Proceedings of the Linked Data in Linguistics Workshop @ LREC2022, pages 78–86
Marseille, 24 June 2022

Computational Morphology with OntoLex-Morph

Christian Chiarcos[1,2], **Katerina Gkirtzou**[3], **Anas Fahad Khan**[4],
Penny Labropoulou[3], **Marco Passarotti**[5], **Matteo Pellegrini**[5]
[1]Applied Computational Linguistics, Goethe University Frankfurt, Frankfurt am Main, Germany
[2]Institute for Digital Humanities, University of Cologne, Cologne, Germany
[3] Institute of Language and Speech Processing, Athena Research Center, Athens, Greece
[4]Istituto di Linguistica Computazionale "A. Zampolli", Consiglio Nazionale delle Ricerche, Italy
[5]Università Cattolica del Sacro Cuore, Milan, Italy
[1]chiarcos@cs.uni-frankfurt.de, [3]{katerina.gkirtzou,penny}@athenarc.gr,
[4]fahad.khan@ilc.cnr.it, [5]{marco.passarotti,matteo.pellegrini}@unicatt.it

Abstract

This paper describes the current status of the emerging OntoLex module for linguistic morphology. It serves as an update to the previous version of the vocabulary (Klimek et al. 2019). Whereas this earlier model was exclusively focusing on descriptive morphology and focused on applications in lexicography, we now present a novel part and a novel application of the vocabulary to applications in language technology, i.e., the rule-based generation of lexicons, introducing a dynamic component into OntoLex.

Keywords: OntoLex, computational morphology, inflection, derivation, compounding, finite state transducers

1. Background and Introduction

This paper describes the current status of the emerging module for linguistic morphology of the OntoLex vocabulary (Cimiano et al., 2016). It serves as an update to Klimek et al. (2019) and introduces a novel part of the vocabulary designed for rule-based generation of lexicons, introducing a dynamic component into OntoLex. The generation component is intended to allow for the dynamic generation of morphological variants of a single lexical entry; that is, it is intended to permit *intensional* as well as *extensional* morphological descriptions. In the latter kind of description all inflected forms of an entry (in the case of an inflected language) are explicitly listed; in the former, morphological information is given in a manner that allows individual forms to be generated dynamically.

Preliminary results in the development of this module have been published by Klimek et al. (2019), but, at the time, with a strict focus on extensional (descriptive) morphology and use cases from lexicography. Since then, we intensified research on intensional morphology and morphological generation and we now present the revised, consolidated model that has emerged. We consider the current draft to be near-final and would like to use this publication to elicit feedback from a broader audience before finalising it and publishing as a W3C Community Report akin to OntoLex-Lemon (Cimiano et al., 2016) and *lexicog*, the OntoLex module for lexicography (Bosque-Gil and Gracia, 2019).

The OntoLex-Lemon (core) model is illustrated in Fig. 1. It was foreseen in OntoLex that more detailed morphological information would be provided at a later point in time. In particular, the OntoLex core model includes the object property ontolex:morphologicalPattern, which, however, remained underspecified until a future module for morphology would have been created. OntoLex-Morph is the current prototype for this module.

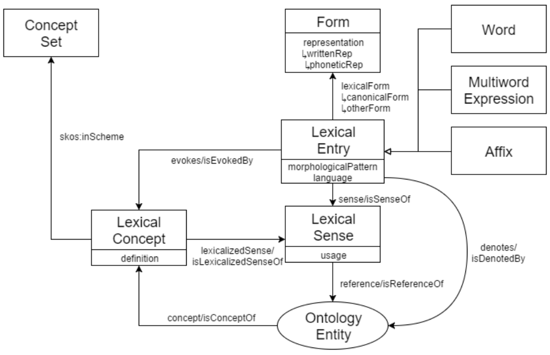

Figure 1: OntoLex-Lemon core model

In this paper, we first describe the OntoLex-Morph vocabulary (Sect. 2) and then elaborate on three use cases (Sect. 3) for inflection, word formation and compounding in three inflecting languages. When developing approaches for the computational application of OntoLex-Morph, we initially focused on inflecting languages with relatively rich morphology (in comparison to English). Section 4 summarises the main achievements, and presents the open issues currently under investigation. Finally, Sect. 5 gives an outlook towards publishing OntoLex-Morph as a W3C vocabulary, i.e., as Community Report of the W3C Community Group Ontology-Lexica, and thus, as a formal addendum to the OntoLex vocabulary.

2. OntoLex-Morph

The current version of OntoLex-Morph module is shown in Fig. 2.

