ToCT: A task ontology to manage complex templates

Zola Mahlaza, C. Maria Keet

Department of Computer Science, University of Cape Town, Cape Town, South Africa

Abstract

Natural language interfaces are a well-known approach to grant non-experts access to semantic web technologies. A number of such systems use simple templates to achieve that for English and more elaborate solutions for other languages. They keep being designed from scratch in an ad hoc manner, since there is no shared conceptualisation of simple templates and there is no model that is formalised using a Semantic Web language to apply the techniques to itself. We aim to address this by proposing a general-purpose solution in the form of a novel model for templates, formalised as a task ontology in OWL, called ToCT. We used it to develop an ontology-driven text generator for isiZulu, a morphologically-rich language, to test its capabilities. The generator verbalises the TBox of an ontology as validation questions. This evaluation showed that the task ontology is sufficiently expressive for the template design, which was subsequently verified with user evaluations who judged the texts positively.

Keywords

Natural language interfaces, Task Ontologies, Ontology-driven tools, Low-resourced languages

1. Introduction

Natural language interfaces are widespread as means of granting non-experts access to semantic web technologies. Natural language generation (NLG) systems have been used to generate museum artefact descriptions from ontologies, amongst other things (see [1]). Most such systems generate only English text and use simple templates. When one needs to capture complex grammatical features like agreement, writing conventions are sometimes used to circumvent the adoption of complex generation techniques, such as braces. For instance, one can use the short French template *Un(e) [parent]* to generate *Un(e) mère* 'One mother', where the braces are used to capture both possible forms of *Un(e)* 'one', a gender-dependent word. When 'simple' templates are not sufficient, such as when agreement cannot be captured with occasional braces, then "grammar-infused templates" [2] are used, i.e., templates + grammar rules.

Most previous versions of grammar-infused templates have been created in an ad hoc way (e.g., [3, 4, 5, 6, 7]). Yet, widespread adoption of grammar-infused templates across different languages relies on the ability to, mainly, specify them systematically and to compare and determine the suitability of these kinds of templates for each natural language. This is currently impossible, since there is no shared conceptualisation of template specifications across languages that exhibit different types of morphology. Consequently, one cannot accurately assess each

FOIS'21 Ontology Showcase: 12th International Conference on Formal Ontology in Information Systems, September 13–16, 2021, Bozen-Bolzano, Italy

zmahlaza@cs.uct.ac.za (Z. Mahlaza); mkeet@cs.uct.ac.za (C. M. Keet)
0000-0001-9829-1480 (Z. Mahlaza); 0000-0002-8281-0853 (C. M. Keet)

template type's novelty, and neither verify their consistency nor achieve interoperability, nor systematically examine these templates' suitability for multiple languages.

We solve these problems by developing ToCT, a task ontology for templates that has support for sub-lexical components. To serve morphologically-rich and agglutinating languages well, its development focuses on two South African low resourced languages, isiZulu and isiXhosa, in addition to capturing the concepts found in English-oriented templates. We evaluated the task ontology in several ways. Competency questions were used to validate the ontology. To show that the ontology can capture words that fall outside the competence of common templates for languages with simple morphology, we created a ontology-driven text generator, described in [8], for the verbalisation of an ontology's TBox as validation isiZulu questions. Most of its questions are judged positively for grammaticality or understandability. ToCT and all other supplementary material can be accessed via the stable identifier https://doi.org/10.5281/zenodo.4704362.

The rest of the document is structured such that the next section discusses related work and Section 3 presents our ontology. We discuss and conclude in Section 5.

2. Related work

Since there are no existing ontologies for template specification, our work relates to existing *ad hoc* template models and lexical models used for ontology lexicalisation, which potentially could be of use for bottom-up ontology development from non-ontological resources.

An XML schema specification for templates for Object-Role Modeling diagrams was used by [9], with fixed textual segments and slots to insert different values. It, together with its *ad hoc* variatons (e.g., [10]), may suffice for languages with a simple morphology (e.g., English) or where one can use writing conventions (recall Section 1), which is not the case for agglutinating and highly inflectional languages.

