A classification of grammar-infused templates for ontology and model verbalisation

Zola Mahlaza[0000−0001−9829−1480] and C. Maria Keet[0000−0002−8281−0853]

University of Cape Town, Cape Town, South Africa
zmahlaza,mkeet@cs.uct.ac.za

Abstract. Involving domain-experts in the development, maintenance, and use of knowledge organisation systems can be made easier through the introduction of easy-to-use interfaces that are based on natural language. Well resourced languages make use of natural language generation techniques to provide such interfaces. In particular, they often make use of templates combined with computational grammar rules to generate grammatically complex text. However, there is no model of pairing templates and computational grammar rules to ensure suitability for less-resourced languages. These languages require a modular design that ensures grammar detachability so as to allow grammar re-use across domains and applications. In this paper, we present a model and classification scheme for grammar-infused templates suited for less-resourced languages and classify existing systems that make use of them. We have found that of the 15 systems that pair templates and grammar rules, and their 11 distinct template types, 13 have support for detachable grammars.

1 Introduction

Involving domain experts in the construction, maintenance, and use of Knowledge Organisation Systems (KOS), such as ontologies, thesauri, and comprehensive metadata systems, requires end-user interaction with the system. A common approach to facilitate these interaction processes, is to verbalise—render in (pseudo-)natural language—the formally represented knowledge and to provide structured natural language for input to make the formalisation step easier. Examples of such systems include generating descriptions from ontologies in order to assist ontology experts in reaching a consensus about the scope and use of an ontology [31] and generating museum artefacts descriptions [1]. Such tools are developed mostly for English, but also other languages, such as Greek [1], Latvian [15], and isiZulu [19]. Technologically, this is typically achieved by using a Controlled Natural Language (CNL) or, more comprehensively, with Natural Language Generation (NLG) techniques [12,28] that predominantly choose a template-based approach. For instance, one could have an AGROVOC [26] template ... is used to make ..., where a domain expert could add, say, 'maize' and the drink 'chicha' or food 'mieliepap' in the open slots, respectively. This becomes a bit more involved with subsumption or broader-than relations, as then there are two template options, ... `is a` ... and ... `is an` ..., and thus needs a rule to select the correct template, being checking whether the first sound of the broader concept is a vowel sound or not. It faces increasing challenges—for both CNLs and NLG—with grammatically richer languages. For instance, an object property like *arbeitet für* 'works for' in some organisation ontology requires knowledge of the gender of the object so as to generate the correct article, and verb conjugation in most Bantu languages require the noun class of the subject to fire a rule to generate the verb in the sentence [8][1]. This so-called *surface realisation* step in generating textual descriptions thus may require multiple additional grammar rules to varying degrees so as to generate correct sentences. There are 'template-based' systems that address these limitations for well-resourced languages by combining templates with natural language Computational Grammar Rules (CGRs) [32]. However, most languages are not well-resourced. Therefore, the few grammar rules that are being, or have been, computerised for Less-Resourced Languages (LRLs) ideally should be reusable for other systems so as to reduce development efforts; e.g., the isiZulu grammar rules for ontology verbalisation [19] were reused for online language learning exercises [14]. This requires that template-based systems have a modular design in some way with detachable grammar rules that have no 'tactical generation' [13] function. To the best of our knowledge, this has not been assessed and categorised systematically. If known, however, it would provide insights that can assist with the design and deployment of multilingual CNLs and NLG for a broader range of languages spoken in the world.

We aim to address this gap by developing a model for pairing templates and natural language CGRs in different ways, which led to 7 different types of CNL or NLG systems that pair templates with CGRs. We analysed 41 previously published tools, identified a subset of 15 out of the 41 tools as making use of templates with CGRs, and classified the 15 tools into the different ways of pairing CGRs and templates. A majority, 8 out of 15, of the classified systems have support for languages other than English. This classification is a step toward simplifying the development of such grammar-enhanced template systems for LRLs.