Class **morph:Morph** is a subclass of `ontolex:LexicalEntry` that represents

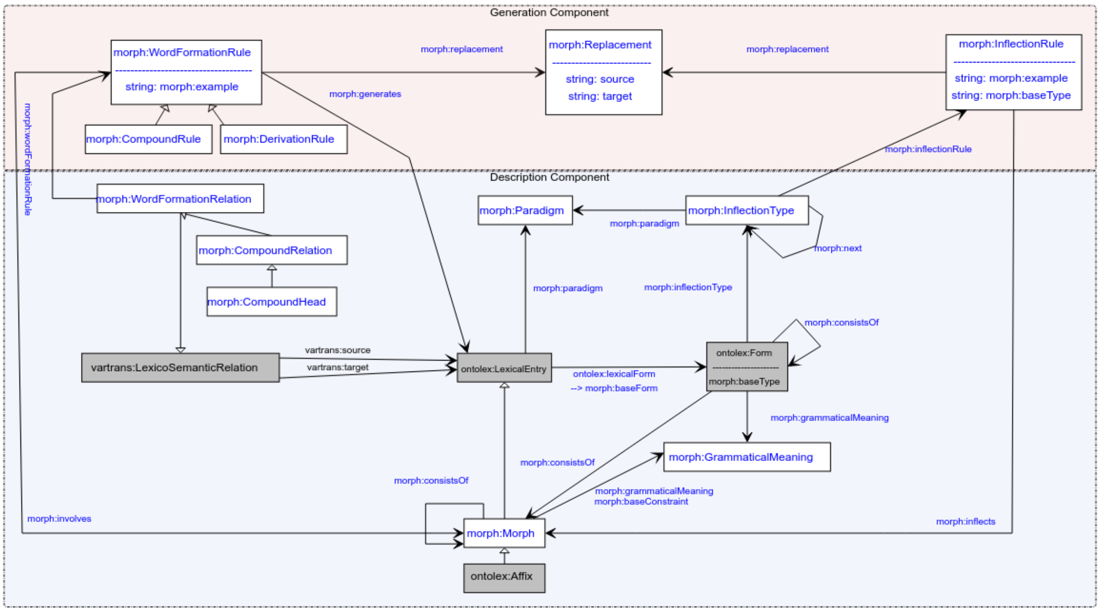

Figure 2: OntoLex-Morph vocabulary, draft version 4.13, April 2022

a concrete primitive element of morphological analysis. Morph is both a subclass of `ontolex:LexicalEntry` and a superclass of `ontolex:Affix`. Certain morphemes such as root, stem and zero morphs may not be affixes, but they are valid morphs; on the other hand, prefix, infix and suffix are being defined as subclasses of `ontolex:Affix` in the LexInfo vocabulary (Cimiano et al., 2011). Note that earlier versions of the vocabulary defined various subclasses of morph:Morph. As this classification is partially redundant with a number of designated classes for morpheme types in LexInfo, we suggest to, instead, extend LexInfo, accordingly.

With respect to intensional morphology, morphs are the core elements of the `WordFormationRules` and `InflectionRules` that `involve` them, resp., `inflect` accordingly. Both kinds of rules are illustrated by an `example` (a string, for descriptive morphology) or by a `replacement`, i.e., a pair of regular expressions that define preconditions and results of a replacement, similar to the `s///` operator in Perl.

In agglutinating languages, inflection may be represented by a *sequence* of morphemes, where morphs (resp., their rules) follow a specific order, e.g., gender before number morphemes. This is modelled by a series of `InflectionTypes` connected by the `next` property. Inflection types are part of a paradigm, but also, directly linked with forms generated by the associated inflection rules. As a necessary pre-condition for generating them, `Paradigm` can also be directly linked with a lexical entry. In existing grammars, rules do not always use the canonical form of a lexical entry as the basis, and in these cases, the corresponding base form can be marked by `baseForm`. Furthermore, lexical entries can group together forms generated from multiple base forms, and if so, the base form and the associated inflection rule can be marked for a `baseType` (a system-specific identifier). A form can then consist of other forms or morphs, and it can be characterized by a `GrammaticalMeaning`, i.e., a feature bundle of LexInfo properties or other annotations, e.g., gloss labels (the latter are not defined by OntoLex-Morph). Likewise, a grammatical meaning can also characterize a morph, e.g., the inflectional features expressed or part of speech and morphosyntactic features of a derived word. Furthermore, grammatical meanings can formulate `baseConstraints` in derivation, e.g., restrictions on the part of speech of the base that a particular affix can be used with.