Only 9 of the 16 text generators that rely on grammar-infused templates [11] support a language other than English. Of those, only 'patterns' [12, 13] support some African languages. These grammar-infused templates have wide domain applicability, since the constituents of the simple template should, theoretically, be able to support all domains. Practically, the existing ones [12, 13] are ad hoc, thus would have to be adjusted to be usable in another application domain. For instance, the original constituents in the isiZulu patterns [14] had to be amended so as to support part-whole relations [15]. The other models for other-than-English tools are also ad hoc and do not use the same shared concepts and relations.

Existing ontology lexicalisation models might be reused as base. However, adapting them means not only adopting their idiosyncrasies but spinning off a variant, since they do not have a large number of the required features. Notably, ontolex-lemon and GOLD and related artefacts have been shown to be not usable off-the-shelf [16, 17] for the languages in question. The only model built for such languages, i.e, [18], requires one create all forms of a lexical entry instead of creating them 'on the fly' when they are used in a template, it has no notion of a template, and provides no means for specifying dependencies between words.

The need for ToCT is demonstrated by the limitations of these existing models and is also cemented by the impracticality of alternative generation approaches, within the context of African languages (see also [13]).

3. Template ontology

To create the task ontology for complex templates, we carried out four main activities: competency question (CQ) creation, knowledge gathering, ontology creation and formalisation, and validation. They were carried out in two iterations, broadly following the main tasks in ontology development methodologies as summarised in [19]. We will discuss these steps in details and also discuss the ontology's contents.

Competency question creation. We created 13 CQs to indicate the scope, where the brackets are used to denote variables. A selection is included here (the full list is available in the supplementary material):

Q1: How many words does [template] have
Q2: How many words that depend on others does [template] have?
Q3: Which property results in a change in surface form between [word]?
Q4: If there is a dependency between [word] and [word], which word is the governor?
Q5: Does [word] have a constant base form?
Q6: Which grammar rule will be activated when forming [word] if it's dependent on another word?
Q7: Can [word] ever be placed in [slot]?
Q8: Can [grammar rule] exist without [word]?

The answerability of the questions will be revisited further below.

Ontology creation. Concerning knowledge gathering, we sourced primary sources, such as grammar textbooks, that describe morphology and grammar (a.o., [20, 21, 22]; a complete list is included in the supplementary material). We also relied on the authors' domain knowledge regarding English-oriented templates, and ZM's knowledge of isiXhosa as L1 speaker and for both authors's knowledge of isiZulu as Lx speakers and researchers.

The first iteration went through the entire development process and focused on knowledge acquisition and support for isiXhosa, and the second iteration focused on improvements for applicability and extensions to also suffice for both isiXhosa and isiZulu. Following best practice, we sought to reuse existing ontologies to aid in the conceptualisation and formalisation stage. We reused the Collections Ontology (CO) [23] and the Model for Language Annotation (MoLA) [24]. We created a module from the CO (omitting the not needed Disjoint(Collection Item) axiom) for the ordered and nonunique associations. The *Language* attribute in *Template* is captured through the object property *supportsLanguage* and the MoLA's inclusive *Language* concept [24]. The resulting model is shown informally in Fig. 1, which was then formalised in OWL, resulting in a so-called 'task ontology', called ToCT, Task ontology for CNL Templates, intended for the specific task of systematic, reusable, and interoperable template specification.