The remainder of this paper is structured as follows: Section 2 presents the related work, Section 3 focuses of the developed the model of pairing templates and CGRs, Section 4 focuses on the classification of existing CNL and NLG systems, Section 6 presents the discussion, and Section 7 concludes the paper.

2 Related work

Contemporary comparisons of surface realisation methods often feature hand-coded templates, hand-coded grammars, and statistical methods [12]. Substantial information is available pertaining to the appropriate conditions for which

[1] e.g., when a person 'eats' something, it is *udla* in isiZulu (one of the 11 official languages of South Africa), but when a giraffe—a noun in a different noun class from person—eats something, then it is *idla*.

each of the methods can be used. For instance, templates are suitable "[w]hen application domains are small and variation is expected to be minimal, [and] realisation is a relatively easy task" [12, pg80], hand-coded grammars for "general-purpose, [and] domain-independent realisation systems" [12, pg80] where it is possible to provide "very detailed input" [12, pg80], and statical methods for when large corpora is available [12, pg81].

There are two types of templates, namely, the fill-in-the-slot templates [27] and the templates that make use of CGRs [32]. Templates of the first kind only have fixed words (including punctuation) and slots. An example is the template ... is a kind of ... for verbalising a simple super-class relation between two classes in an ontology so that when it is used to verbalise *<Benign hypertensive renal disease> SubClassOf <Hypertensive renal disease>*, we obtain Benign hypertensive renal disease is a kind of hypertensive renal disease [22]. The kind of values that can be inserted into the slots may differ based on the application; e.g. medical concepts [22] or soccer match information [33]. Despite their simplicity, these templates are used in a variety of applications including multilingual conceptual data model and application ontology verbalisation [16], tailored soccer summaries [33], etc. Templates of the second kind have CGRs that bestow them the ability to offer linguistic flexibility without necessarily having to build a complete grammar-based realiser like FUF/SURGE, MUMBLE, or RealPro [12, pg80]. For instance, syntax templates [32] introduce variation by attaching a syntax tree to each template. We will henceforth refer to templates of the second kind as *grammar-infused templates*.

To the best of our knowledge, there is no classification scheme that can be used to differentiate the systems that use templates. There only exists a classification of CNLs, named PENS, that may be used for the languages generated by the systems [21], which is based on four features of CNLs, namely: precision, expressiveness, naturalness, and simplicity. While PENS can be used to categorise the language generated by an ontology verbaliser, it is not suitable for differentiating the systems since it does not provide information pertaining to how templates and CGRs can be paired. A classification of the grammar-infused templates, as opposed to their resulting CNL, is important on its own because it can help when choosing an appropriate CNL or NLG system in cases where a modular design may be a priority.

A new method of differentiating NLG systems is required because the "in-depth vs. shallow generation" [6] differentiation is also insufficient for such as task. While it may be useful for a classification system whose sole purpose is to identify which grammar-infused template is suitable for small applications due to its "cost factor" [6], it is not informative regarding the grammar's detachability. We illustrate this using TEMSIS [6] and RoundTrip Ontology Authoring (ROA) [11]. While TEMSIS and ROA belong to the same class of shallow generators, they exhibit differences regarding grammar detachability. ROA's templates are designed to depend on an external grammar engine, SimpleNLG [13], for generating the third person singular forms of some words. On the other hand, TEMSIS's templates and associated CGRs are not separable because the deletion of the templates results in the deletion of the rules.

A new classification scheme that is based on the relationships between the templates and the natural language's CGRs and the detachability of the CGRs is required since existing schemes are insufficient. In the next section, we introduce two relationships for pairing templates templates with CGRs and the types of templates with CGRs that arise based on those relationships.

3 Pairing templates and grammar rules

A pairing of CGR sets and templates to ensure suitability for LRLs exhibits two features:

(1) *detachability*: a modular design that facilitates grammar re-use across domains.

(2) *scaffolding*: the possibility to encode CGRs within the underlying templates since the existing grammar engines in some languages may be limited if they even exist.