Whereas inflection operates on forms, derivation is a *lexical* process. So, a `WordFormationRule` generates a `ontolex:LexicalEntry`, either by means of a `DerivationRule` or a `CompoundRule`. Whereas a word formation rule formulates or illustrates a general pattern, the lexico-semantic relation between two concrete lexical entries (the base and the derivation or a constituent word and the compound) is modelled as a `WordFormationRelation`. In compounding, the head can be marked by the subclass `CompoundHead`, if no head is explicitly marked, one can either use `CompoundRelation` or the decomp module of OntoLex (Cimiano et al., 2016).

3. Selected Use Cases

We illustrate inflection, word formation and morphological generation with three examples from computational linguistics and computational philology.

3.1. Inflection in Modern Greek

LEXIS (Anagnostopoulou et al., 2000) is a computational lexicon for Modern Greek intended for use in

NLP applications and modelled according to the PAROLE/SIMPLE model (Parole Consortium, 1996).

The basic unit in LEXIS is the Morphological Unit (MU), a single word with its assigned part of speech tag, which corresponds to the traditional notion of "lemma". Each MU is linked to Graphical Morphological Units (GMu), which correspond to orthographic variants of the lexical entry (e.g., "τραίνο" and "τρένο" [*train*]). Inflectional information is attached at the GMu level, in the form of an inflectional paradigm and a number of stems, each of which takes a number. The inflectional paradigm (GInP) is like an abstract "inflectional table", where each row (corresponding to an abstract/prototypical inflected wordform) is the combination of a numbered stem, a specific ending (suffix) and a bundle of grammatical features (e.g., case, number, person, tense, etc.). Full wordforms are not included in LEXIS; in principle, they should be produced with a generation algorithm that exploits co-indexing information in the entries of stems and inflectional paradigms.

For instance, the lemma ”άνϑρωπος” [*person*] is a common noun with two stems, the stems "άνϑρωπ-" and "ανϑρώπ-". These can be represented in Ontolex-Morph as `ontolex:Form` and related to the lemma via the `morph:baseForm` property. Each of them takes a number as a value for the property `morph:baseType`.

```
<anthropos>
    a ontolex:Word, morph:Morph ;
    rdfs:label "άνϑρωπος"@el, "person"@en;
    lexinfo:partOfSpeech lexinfo:noun ;
    morph:paradigm <efyvos_paradigm> ;
    morph:baseForm [
        a ontolex:Form ;
        ontolex:writtenRep "άνϑρωπ"@el ;
        morph:baseType "1" ] ;
    morph:baseForm [
        a ontolex:Form ;
        ontolex:writtenRep "ανϑρώπ"@el ;
        morph:baseType "2" ] .
```

Following the Ontolex-Morph model, each prototypical word form can be represented as a `morph:InflectionRule` which takes for the property `morph:baseType` a literal in the form of a number and a property `morph:replacement`, which combines together a `morph:source` and a `morph:target`, the latter being the ending and the former assumed to be derived from the `morph:baseType`.

```
 <inflRule_MaSgGe1>
    a morph:InflectionRule ;
    morph:baseType "1" ;
    morph:replacement [
        a morph:Replacement ;
        morph:source "$" ;
        morph:target "ου"@el ] .
```

The above `InflectionRule` can therefore generate the wordform:

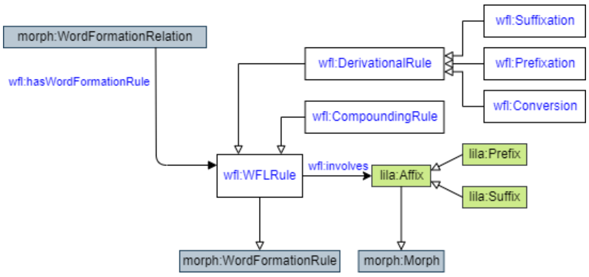

Figure 3: Architecture of the WFL ontology.

```
<anthropou1_form> a ontolex:Form ;
    ontolex:writtenRep "άνϑρωπου"@el .
```

3.2. Word Formation in Latin

Word Formation Latin (Litta and Passarotti, 2019, WFL) is a derivational lexicon of Latin characterised by a step-to-step morphotactic approach: each lexeme is connected to the lexeme from which it is directly derived (if any) via a word formation rule. This resource has been recently modelled in an ontology (Pellegrini et al., 2021) in order to include it into the LiLa Knowledge Base[1] of interoperable linguistic resources for Latin (Passarotti et al., 2020).