Ontology content. In the ontology, a template is a sequence of ordered words, slots, spaces, and punctuation. A *Word* has the regular meaning and there are two kinds of words in the ontology; fixed words and changing words. Fixed words, captured with *Unimorphic word*, are signs with a single written representation. Changing words, captured with *Polymorphic word*, are signs with multiple written representations because they possess at least one morpheme whose value is dependent on other words. A *Phrase* is an sequence of *Unimorphic* and/or *Polymorphic Words*. A *Slot* is a placeholder in a template and its value is drawn from a finite collection of *Words* or *Phrases* where the collection is referred to as slot fillers. The slot's label is a sign used to

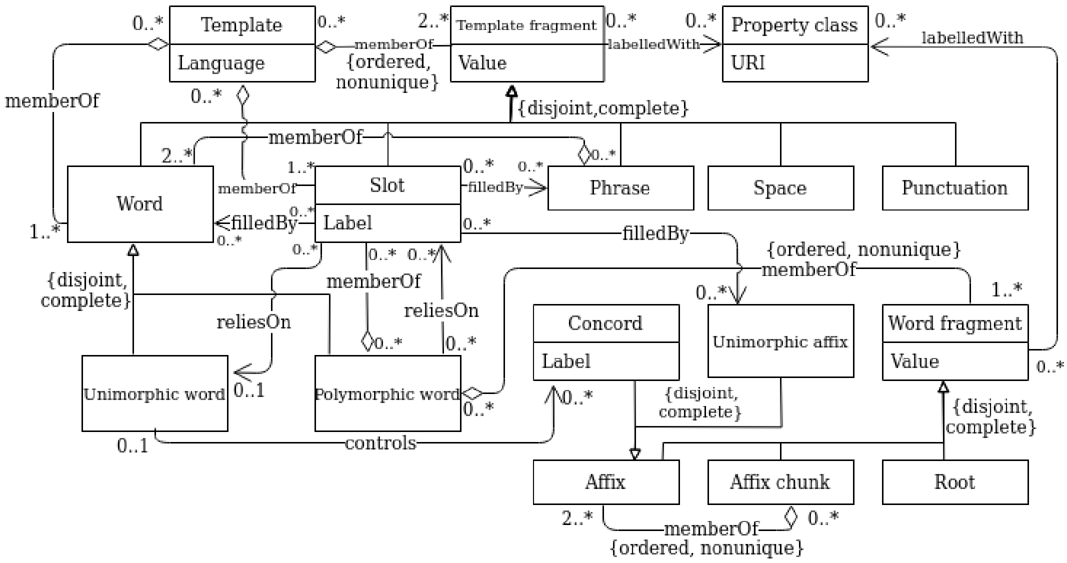

Figure 1: Informal rendering of the ToCT for template specifications, in UML class diagram notation.

represent the role played by all the slot fillers. The label is not necessarily a word or phrase that exists in natural language. For instance, "*[first_name]*" is a valid slot label even though it does not exist in any natural language—only its constituent signs are valid English words (e.g., "*first*" and "*name*"). The *Space* and *Punctuation* concepts have their usual linguistic meaning.

There are three types of primary sub-lexical items in ToCT; *Root*, *Unimorphic affix*, and *Concord*. A *Root* is a morpheme that carries the principal meaning of a word and it cannot be decomposed to finer granularity without losing its identity. A *Unimorphic affix* is a morpheme with a fixed value that carries some abstract meaning and it is distinct from the *Root*. A *Concord* is also a morpheme that is distinct from the *Root*, it carries some abstract meaning, it is appended to a root or stem for morphological agreement with other words, and its value is drawn from a finite collection of possible values (i.e., the concord fillers) based on the linguistic properties of a controlling *Unimorphic Word*. The concord's label represents the role played by all the concord fillers. The properties of the various template and word portions can be specified via the *Property class*. An *Affix chunk* is a sequence of *Affixes*.

The *labelledWith* relation specifies the properties of a template/word portion. The *memberOf* relation is a subtype of member/collection meronymic relation where (1) the collection is ordered, (2) the member and collection are non-agentive social objects, and (3) the collection is essentially dependent on the member. The *filledBy* relation specifies that a *Word* or *Phrase* is member of a specific slot's fillers and *controls* specifies a rigid essential necessitation between a unimorphic word and a concord's value. The *reliesOn* relation specifies a rigid existential necessitation between (1) a slot and unimorphic word or (2) between a polymorphic word and slot. It signifies that a polymorphic word inserted into the slot cannot have a value without a *Unimorphic word* and the polymorphic word's value depends on the *Unimorphic word* that is inserted into the slot. They all have inverses, which are not shown in Fig. 1 for brevity.