The model is sketched graphically in Figure 1. In this model, the CGR sets can be paired with the templates through two kinds of relationships: embedding and attachment where the latter can be compulsory or partial. The differentiation between partial and compulsory attachment allows one to specify that certain grammar rules must necessarily accompany a template (e.g. syntax trees in syntax templates [32]) or that certain grammar rules are compulsory only to a subset of the templates (e.g. noun pluralisation in patterns [8]).

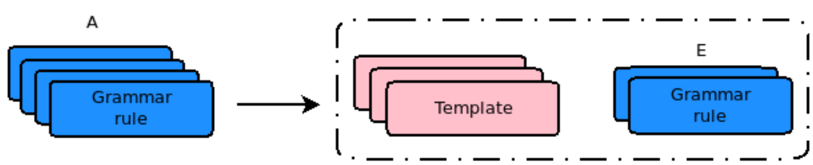

Fig. 1. Grammar-infused templates where templates are paired with CGR sets through two kinds of relationships; attachment and embedding. Attachment is illustrated through a directed arrow and embedding through a box whose border is a dotted line. The attached set is labelled with A and the embedded set is labelled with E.

In order to define embedding and the two kinds of attachment, we first introduce our definition of a natural language's grammar and its segmentation. Let a natural language grammar, denoted by G, be the set consisting of operations $O(W, \theta)$ whose arguments are words, W, and their features θ. The grammar can be segmented into portions using F, an ordered set of boolean predicate functions, $f : G \times G \rightarrow \{0, 1\}$ of any arbitrary size. The segmentation, and its subsets, govern the membership of operations to the grammar and its subsets. In particular, an operation's membership to the subsets labelled A and E in Figure 1 is determined by two different subsets of F.

Embedding exists between templates and a CGR set if and only if the CGR set ceases to exist when the templates are destroyed. This relationship refers to the tight coupling of a small portion of grammar functions with the templates. Attachment exists between templates and CGR sets if the grammar rule sets continue to exist even after the deletion of the templates. A CGR set is compulsorily attached if every template must use at least one rule from it. A CGR set is partially attached if not all templates must use rules from it.

Examples of the embedding and two attachment relationships can be seen in the isiZulu verbaliser [20] and GoalGetter [32]. Embedding and partial attachment can be observed in the isiZulu verbaliser and compulsory attachment in GoalGetter. The isiZulu verbaliser's grammar-infused templates, the so-called patterns, responsible for verbalising subsumption and universal quantification use a separate noun pluralisation module. However since other grammar-infused templates from within the same verbaliser, such as those for existential quantification, do not make use of rules from that module, we can conclude that the noun pluralisation rules are partially attached. Embedding can be seen in how the verbaliser encodes agreement between words. IsiZulu requires quantifiers, verbs, and other parts of speech to be in agreement with their governing noun through one or two morphemes. The isiZulu verbaliser encodes the agreement explicitly within each pattern such that the deletion of pattern will result in the loss of that agreement rule hence we say that such rules are embedded. GoalGetter's syntax templates are defined as the tuple $\sigma = (S, E, C, T)$ "where S is a **syntax tree** (typically for a sentence) with open slots in it, E is a set of links to additional syntactic structures (typically NPs and PPs) which may be substituted in the gaps of S,C is a condition on the applicability of σ, and T is a set of **topics**" [32, pg18]. Each template must, by definition, have a syntax tree hence we can conclude that the set of syntax rules is compulsory attached.

Grammar-infused templates can be categorised into different kinds of families. In order to explain how categorisation is conducted, let $T_e = (T_b, A_p, A_c, E)$ be an grammar-infused template where T_b is the underlying basic fill-in-the-slot template, A_p is the set of partially attached CGRs, A_c is the set of compulsory attached CGRs, and E is the set of embedded CGRs. There are seven families of grammar-infused templates to which a set containing grammar-infused templates, T_e, can belong and they are shown in Figure 2.