The proposed modelling is fully compatible with the architecture of OntoLex-Morph as described above, and integrated with it. At the moment, this is done by specifying subclass relations, after the final release of OntoLex-Morph, we might directly use OntoLex-Morph vocabulary. So far, OntoLex-Morph and the WFL ontology were developed in parallel, but with mutual influences on each other. For example, Fig. 3 illustrates the distinction between relations and rules, as it is applied both in WFL and in OntoLex-Morph. Each `ontolex:LexicalEntry` of the WFL ontology is linked to the one(s) it derives from and/or to the ones that derive from it by means of a specific instance of the class `morph:WordFormationRelation`. In turn, each `morph:WordFormationRelation` is linked through the property `wfl:hasWordFormationRule` to a specific `wfl:WFLRule`. Rules are then arranged in a hierarchy of subclasses that reflects the distinction made in WFL between derivation (prefixation, suffixation, conversion) and compounding. Rules are also connected with the `lila:Affix` they display (if any) by means of the property `wfl:involves`.

For instance, there is a `morph:WordFormationRelation` between *felix* 'happy' and *felicitas* 'happiness', that instatiates the specific `wfl:WFLrule` creating the latter from the former. This rule belongs to the class of suffixal rules creating deadjectival nouns, which is a subclass of `wfl:Suffixation`. The rule is also stated to involve the suffix *-tas*:

```
:li_103068 a ontolex:LexicalEntry ;
```

[1] `https://lila-erc.eu`.

```
    rdfs:label "felix" .

:li_103063 a ontolex:LexicalEntry ;
    rdfs:label "felicitas" ;

:r18023_li_103068_li_103063 a
  morph:WordFormationRelation ;
    vartrans:source :li_103068 ;
    vartrans:target :li_103063 ;
    wfl:hasWordFormationRule
:Derivation_Suffix_li_103068_To_li_103063.

:Derivation_Suffix_li_103068_To_li_103063
  a wfl:AdjectiveToNoun ;
  rdfs:label
"felix To felicitas involving -tas/tat" ;
  wfl:involves
<http://lila-erc.eu/data/id/suffix/24> .

<http://lila-erc.eu/data/id/suffix/24> a
  lila:Suffix ;
    rdfs:label "-tas/tat" .

wfl:AdjectiveToNoun rdfs:subClassOf
  wfl:Suffixation .
```

It is useful to spend a few words on the treatment of compounding in the WFL ontology. As can be seen below, compounds are modelled in the same way as other morphologically complex words, except that there are two relations: the one between the compound and its first constituent on the one hand, the one between the compound and its second constituent on the other hand. Both relations point to the same rule. The order of constituents is coded via a datatype property `wfl:positionInWFR`. This choice is motivated by the fact that the class `morph:WordFormationRelation` is a sub-class of `vartrans:LexicalRelation`, from which it inherits the requirement of having exactly one source and one target.

```
:li_88060 a ontolex:LexicalEntry ;
    rdfs:label "ager" .

:li_94916 a ontolex:LexicalEntry ;
    rdfs:label "colo" .

:li_88174 a ontolex:LexicalEntry ;
    rdfs:label "agricola" .

:r8833_li_88060_li_88174 a
  morph:WordFormationRelation ;
  rdfs:label "ager > agricola" ;
  wfl:positionInWFR 1 ;
  wfl:hasWordFormationRule
:Compounding_li_88060_li_94916_To_li_88174;
  vartrans:source :li_88060 ;
  vartrans:target :li_88174 .

:r8833_li_94916_li_88174 a
  morph:WordFormationRelation ;
    rdfs:label "colo > agricola" ;
  wfl:positionInWFR 2 ;
  wfl:hasWordFormationRule
:Compounding_li_88060_li_94916_To_li_88174;
    vartrans:source :li_94916 ;
    vartrans:target :li_88174 .
```

It has been mentioned above that compounding can also be modelled using the vocabulary of the Decomposition module of OntoLex. However, this option is not adequate to model WFL data. First, it is not desirable to be forced to use different vocabularies for word formation processes that are treated homogeneously in WFL – namely, OntoLex-Decomp for compounding and OntoLex-Morph for derivation and conversion. Second, OntoLex-Decomp does not allow to reify the relations between the lexical entries involved in compounds, but these relations are needed in order to provide a connection to the compounding rules present in WFL, as can be seen in the listing above.

This motivates the choice of giving the possibility of modelling compounding also using OntoLex-Morph, alongside OntoLex-Decomp. The choice between the two is left to the data creator, as it crucially depends on the nature and organization of the data themselves: if compounds are simply split into their different constituents, then OntoLex-Decomp will suffice; if additional information is provided and/or compounding is treated by means of full-fledged relations – as happens not only in WFL, but also in other important resources tackling derivational morphology, e.g. DeriNet 2.0 (Vidra et al., 2019) – then it will be possible (or even necessary) to resort to Morph.