Ontology validation. First, to take advantage of a reasoner to possibly detect and remove inconsistencies, we formalised the model (or: output of the conceptualisation stage) in OWL and used the FaCT++ 1.6.5 reasoner to detect any inconsistencies; none were found.

Second, in revisiting the CQs, we updated three to improve their clarity, motivated be additional domain knowledge gained during the process: CQ Q1 and Q2 were refined as EQ1 and EQ2, respectively, and Q8 was refined into EQ8.1 and EQ8.2 (see online material). The 14 CQs were translated into SPARQL (see online material) and the translatable questions were used to query ToCT. Eight questions were translatable and answerable after querying the ontology. EQ8.1 and EQ8.2 were answerable through the analysis of the ontology. The remaining questions (Q6, Q9, Q10 and Q13), were not answerable as they were no longer in scope: they pertain to linguistic rules that turned out to be unnecessary for morphologically-enhanced templates.

4. Use of the ontology in an application

We now turn to illustrate the capabilities of the ontology, the datasets used to ascertain the capabilities, and also demonstrate how to use the ontology. We have used ToCT to create a dataset of templates for the isiZulu question generator [8] and the templates used to validate the text generation Java source code. The templates include language features that are outside the competence of English-oriented templates. We demonstrate how the verbaliser's templates were captured by using a template taken from [25] and an isiZulu pattern taken from [13]. Due to space considerations, we will omit some details from the ToCT demonstrative templates:

English template The template we have chosen was created for generating text for the E2E challenge [25]: ⟨name⟩ is a ⟨familyFriendly⟩ ⟨eatType⟩ which serves ⟨food⟩ food.

In the given template, the braces are used to denote the positions of slots and the labels enclosed by them are used to specify the 'meaning' of each slot. We capture the template using ToCT and obtain the following template, expressed in Turtle syntax (explained afterwards):

```
1  <restaurantTempl> a toct:Template
2         ; co:firstItem <restNameSlot>
3         ; co:lastItem <foodWor>
4         ; co:item <isPhr>, <famFrSlot>,
5         <eatTySlot>, <servPhr>, <foodSlot> .
6  <restNameSlot> a toct:Slot
7         ; toct:hasLabel "name" .
8
9  <isPhr> a toct:Phrase
10        ; toct:hasValue "is a "
11        ; co:firstItem <isWor>
12        ; co:lastItem <aWord> .
13 <isWor> a toct:UnimorphicWord
14        ; toct:hasValue "is" .
```

The snippet uses three prefixes, which refer to our ToCT (abbreviated *toct*), and the two reused ontologies, as mentioned above (*mola* and *co*). The snippet includes only template's first two fragments. We omitted the elements that separate words and slots. The original English template uses only slots and fixed segments. There are two types of fixed segments in the templates, namely words and phrases. As such, all its components can be captured via ToCT's *Slot* (e.g., line 6), *UnimorphicWord* (e.g., line 14), and *Phrase* (e.g., line 9) concepts. The slot's label (e.g., line 7) is intended for human users. The slots' meaning can captured via the *labelledWith* relation and external annotation ontologies. This is intended for computer use and one of its purposes is template search and discovery.

When one uses *ad hoc* models for templates, they often work with a small collection of self-created templates. Since ToCT supports template sharing and reuse, one is likely to have a large repository of templates, especially for real-world systems. In such cases, one can search for potential templates by using the 'meaning' of the slots. ToCT's annotation of slots via the *labelledWith* relation, when used with an ontology-based search engine, may improve the search experience by using inferred knowledge (e.g., Discover fruit-related templates when searching for tomato-related templates since $Tomato \sqsubseteq Fruit$). Technically, one can achieve the same goal by annotating XML templates, however, ToCT goes beyond that and offers the advantages associated with semantic web technologies and also caters for languages that are morphologically rich.