In the figure, each area denotes the pairing or lack thereof of two or three of the relationships; embedding, partial attachment, and compulsory attachment. In particular, a set of grammar-infused templates belong to one of the following families if all its grammar-infused templates meet the following criteria:

(1) *P* family: there is a set of CGRs that is attached to a subset of the templates, no set of CGR attached to all the templates, and no CGRs are embedded in the templates (i.e. $A_c = E = \emptyset$ and $A_p \neq \emptyset$ where $\emptyset$ denotes the empty set.)
(2) *C* family: there is a set of CGRs attached to all the templates, no set of CGRs attached only to a subset of the templates, and no CGRs are embedded in the templates (i.e. $A_p = E = \emptyset$ and $A_c \neq \emptyset$.)

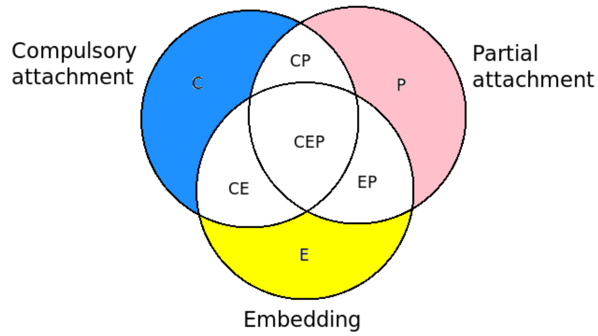

Fig. 2. Seven different types of grammar-infused templates. The primary relations (and their abbreviations) that are used to define the family types are P = partial attachment, C = compulsory attachment, and E = Embedding. CP, CE, EP, and CEP are combinations of the primary three relations.

(3) *E* family: there is a set of CGRs embedded in the templates, and no set of CGRs is attached to all or some of the templates (i.e. $A_c = A_p = \emptyset$ and $E \neq \emptyset$.)
(4) *CP* family: there is a set of CGRs that is attached to a subset of the templates and another set of CGRs attached to all the templates, but no CGRs are embedded in the templates (i.e. $A_c \neq \emptyset$ and $A_p \neq \emptyset$ and $E = \emptyset$.)
(5) *CE* family: there is a set of CGRs that is attached to all templates and another set of CGRs is embedded in the templates, but no CGRs are attached to a subset of the templates (i.e. $A_c \neq \emptyset$ and $E \neq \emptyset$ and $A_p = \emptyset$.)
(6) *EP* family: there is a set of CGRs embedded in the templates and a set of CGRs that is attached to a subset of the templates, but no set of CGRs that is attached to all the templates (i.e. $A_p \neq \emptyset$ and $E \neq \emptyset$ and $A_c = \emptyset$.)
(7) *CEP* family: family: there is a set of CGRs embedded in the templates, a set of CGRs that is attached to a subset of the templates, and another set of CGRs is attached to all the templates (i.e. $A_c \neq \emptyset$, $A_p \neq \emptyset$ and $E \neq \emptyset$.)

In order to illustrate a formalism's membership to one of the above families, we return to the running example of GoalGetter. Let us suppose that GoalGetter's syntax templates do not have any other rules besides the syntax trees. Then we can conclude that the formalism belongs to C family since it has compulsory attached rules but no embedded or partially attached rules.

4 Classification of grammar-infused templates

The aim of the classification is to identify the different ways templates and natural language CGRs are paired in the template formalisms used by various NLG tools. The classification will provide us with an understanding of the conditions in which each formalism is used. This improves the task of selecting an appropriate existing CNL/NLG system for text generation based on the one's requirements or available grammar rules.

We collected a set of 41 tools: 7 general linguistic realisers [13], 5 systems from [32], and 29 systems and verbalisers based on both a recent review [4] and a search for systems with support for a language other than English explicitly. Each of the 41 tools was annotated with either:

(1) *templates*: systems that make use of traditional templates, be this for a single template per unit of information or offering a selection among equivalent alternates for variation purposes only;
(2) *grammar*: these systems use full fledged grammars and essentially manage the surface realisation by availing of the grammar only, rather than also a template to constrain the sentence;
(3) *statistical methods*: systems that generate text using probabilistic grammars learned from corpora and generators that make use of statistical models to rank the output of a grammar that is either hand-coded or learned from corpora.
(4) *template + grammar*: those systems that use 'grammar-infused templates', in some configuration to enhance the grammatical correctness of the generated sentences. These must not make use of statistical methods.
(5) *other*: systems that do not fall into any of the previous categories (which may be because insufficient information was presented in the documentation).