3.3. Generation for German

Chiarcos et al. (accepted) recently described the application of OntoLex-Morph to convert and link various morphological resources for German. While the focus of this work was primarily on the encoding and integration of different types of morphological resources on a unified basis, the capacity to merge is a trivial (and intended) side-effect of RDF conversion and was the general purpose and original motivation of OntoLex-Morph (Klimek et al., 2019).

In this section, we focus on morphological generation rules resulting from converting an FST grammar, as this aspect was only superficially touched by Chiarcos et al. (accepted): We transform a German finite state transducer into OntoLex-Morph, the Stuttgart FST library with the SMOR grammar (Schmid, 2005) and the Morphisto lexicon (Zielinski et al., 2009). In order to replicate complete finite state transducers in OntoLex-Morph, we made use of the `morph:InflectionType` concept. In this conversion, every state is represented by an independent inflection type, and transitions between states are modelled by means of `morph:next`. For generation, we use a simple path traversal over these inflection types to retrieve a sequence of replacements. In this case, however, the traversal is not conducted as part of the actual generation process, but with the goal of achieving optimal run-time performance. Instead sequences of Perl-style replacements are compiled out, where regular expressions and capturing groups are used to emulate the effect of replacement operations associated with state transitions in the underlying transducer. The resulting sequences of replacement operations can subsequently be executed in any programming language that supports regular expressions. So, instead of doing morphological generation directly, they, instead, bootstrap a morphological generator from OntoLex-Morph. The operation needed to create a morphological generator from OntoLex-Morph inflection rules is a single SPARQL query that traverses the sequence of inflection types and collects a series of replacement operations as defined in the inflection rules:

```
SELECT DISTINCT ?itype ?transformation
WHERE {
 { SELECT ?a ?end ?pathid
   (GROUP_CONCAT(?repl; separator=";")
    AS ?transformation)
   WHERE {
    ?a a morph:InflectionType.
    ?a morph:next* ?b.
    ?b morph:inflectionRule ?rule.
    ?rule morph:replacement ?repl.
    FILTER(?repl != "s/$//")
    ?b morph:next* ?end.
    MINUS { ?end a morph:InflectionType;
                 morph:next [] }
  } GROUP BY ?a ?end
 }
 BIND(?a as ?itype)
}
```

This query uses a nested `SELECT` to aggregate from one inflection type to the sequence of following inflection types.[2] The result of this aggregation query, then, is a sequence of replacements with regular expressions that can be directly executed with Perl, or Sed, or transformed with minimal overhead to any other programming language that support Perl-style regular expressions (e.g., Java, Python, ... or even SPARQL). The actual generator is therefore a thin wrapper around this query to create a replacement script, whereas the OntoLex lexicon is not *directly* used for the transformation. The replacement script then reads lexical entries with their base forms (or, if these are not available, canonical forms), part of speech information (represented using the LexInfo property `lexinfo:partOfSpeech`) and paradigms. The replacement script assumes the presence of special symbols that trigger generation rules (e.g., `<Sg>` for singular number, etc.) In the current implementation, these are automatically created from an existing OLiA Annotation Model for the Morphisto morphology (Chiarcos, 2010).

Take the inflection of *zufällig* adj. 'random', this is marked as `lexinfo:partOfSpeech lexinfo:adjective`, so that one of the compatible annotation model concepts is `:ADJ_Pos_Fem_Nom_Sg`. As this carries `olias:hasTagEndingWith "<+ADJ><Pos><Fem><Nom><Sg>"`, and the base form is `zufällig`, this is concatenated into the input string `zufällig<+ADJ><Pos><Fem><Nom><Sg>` (which is paired with information about the associated lexinfo features). The associated paradigm in Morphisto is `:paradigm%23Adj%2B` from which we get to the inflection type `:type%23Adj%2B`. For this inflection type, the above query retrieved (among other possible paths) the replacement series illustrated in Fig. 4 (with replacement results for our input word added as comment).

While this works sufficiently well, Chiarcos et al. refrained from modelling morphophonological operations by means of this technology.[3] Instead, special symbols intended for resolution with two-level rules were simply omitted. The SPARQL query thus adds two additional replacements: The first marks forms that contain unresolved tags as being heuristic (insertion of initial $\star$), and the second removes all tags. The result string is `*zufällige` and (except for the unimplemented lower case rule), this is actually the correct form. However, morphophonological rules do at times have a profound impact on the surface form of an inflected word, e.g., the insertion of epenthetic vowels or assimilation effects. As a result, a certain number of hypothetical forms predicted by such replacements are indeed, ungrammatical. But even if they are grammatical, they need to be marked in the data. We thus suggest to introduce the novel class hypothetical form. This is necessary for Morphisto because the capturing of morphophonological alternations may imperfect. So, any form produced by the application of rules to a base or canonical form is hypothetical unless confirmed by external evidence from a dictionary or a corpus.