IsiZulu Pattern The chosen pattern was created for verbalising object properties in the present tense. It generates texts of the sort *wonke umuntu udla isithelo esisodwa* 'each human eats at least one piece of fruit' where all the underlined text are slots in the pattern: $\langle$QC(all) for $NC_x\rangle$onke $\langle$pl. N_1, is in $NC_x\rangle$ $\langle$SC of $NC_x\rangle\langle$verb stem$\rangle$a $\langle N_2$, is in $NC_y\rangle$ $\langle$EC for $NC_y\rangle\langle$QC for $NC_y\rangle$dwa.

We encode this in ToCT and obtain the following template (partial for brevity):

```
1  <plainOPPresentTense> a toct:Template
2  ; toct:supportsLanguage <kzn_zulu>
3  ; co:firstItem <all>
4  ; co:lastItem <only>
5  ; co:item <noun1Slot>, ...,
6          <noun2Slot> .
7  <kzn_zulu> a mola:Dialect; ...
8  ; mola:isFamily <isiZulu> .
9
10 <all> a btm:PolymorphicWord
11 ; co:index '1'^^xsd:positiveInteger
12 ; toct:reliesOn <noun1Slot>
13 ; co:fistItem <quantConcord>
14 ; co:lastItem <onke>
15 ; co:nextItem <noun1Slot>.
```

The snippet illustrates the declaration of a template by showing three elements: the main template container, partial details of the dialect supported by the template, and the template's first word. Its lines 7-8 denote a dialect of isiZulu spoken in KwaZulu-Natal, South Africa that is part of the isiZulu family (line 8). The order of the template's words is specified through the Collections Ontology's *firstItem* and *lastItem* (lines 3-4). It is complemented by having specified the ordering of words that fall in between the first and last is through the object property *nextItem* (e.g., line 15).

The actual words are subclassses of *co:ListItem* and they are specified as being contained in the template via *co:item* (lines 5-6) thus the cardinality constraints can be enforced. Lines 10-15 illustrate the declaration of a *Polymorphic word.* The word is made up of two affixes: thequantitative concord and the root *-onke*, so can result in the surface forms *zonke, bonke, lonke*, etc. The polymorphic word's value is determined by a value inserted into the slot *noun1Slot* since the polymorph is specified as relying on its for its form (line 12). Practically, the polymorphic word cannot have a final form unless a value that controls *quantConcord* is inserted into the *noun1Slot* slot. This snippet demonstrates the capability to encode sub-lexical components and specify dependencies between some words, as needed for languages like isiZulu.

The verbaliser [8] with its templates were tested on an extended version of the the test ontology of [13]. We observed that generated texts are free of morphological agreement and phonological conditioning errors and a majority of the questions are judged positively for grammaticality or understandability. These templates are available as supplementary material.

5. Discussion and conclusions

Having introduced, evaluated, and demonstrated how to use ToCT, we now turn to discuss the strengths of our approach compared to existing methods. There are no existing ontologies for template specification other than ToCT introduced here. While the focus was on catering for templates for African languages, the plurality of incompatible verbalisers for English suggest it can benefit from at least systematization, if not standardisation. ToCT can assist with that and its strengths lie with grammatically more challenging languages with features such as agglutination and having to deal with gender and noun classes that affect other words in a sentence. More specifically, it does not require the creation of multiple forms of the same template to cater for changing words. It also does not pose a categorical limit for slot fillers; any affix, word, or phrase can be used and their POS or sub-lexical category can be specified separately via the *Property class* and *labeledWith* object properties (cf. pattern's [14] original coverage of only nouns, verbs, and pronouns, but not, e.g., attributes). Consequently, with ToCT, subject domain semantics are not buried in the code and there is an explicit template specification thus enabling reuse.