There were 9 systems annotated with *templates*, 6 *grammar*, 3 *statistical methods*, 15 *templates + grammar*, and 8 with *other*. The systems that do not belong to the *template + grammar* group were filtered out as out-of-scope, and the remaining 15 systems were categorised into their respective grammar-infused template family using the model introduced in Section 3. The list of all the 41 considered NLG tools is available as supplementary material at https://github.com/AdeebNqo/grammarinfusedtemplates.

The resulting classification of the formalisms from the systems is given in Table 1. We illustrate the classification process into the various families with a selection of the systems.

Davis et al.'s [10] Grammatical Framework (GF) verbaliser generates multilingual text from models of business processes. Its syntax templates have two layers, namely, the abstract and concrete. The abstract segment is application specific and language independent while the concrete segment is language specific. In particular, the concrete segment attaches a syntax tree to the basic text through GF's resource grammar. However, the function for creating such trees persists in GF even when Davis et al.'s templates are removed. Its set of compulsory rules is not empty as its made up of the syntax trees. Furthermore, since there are no other rules associated with the templates, we categorise the formalism as belonging to the C family (Compulsory attachment).

Androutsopoulos et al.'s [1] NaturalOWL verbaliser generates bilingual museum artefact descriptions from an ontology. NaturalOWL's sentence plans are defined such that they "completely specify the surface (final) form of each sentence" [1, p699]. A plan is a sequence of slots where each slot is filled either by

[2] grammar-infused templates created using the KPML environment may belong to any of the seven families.

Table 1. Classification of grammar-infused templates for nine verbalisers, three NLG systems, and three realisers that have support for grammar-infused templates.

System/tool	**Family**	**Language(s)**	**Formalism**
Verbalisers			
Davis et al [10]	C	English, Dutch	GF syntax template
Stevens et al. [31]	C	English	Definite Clause Grammar (DCG) template
Kaljurand [17]	C	English	DCG template
Lim and Halpin. [23]	P	Malay, Mandarin	Logic pattern
Androutsopoulos et al. [1]	EC	English, Greek	Sentence plan
Gruzitis et al. [15]	EC	Latvian	GF syntax templates with synonymy
Davis et al. [11]	EP	English	XML template
Byamugisha et al. [7]	EP	Runyankore	Pattern
Keet et al. [20]	EP	IsiZulu	Pattern
NLG systems			
Stenzhorn [30]	EP	English, German, French, Italian, Russian, Bulgarian, Turkish	XML template
van Deemter [32]	EP	English, Dutch, German	Syntax template
Wilcock [34]	EP	-	Syntax template
Surface realisers			
McRoy et al. [25]	E	-	Template Specification Language (TSL)
Busemann [5]	E	-	Template Generation Language (TGL)
Bateman [2]	All[2]	Greek, English, German, Dutch, French, Japanese, Spanish	Extended Sentence Plan Language (SPL)

verbs, nouns, or adjectives from its detailed lexicon. The lexicon items are tables that encode all their inflectional forms. NaturalOWL's set of embedded functions is not empty as it is made up of the rules specifying agreement between lexical items. NaturalOWL's inflection rules are not specified in the sentence plan. They are tightly coupled with the external lexicon and not provided through an actual grammar. Nonetheless, we can conclude that its set of compulsory attached rules is not empty because they are provided via the lexicon. Since it does not have any other CGRs besides those, we categorise it as belonging to the EC family.

Davis et al.'s [11] ROA verbaliser generates English text from an ontology. The ROA verbaliser's templates are made up of three parts, namely, the *in*, *out*, and *ignoreIf* elements. The *in* element specifies the template's input RDF triple. The *out* specifies the singular and plural form of the underlying traditional template. Each form of the traditional template makes use of items from the *in* element and they are annotated with a value for grammatical number. The *ignoreIf* element specifies that a template should be ignored if its conditions are met. The set of its embedded functions is not empty as it contains the rules for specifying the singular/plural forms of the phrases in the underlying template. The set of partially attached rules is also not empty as it contains SimpleNLG's [13] rules creating the third person singular inflection. Since the templates do not have any additional rules, we categorise the tool as being from the EP family.