For inflection, the Morphisto generator produced over 400,000 triples for 41,100 hypothetical forms. For evaluation, we evaluate the generated hypothetical form by parsing them with the Ubuntu 20.4 package 'fst', and the Morphisto/SMOR grammar. Out of 25,859 different written representations of the generated hypothetical forms (excluding those identical to base forms), our generator achieved a precision of 78.5% against SMOR/Morphisto,[4] i.e., 20,308 of 5,551 written representations of hypothetical forms (exclud-

[2] The query is slightly simplified, we omit the creation of pathids.

[3] Technically, this would have been possible, but the application of a concept named 'inflection type' to assimilation rules would contradict linguistic intuitions associated with the term 'inflection type'.

[4] We do not calculate recall, as our conversion only encompasses the inflection component of of SMOR, and neither the derivation nor compounding rules that it also provides.

```
# from OntoLex-Morph
s/$/<FB>/;              # zufällig<+ADJ><Pos><Fem><Nom><Sg>
s/<+ADJ><Pos>//;        # zufällig<Fem><Nom><Sg>
s/<Fem><Nom><Sg>/e/;    # zufällige
s/$/<Low#>/;            # zufällige<Low#>

# remove special symbols that trigger morphophonological
# replacements
s/^\\(.*<\\)/\\*\\1/;   # *zufällige<Low#>
s/<[^>]*>//g;           # *zufällige
```

Figure 4: Automatically generated sequence of replacement operations retrieved from the OntoLex-Morph edition of SMOR/Morphisto, using the word *zufällig* ‘random’ with grammatical features as sample input

ing those identical to the base form) could be successfully parsed. As for the remaining 21.5%, these can be attributed to the insufficient support for morphophonological rules in OntoLex-Morph as well as invalid combinations of alternative base forms and inflection rules that are filtered out in SMOR in subsequent processing steps.

It is to be noted that vanilla morphological generation from OntoLex-Morph is a baseline functionality that has advantages in portability and sustainability, but that it lacks optimizations of FST, e.g., in disambiguation strategies and filtering conditions performed at the second level of two-level morphologies.

4. Discussion

The goal of this paper was to demonstrate to what extent the OntoLex-Morph vocabulary in its most recent edition can be used for modelling existing lexical resources concerned with or designed for computational morphological analysis and generation. This complements the work of Klimek et al. (2019) who discussed applications of OntoLex-Morph for descriptive morphological analysis in the realm of digital lexicography with a more technically oriented perspective. In particular, we aimed to evaluate its applicability to broad band-width of use cases in this domain, illustrated here for three representative resources.

4.1. Achievements

Providing morphological datasets as OntoLex and in RDF provides the natural benefits of linkability, in this paper, we thus focus on the *coverage* of OntoLex-Morph for representative use cases, focusing on language technology resources for inflectional morphology (for Greek), derivation and compounding (for Latin) and the general usability for morphological generation (for German). By focusing on existing resources in three different languages, we also expect a certain degree of heterogeneity in the requirements.

Linkability and (Re-)Usability Overall, using OntoLex and OntoLex-Morph for machine-readable dictionaries and morphological resources has the great advantage that these can be trivially linked, merged and integrated. This is a general characteristic of RDF and LLOD technology and to establish a community standard that facilitates such integration operations over legacy as well as digital-born data has been the initial motivation for developing OntoLex and OntoLex-Morph. Unsurprisingly, this has been repeatedly confirmed since, e.g., for lexical resources and knowledge graphs (McCrae et al., 2011), lexical resources with other lexical resources (Eckle-Kohler et al., 2015), lexical and morphological resources (Racioppa and Declerck, 2019) and morphological resources with other morphological resources (Chiarcos et al., accepted). We thus consider the benefit of *linkability* for morphological resources to be sufficiently established by earlier research – as well as the benefits that this entails with respect to representation and modelling (graphs can represent any linguistic data structure), structural and conceptual interoperability (generic data structures, shared vocabularies, uniform access protocol), federation (querying over distributed data), dynamicity (access remote resources at query time) and the availability of a mature technical ecosystem (Chiarcos et al., 2013; Cimiano et al., 2020). But these benefits are inherent to LLOD and not specific to OntoLex-Morph, so we did not specifically evaluate them.

