The six sets of sentences were assigned a quality category on a 5-point Likert scale: 1. Incomprehensible; 2. Almost completely incomprehensible; 3. Somewhat understandable; 4. Understandable but obvious errors; 5. Easy to understand no obvious errors.

4.2 Results and Discussion

Table 1 includes the main quantitative results. Although the template program does not have 100% coverage for any of the ontologies, the only missing axioms are those that are explicitly ignored (data types, keys, universal class and property). The “sentences written” column shows that there is a difference in number of sentences generated for all the ontologies for that reason, whose detailed analysis is included in the online supplementary material.

The evaluation of the sentence quality is shown in Table 2. The quality of the template-based approach is higher on average, though not statistically significantly (Mann-Whitney, $p=0.12852$). This is mainly because more time was available to refine the templates than GF’s concrete grammar.

As can be seen in Table 1, the template-based approach outperformed the GF-based approach on almost all measured metrics. It should be noted, however, that the GF-based approach includes also a translation module and generates the GF files dynamically. Also the GF \leftrightarrow OWL file interaction took extra time to develop, taking away time to refine the GF-based verbalisations. The on-the-fly translation takes more time to compute the results

Ontologies	ax.	Ax. verb.		Pct. verb.		Sent. written		Time (s)		Time/sent. (ms)	
		T	G	T	G	T	G	T	G	T	G
Pizza	712	707	711	99.3	99.9	707	711	1.7	2.1	2.4	3.0
African Wildlife	56	55	56	98.2	100.0	57	57	1.3	1.7	23.2	30.9
Computer Science	52	48	44	92.3	84.6	48	44	1.1	1.7	23.1	38.4
Wine	657	635	628	96.7	95.6	635	628	5.6	5.5	8.8	8.7
University	95	91	94	95.8	99.0	91	94	1.1	1.4	12.0	15.2
Stuff	136	134	110	98.5	80.9	175	110	1.2	1.5	6.9	13.7
<i>Average</i>				<i>96.8</i>	<i>93.3</i>	<i>285.5</i>	<i>273.8</i>	<i>2.0</i>	<i>2.3</i>	<i>12.7</i>	<i>18.3</i>

Table 1: Percentage ontology verbalisation for templates (T) and the GF program (G); |ax.| = total number of axioms; ax. verb. = axioms verbalised, pct. verb. = percentage verbalised; sent. = sentences.

Ontology	Template	GF
African Wildlife	5	4
Computer Science	4	4
Pizza	5	3
Stuff	4	2
University	4	4
Wine	4	4
<i>Average</i>	<i>4.4</i>	<i>3.5</i>

Table 2: Qualitative evaluation for the Grammar-based and template-based approaches.

compared to matching the OWL file with the templates in the XML file. While this is relatively minor with a small ontology, for a user to wait 6 seconds even with the wine ontology might become prohibitively slow with larger ontologies as well as in use cases that require runtime sentence generation. Finally, what also contributed to the template’s success is Afrikaans, which has very few morphological issues and not a complex system of concordial agreement like, e.g., isiZulu (Keet and Khumalo, 2014), that is also spoken in South Africa.

5 Conclusions

There is no clear ‘winner’ between a template-based approach and GF when one has to start from scratch with a natural language that is relatively amenable to templates, such as Afrikaans. Both are feasible, with the GF-based approach requiring more upfront investment and the template-based approach being easier to understand and therefore easier to refine and extend, provided a multilingual system does not become a requirement.

References

- I. Androutsopoulos, G. Lampouras, and D. Galanis. 2013. Generating natural language descriptions from owl ontologies: the naturalowl system. *Journal of Artificial Intelligence Research*, 48:671–715.
- K. Angelov and A. Ranta. 2009. Implementing controlled languages in GF. In *Proc. of CNL’09*.
- N. Bouayad-Agha, G. Casamayor, and L. Wanner. 2014. Natural language generation in the context of the semantic web. *Semantic Web J.*, 5(6):493–513.
- N. Gruzitis, G. Nespore, and B. Saulite. 2010. Verbalizing ontologies in controlled baltic languages. In *Proc. of HLT–The Baltic Perspective*, volume 219 of *FAIA*, pages 187–194. IOS Press, Riga, Latvia.
- C. M. Keet and L. Khumalo. 2014. Basics for a grammar engine to verbalize logical theories in isiZulu. In *Proc. of RuleML’14*, volume 8620 of *LNCIS*, pages 216–225. Springer, Aug. 18–20, 2014, Prague, Czech Republic.
- T. Kuhn. 2013. A principled approach to grammars for controlled natural languages and predictive editors. *J. of Logic, Lang. and Inf.*, 22(1):33–70.
- A. Ranta. 2011. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford.
- R. Reiter, E. & Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3:57–87.
- H. Safwat and B. Davis. 2016. CNLs for the semantic web: a state of the art. *Language Resources & Evaluation*, in print:DOI: 10.1007/s10579–016–9351–x.
- A. Third, S. Williams, and R. Power. 2011. OWL to English: a tool for generating organised easily-navigated hypertexts from ontologies. poster/demo paper, Open University UK. ISWC’11, 23–27 Oct 2011, Bonn, Germany.