Table 1 shows that there are three GF-based systems [15,10,29] that differ in how they pair templates with CGRs. In particular, one system's templates are either not grammar-infused [29] and the other two system's templates belong to different families [15,10].

5 Use case: model verbalisation in isiZulu

We illustrate the usefulness of the classification within the context of creating a model for a lexicon to be used for an ontology thereby support multilingual access to Semantic Web data (e.g. [3,24]). In particular, we show the use of the classification from Section 4 when verbalising the model to improve the participation of domain experts in the validation of the model.

IsiZulu does not have ontology lexicalisation support, which is mainly due to its complex grammar that RDFS's *label* and *comment* properties, and even declarative modes such as *lemon* [24], do not cater for sufficiently [9]. The Bantu Language Model (BLM) [3] that has been created to support isiXhosa may be suitable for isiZulu since the two languages are mutually intelligible and exhibit grammatical similarities. However, BLM's suitability for isiZulu has not been tested and the participation of domain experts in testing it may be impeded by their inability to understand the formal language in which the model is codified. This can be addressed through the verbalisation of the model in isiZulu using grammar-infused templates. Here, our proposed classification scheme can be used to choose an appropriate grammar-infused template language or tool.

We have identified two use cases where there are different requirements regarding grammar availability and re-use, and we illustrate how to choose a suitable template language or tool to verbalise the model under development in each case. *Case 1*: when CGR re-use is not a priority and few templates are sufficient, then a template approach that has the ability to embed CGRs (E/CE/EP/CEP families) may be suitable for use as-is or as a foundation for building an NLG tool. *Case 2*: when all CGRs must be reusable and the available noun pluralisation and verb conjugation rules [18,19] are sufficient, then a template approach that has the ability to attach CGRs—C/P/CP/CE/EP/CEP families—is appropriate. However, practically, since the only existing grammar-infused template formalism with support for isiZulu (i.e., patterns [20]) belongs to the EP family, then we conclude that all cases can be supported albeit in a limited manner for case 1. Support for case 1 is limited because those patterns [20] can only embed morphological agreement rules.

In summary, when building a natural language text generator and there are requirements regarding the availability and re-usability of the CGRs, then the classification could be useful in choosing an appropriate template formalism.

6 Discussion

We have developed the first classification scheme of grammar-infused templates. Our classification of existing NLG tools shows that most grammar-infused templates have detachable grammars, hence, support grammar re-use in some form. Nonetheless, we have observed that the technology used to create the templates does not guarantee a form of grammar-infusion.

The use of a complex grammar formalism to encode templates that are not grammar-infused can be observed by assessing the differences between the three verbalisers that make use of templates encoded in GF [10,15,29]. In the concrete layer, Sanby et al. [29] create a GF application grammar such that the concrete syntax is only responsible for inserting the values of the slots in the appropriate template. Essentially, their concrete 'grammar' is a basic fill-in-the-slot template despite using GF, as the rules in the Afrikaans resource grammar were deemed not usable. In contrast, Davis et al. [10] and Gruzitis et al. [15] make use of GF's resource grammar library thereby attaching syntax trees to their templates. This demonstrates that the technology used for encoding templates is not a reliable feature for classifying grammar-infused templates, but that our classification can bring afore.