Applicability Overall, we found that the OntoLex-Morph vocabulary was applicable to the resources addressed in this paper with relative ease. Although we encountered a number of borderline cases in which the current modelling leaves up either challenges or desiderata (see below), the typical cases could be represented in OntoLex-Morph, for inflection (Sect. 3.1), for word formation (Sect. 3.2) and for morphological generation in general (Sect. 3.3). We used the experiences we made while applying the OntoLex-Morph vocabulary on novel data and questions that were raised in the process to refine and clarify the current model draft.[5]

Rule-based generation In Sect. 3.3, we described how OntoLex-Morph resources can be used to bootstrap replacement scripts that emulate finite state trans-

[5] https://github.com/ontolex/morph/blob/master/draft.md

ducers by means of regular expressions. This is only a baseline functionality as aspects of morphophonology have not been addressed, but only "deep" morphology, but it was nevertheless successful in achieving a considerable degree of precision with a formalism (Perl-style regular expressions) that can be easily ported into any programming language, whereas the original FST grammar depended on a 2005 library (Schmid, 2005).

4.2. Challenges

Variation in inflection Another challenge which we are focusing on as part of the development of OntoLex-Morph is the representation of variants. This occurs, for instance, when more than one form realises the same cell in an inflection table for a given paradigm; this is also known as *overabundance* (Thornton, 2019). This can be due to dialectal, diachronic or simply orthographic variation. It is more common to have such variants in the case of languages without a standardised orthography and especially historical languages such as Old English. Indeed, it is not difficult to find examples in the latter, e.g., the first person preterite indicative form of the verb *cuman* 'to come' is often listed as both *cwom* and *com*. Overabundance is also widely attested in Latin data, where especially interesting are cases of lexemes that display variation between forms that belong to different inflection classes, for instance LAVO 'wash', that can be inflected according to either the 1st (e.g. PRS.ACT.INF *lavare*) or 3rd (e.g. PRS.ACT.INF *lavere*) conjugation. We are thus clearly dealing with morphological (rather than simply orthographic) variation. A current challenge is to find a systematic way of dealing with these cases that is compatible with the generative component of OntoLex-Morph. A related problem is suppletion, i.e., cases in which different forms of the same lexical entry are formed from different etymological roots. This is the case of the Old English verb *wesan* 'to be' whose infinitive represents one underlying root, whereas its indicative present singular forms are based on *two* other roots (*eom* 1.sg. '(I) am'; *bist* 2.sg. '(you) are'). This pattern is also preserved in modern English, and with once-regular morphological processes getting increasingly intransparent over time, has even expanded to form novel pairs of 'irregular' forms that appear to operate with different stems, e.g., in verbs like *bring* and *think*, whose nasal complement was lost in the past forms *brought* and *thought* after Germanic *-kt-* shifted to Old English *-ht-*. The same pattern is also observed in modern Greek, where alternative wordforms for the same grammatical meaning co-exist. Alternatives may be associated with alternative endings, e.g. πατέρ-ες and πατερ-άδες or alternative stems, as in the example listed in section 3.1, i.e. άνθρωπ-ου and ανθρώπ-ου. One of the forms may be marked as a dialectal, archaic, more formal, or colloquial variant but there are also cases where the two forms are just alternatives; such a case is that of contracting verbs, e.g. αγαπ-άω and αγαπ-ώ.

Phonological processes As mentioned in Sect. 3.3, only rules concerned with 'deep morphology' have been formalized, but not morphophonological processes that deal with phonological processes like assimilation or apophony, i.e., the second level in classical two-level morphologies. A particular problem here is that, at least in word formation in Latin, these are not fully predictable, and this prevents the simple juxtaposition of formative elements from generating the actual surface form of derivatives.

Markers of morphological variation When modelling linguistic variation at the morphological level, we are faced with the need for attributing markers (labels of style, dating, dialect, etc.) to wordforms, in the same way that traditional dictionaries assign them to lemmas. That is, as we have archaic, older, dialectal, formal lemmas, we also have inflectional variants that can be marked. For instance, in the example (Sect. 3.1), the form άνθρωπου is used in a more informal context compared to ανθρώπου. In modern Greek, a lot of dual wordforms originate from "katharevoussa"[6]. It remains an open question whether and how these markers would be modelled within the morph module in a uniform and generic way, and specifically in inflection rules so that a mechanism could be triggered to copy these markers (together with grammatical features) to the generated written forms as well, while keeping the model simple.