English-oriented templates are longer when encoded via ToCT vs. via other existing methods (e.g., XML-based templates [9]). It may also be difficult to visualise what is captured by each multipart template fragment. These issues should not prohibit the adoption of ToCT because it is designed for computer-use and we do not expect engineers to create, manage, and and read the ttl encodings of the templates directly. Instead, software systems with graphical user interfaces are envisioned for the creation, management, and visualisation of the artefacts (à la Protégé [26] for ontologies).

To conclude, then, we showcase a task ontology for morphologically-enhanced templates that has been shown to be able to capture templates that are outside the competence of existing models of English-oriented templates. This extended coverage as well as principle of reusability with an ontology offer an advantage over 'patterns' as well, since it supports all the morphological dependencies and the insertion of all POS categories into slots. The evaluation of ToCT via a question generator demonstrated that the generated texts are well-formed and a majority are judged positively for grammaticality or understandability. This demonstrates that it is valuable in supporting the creation of natural language interfaces for morphologically-rich languages within the Semantic Web community and beyond, with good prospects for widespread adoption of the ontology. ToCT is also available on Github[1] for update requests and issue reporting.

Acknowledgments

This work was financially supported by Hasso Plattner Institute for Digital Engineering through the HPI Research School at UCT and the National Research Foundation (NRF) of South Africa (Grant Number 120852).

[1] Accessible via https://github.com/AdeebNqo/ToCT

References

[1] N. Bouayad-Agha, G. Casamayor, L. Wanner, Natural language generation in the context of the semantic web, Semantic Web 5 (2014) 493–513.

[2] Z. Mahlaza, C. M. Keet, A classification of grammar-infused templates for ontology and model verbalisation, in: Proceedings of the 14th International Conference on Metadata and Semantics Research, MTSR'19, volume 1057 of *CCIS*, Springer, 2019, pp. 64–76. 28-31 Oct 2019, Rome, Italy.

[3] B. Davis, R. Enache, J. van Grondelle, L. Pretorius, Multilingual verbalisation of modular ontologies using GF and lemon, in: Proceedings of the Third Workshop on Controlled Natural Language, CNL'12, 2012, pp. 167–184. 29–31 August 2012, Zurich, Switzerland.

[4] S. H. Lim, T. A. Halpin, Automated verbalization of ORM models in Malay and Mandarin, International Journal of Information System Modeling and Design 7 (2016) 1–16.

[5] I. Androutsopoulos, G. Lampouras, D. Galanis, Generating natural language descriptions from OWL ontologies: the NaturalOWL system, Journal of Artificial Intelligence Research 48 (2013) 671–715.

[6] N. Gruzitis, G. Nespore, B. Saulite, Verbalizing ontologies in controlled Baltic languages, in: I. Skadina, A. Vasiljevs (Eds.), Proceedings of the Fourth International Conference on Human Language Technologies - The Baltic Perspective, HLT 2010, volume 219 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2010, pp. 187–194. October 7-8, 2010, Riga, Latvia.

[7] C. M. Keet, M. Xakaza, L. Khumalo, Verbalising OWL ontologies in isiZulu with Python, in: The Semantic Web: ESWC 2017 Satellite Events, volume 10577 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 59–64. 30 May - 1 June 2017, Portoroz, Slovenia.

[8] Z. Mahlaza, C. M. Keet, OWLSIZ: An isiZulu CNL for structured knowledge validation, in: Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web, WebNLG+ 2020, Association for Computational Linguistics, 2020, pp. 15–25. 18 December 2020, Dublin, Ireland (Virtual).

[9] M. Jarrar, C. M. Keet, P. Dongilli, Multilingual verbalization of ORM conceptual models and axiomatized ontologies, Technical Report, Starlab, Vrije Universiteit Brussel, Belgium, 2006.

[10] B. Davis, A. A. Iqbal, A. Funk, V. Tablan, K. Bontcheva, H. Cunningham, S. Handschuh, Roundtrip ontology authoring, in: Proceedings of the 7th International Semantic Web Conference, ISWC 2008, volume 5318 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 50–65. October 26-30, 2008, Karlsruhe, Germany.