The tools that we considered for the classification proposed are general purpose surface realisers, NLG systems, and verbalisers. There are systems that were not included due to insufficient information. There are CNLs reviewed by Safwat and Davis [28] that sound relevant by name, but they focus on the step of ontology authoring rather than verbalisation, that were not included because of the nine CNLs, five of them either "[do] not focus on bidirectionality", "sentences may look unnatural", or the CNL's sentences are "less readable than pure natural language" [28]. The remaining four systems do not present sufficient information pertaining to how they verbalise a conceptual data model or ontology. In particular, they either focus solely on document authoring and do not specify how they might be used to generate text or they make use of template-like language elements for authoring ontologies but it is unclear whether those same elements are used as-is for verbalisation. The only authoring and verbalisation system that is included in the scheme is ROA since its verbalisation procedure is well documented [11]. This lack of sufficient technical information makes it a challenge for one to decide whether such systems might be appropriate to re-use or as a base for new systems. Conversely, the classification scheme we developed could contribute to clarifying such matters for future tools, by means of succinctly specifying the nature of a system's template and CGR pairing thereby assist prospective re-use or repurposing.

7 Conclusions

We have developed a model of pairing templates with natural language CGRs, used it to develop a classification scheme where an grammar-infused template formalism may belong to one of 7 families, and classified existing grammar-infused templates. Most existing grammar-infused templates belong to the C/P/CE/EP families; hence support detachable CGRs.

We are currently working on devising modular architectures for NLG systems that have the capability to also process agglutinative languages.

References

1. Androutsopoulos, I., Lampouras, G., Galanis, D.: Generating natural language descriptions from OWL ontologies: the NaturalOWL system. J. Artif. Intell. Res. **48**, 671–715 (2013)
2. Bateman, J.A.: Enabling technology for multilingual natural language generation: the KPML development environment. Natural Language Engineering **3**(1), 15–55 (1997)
3. Bosch, S.E., Eckart, T., Klimek, B., Goldhahn, D., Quasthoff, U.: Preparation and usage of Xhosa lexicographical data for a multilingual, federated environment. In: Proc. of LREC 2018, Miyazaki, Japan, May 7-12, 2018. (2018)
4. Bouayad-Agha, N., Casamayor, G., Wanner, L.: Natural Language Generation in the context of the Semantic Web. Semantic Web **5**(6), 493–513 (2014)
5. Busemann, S.: Best-First Surface Realization. arXiv e-prints cmp-lg/9605010 (May 1996)
6. Busemann, S., Horacek, H.: A flexible shallow approach to text generation. In: Proc. of INLG 1998, Ontario, Canada, August 5-7, 1998 (1998)
7. Byamugisha, J., Keet, C.M., DeRenzi, B.: Bootstrapping a Runyankore CNL from an isiZulu CNL. In: Proc. of CNL 2016, Aberdeen, UK, July 25-27, 2016. pp. 25–36 (2016)
8. Byamugisha, J., Keet, C.M., Khumalo, L.: Pluralising nouns in isiZulu and related languages. In: Proc. of CICLing 2016, Konya, Turkey, April 3-9, 2016. pp. 271–283 (2016)
9. Chavula, C., Keet, C.M.: Is *lemon* sufficient for building multilingual ontologies for Bantu languages? In: Proc. of OWLED 2014, Riva Del Garda, Italy, October 17-18, 2014. pp. 61–72 (2014)
10. Davis, B., Enache, R., van Grondelle, J., Pretorius, L.: Multilingual verbalisation of modular ontologies using GF and lemon. In: Proc. of CNL 2012, Zurich, Switzerland, August 29-31, 2012. pp. 167–184 (2012)
11. Davis, B., Iqbal, A.A., Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Handschuh, S.: Roundtrip ontology authoring. In: Proc. of ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. pp. 50–65 (2008)
12. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. J. Artif. Intell. Res. **61**, 65–170 (2018)
13. Gatt, A., Reiter, E.: SimpleNLG: A realisation engine for practical applications. In: ENLG 2009 - Proc. of the 12th ENLG, March 30-31, 2009, Athens, Greece. pp. 90–93 (2009)
14. Gilbert, N., Keet, C.M.: Automating question generation and marking of language learning exercises for isiZulu. In: Proc. of CNL 2018, Maynooth, Co. Kildare, Ireland, August 27-28, 2018. pp. 31–40 (2018)
15. Gruzitis, N., Nespore, G., Saulite, B.: Verbalizing ontologies in controlled Baltic languages. In: Human Language Technologies - The Baltic Perspective - Proc. of the Fourth International Conference Baltic HLT 2010, Riga, Latvia, October 7-8, 2010. pp. 187–194 (2010)

16. Jarrar, M., Keet, C.M., Dongilli, P.: Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Tech. rep., Starlab, Vrije Universiteit Brussel, Belgium (February 2006)
17. Kaljurand, K.: Attempto Controlled English as a Semantic Web Language. Ph.D. thesis, Faculty of Mathematics and Computer Science, University of Tartu (2007)
18. Keet, C.M., Khumalo, L.: Grammar rules for the isiZulu complex verb. Southern African Linguistics and Applied Language Studies **35**(2), 183–200 (2017)
19. Keet, C.M., Khumalo, L.: Toward a knowledge-to-text controlled natural language of isiZulu. Language Resources and Evaluation **51**(1), 131–157 (2017)
20. Keet, C.M., Xakaza, M., Khumalo, L.: Verbalising OWL ontologies in isiZulu with Python. In: The Semantic Web: ESWC 2017 Satellite Events. LNCS, vol. 10577, pp. 59–64. Springer (2017), 30 May - 1 June 2017, Portoroz, Slovenia
21. Kuhn, T.: A survey and classification of controlled natural languages. Computational Linguistics **40**(1), 121–170 (2014)
22. Liang, S.F., Scott, D., Stevens, R., Rector, A.: Unlocking medical ontologies for non-ontology experts. In: Proc. of BioNLP 2011 Workshop. pp. 174–181. Stroudsburg, PA, USA (2011)
23. Lim, S.H., Halpin, T.A.: Automated verbalization of ORM models in Malay and Mandarin. IJISMD **7**(4), 1–16 (2016)
24. McCrae, J.P., Bosque-Gil, J., Gracia, J., Buitelaar, P., Cimiano, P.: The Ontolex-Lemon model: development and applications. In: Proc. of eLex 2017, Leiden, the Netherlands, 19-21 September 2017. pp. 19–21 (2017)
25. McRoy, S.W., Channarukul, S., Ali, S.S.: Yag: A template-based generator for real-time systems. In: Proc. of INLG 2000. pp. 264–267. Stroudsburg, PA, USA (2000)
26. Rajbhandari, S., Keizer, J.: The AGROVOC concept scheme - a walkthrough. Journal of Integrative Agriculture **11**(5), 694–699 (2012)
27. Reiter, E.: NLG vs. templates. CoRR **cmp-lg/9504013** (1995), http://arxiv.org/abs/cmp-lg/9504013
28. Safwat, H., Davis, B.: CNLs for the Semantic Web: a state of the art. Language Resources & Evaluation **51**(1), 191–220 (2017)
29. Sanby, L., Todd, I., Keet, C.M.: Comparing the template-based approach to GF: the case of Afrikaans. In: Proc. of WebNLG 2016, Edinburgh, UK, September 6, 2016. pp. 50–53 (2016)
30. Stenzhorn, H.: Xtragen - A natural language generation system using XML and Java-technologies. In: Proc. of NLPXML@COLING 2002, Taipei, Taiwan, August 24 - September 1, 2002 (2002)
31. Stevens, R., Malone, J., Williams, S., Power, R., Third, A.: Automating generation of textual class definitions from OWL to English. J. Biomedical Semantics **2**(S-2), S5 (2011)
32. van Deemter, K., Theune, M., Krahmer, E.: Real versus Template-Based Natural Language Generation: A False Opposition? Computational Linguistics **31**(1), 15–24 (2005)
33. van der Lee, C., Krahmer, E., Wubben, S.: PASS: A Dutch data-to-text system for soccer, targeted towards specific audiences. In: Proc. of INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017. pp. 95–104 (2017)
34. Wilcock, G.: Pipelines, templates and transformations: XML for natural language generation. In: Proc. of NLPXML@NLPRS 2001, Hitotsubashi Memorial Hall, National Center of Sciences, Tokyo, Japan, November 27-30, 2001 (2001)