[11] Z. Mahlaza, C. M. Keet, Formalisation and classification of grammar and template-mediated techniques to model and ontology verbalisation, International Journal of Metadata, Semantics and Ontologies 14 (2020) 249–262.

[12] J. Byamugisha, C. M. Keet, B. DeRenzi, Evaluation of a Runyankore grammar engine for healthcare messages, in: Proceedings of the 10th International Conference on Natural Language Generation, 2017, pp. 105–113. September 4-7, 2017, Santiago de Compostela, Spain.

[13] C. M. Keet, L. Khumalo, Toward a knowledge-to-text controlled natural language of isiZulu, Language Resources and Evaluation 51 (2017) 131–157.

[14] C. M. Keet, L. Khumalo, Toward verbalizing ontologies in isiZulu, in: B. Davis, K. Kaljurand, T. Kuhn (Eds.), Proceedings of the 4th Workshop on Controlled Natural Language, CNL'014, volume 8625 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 78–89. August 20-22, Galway, Ireland.

[15] C. M. Keet, L. Khumalo, On the verbalization patterns of part-whole relations in isiZulu, in: Proceedings of the Ninth International Conference on Natural Language Generation, INLG'16, 2016, pp. 174–183. September 5-8, 2016, Edinburgh, UK.

[16] C. Chavula, C. M. Keet, Is *lemon* sufficient for building multilingual ontologies for bantu languages?, in: Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014), Co-Located with 13th International Semantic Web Conference on (ISWC 2014), Riva Del Garda, Italy, October 17-18, 2014., volume 1265 of *CEUR Workshop Proceedings.*, CEUR-WS.org, 2014, pp. 61–72. October 17-18, 2014, Riva Del Garda, Italy.

[17] C. Chavula, C. M. Keet, An Orchestration Framework for Linguistic Task Ontologies, in: Proceedings of the 9th International Conference on Metadata and Semantics Research, MTSR 2015, 2015, pp. 3–14. September 9-11, 2015, Manchester, UK.

[18] S. E. Bosch, T. Eckart, B. Klimek, D. Goldhahn, U. Quasthoff, Preparation and usage of Xhosa lexicographical data for a multilingual, federated environment, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, 2018.

[19] E. Simperl, M. Mochol, T. Bürger, Achieving maturity: the state of practice in ontology engineering in 2009, International Journal of Computer Science and Applications 7 (2010) 45–65.

[20] J. Oosthuysen, The Grammar of isiXhosa, SUN press, 2017.

[21] F. Katamba, Bantu Nominal Morphology, in: D. Nurse, G. Philippson (Eds.), The Bantu Languages, Routledge, 2014, pp. 103–120.

[22] N. N. Canonici, Elements of Zulu morpho-syntax, rev. ed., Zulu Language & Literature, University of Natal, Durban, 1995.

[23] P. Ciccarese, S. Peroni, The collections ontology: Creating and handling collections in OWL 2 DL frameworks, Semantic Web 5 (2014) 515–529.

[24] F. Gillis-Webber, S. Tittel, C. M. Keet, A model for language annotations on the web, in: B. Villazón-Terrazas, Y. Hidalgo-Delgado (Eds.), Proceedings of the First Iberoamerican Conference on Knowledge Graphs and Semantic Web, Springer, Cham, 2019, pp. 1–16. June 23-30, 2019, Villa Clara, Cuba.

[25] Y. Puzikov, I. Gurevych, E2E NLG challenge: Neural models vs. templates, in: E. Krahmer, A. Gatt, M. Goudbeek (Eds.), Proceedings of the 11th International Conference on Natural Language Generation, Association for Computational Linguistics, 2018, pp. 463–471. November 5-8, 2018, Tilburg University, The Netherlands.

[26] M. A. Musen, The protégé project: a look back and a look forward, AI Matters 1 (2015) 4–12.