At the moment, we would consider such markers to be beyond the scope of OntoLex-Morph. It is, of course, necessary for successfully generating context-adequate forms, but we would see the individual attributes and features more in the general scope of the LexInfo vocabulary. Indeed, LexInfo provides a rudimentary vocabulary, e.g., with `lexinfo:register` and values such as `lexinfo:dialectRegister`, with `lexinfo:temporalQualifier` and values such as `lexinfo:archaicForm`, or with `lexinfo:dating` and values such as `lexinfo:old`. Neither of these terms fits katharevoussa directly, but, in fact, a language-specific instance of `lexinfo:Register` or `lexinfo:TemporalQualifier` could also be created – unless the data providers decide to live with the imprecision of standard LexInfo terminology. However, what is important with regard to OntoLex-Morph is that it must provide the necessary prerequisites for adding such markers to morphologically relevant data structures, (morphological rules, lemmas, forms, etc.), i.e., they must be concepts, not properties. And, indeed, this is the case already. But even in this case, it would be desirable if the OntoLex-Morph vocabulary would eventually be accompanied by best practice recommendations for the assignment of markers and provenance.

[6] Katharevoussa is an archaic form of Greek constructed on the basis of the Attic dialect and used in formal settings; although its use is fading, it is still encountered in older texts.

4.3. State of Modelling

The OntoLex-Morph diagram has changed significantly since (Klimek et al., 2019), but only few vocabulary elements have changed their definition.[7] We thus consider the vocabulary stable, and revisions are now limited to cases when a change in the vocabulary meets the needs of *multiple* data providers or potential users. Selected suggested revisions include the revision of inflection type and the extension of LexInfo.

Inflection type An aspect that is still under discussion, as it can pose non-trivial problems when modelling data with this module, concerns the class morph:InflectionType. Since it was intended to account for the different slots available for values of different morphosyntactic properties in agglutinative languages, such problems emerge especially in fusional languages like Greek and Latin, where there are no such slots and the different values are expressed cumulatively by means of the same affix.

In Latin – like in many other languages – inflection rules are sensitive to inflection class distinctions: for instance, the rule to obtain the PRS.ACT.IND.2SG from the infinitive of 1st conjugation verbs (e.g. *amare* → *amas* 'to/you love') is different than the one of 3rd conjugation verbs (e.g. *dicere* → *dicis* 'to/you say'). Inflection classes can easily be coded as instances of the class morph:Paradigm. However, given this state of affairs it could be useful to have a property linking each morph:Paradigm to all the morph:InflectionRules it consists of, without having to go through morph:InflectionType as required in the current draft.

As the inflection type class has been created for agglutinating, not inflecting languages, it is unsurprising that it seems to be unnecessary here, and could be replaced by a direct link to inflection rule. At the same time, we suggested a novel application of inflection type to encode finite states, and it was mostly terminological issues that kept us from modelling morphophonological processes with 'inflection type', so that we suggested to model the order of morphemes as a sequence of inflection rules, instead, as their naming is less confusing.

A possible revision that would cater all three requirements would be to eliminate inflection type completely, i.e., to transfer all its properties to inflection rule, to connect grammatical meaning with inflection rule, and to encode the information that inflection type was originally meant for (position of a morphological 'slot' and its characteristics) as part of GrammaticalMeaning. This modelling, however, needs to be evaluated for its application to agglutinative languages and the original intended application of inflection type to represent morphological 'slots'.

LexInfo A number of suggested additions to LexInfo have been mentioned throughout this paper. This includes the introduction of additional subclasses of ontolex:LexicalEntry and morph:Morph to complement the classes lexinfo:Suffix, lexinfo:Prefix and lexinfo:Infix that LexInfo currently defines as subclasses of ontolex:Affix. In addition to subclasses of ontolex:Affix, we would require lexinfo:RootMorph and lexinfo:StemMorph as subclasses of morph:Morph, resp., ontolex:LexicalEntry.

A possible addition to LexInfo is in subproperties and object values of ontolex:usage, where morphological resources call for introducing object values such as lexinfo:hypotheticalForm (or, lexinfo:nonattestedForm), lexinfo:reconstructedForm and lexinfo:incorrectForm, which can be modelled in analogy to the properties lexinfo:register, and lexinfo:domain by means of a property lexinfo:evidence.

5. Summary and Outlook

In this paper, we described the recent extension of OntoLex-Morph with respect to computational morphology, and in particular, vocabulary elements necessary for describing morphological generation by means of rules, forms and morphs. This paper complements our earlier work on OntoLex-Morph (Klimek et al., 2019) that took a stronger focus on requirements from lexicography and the language sciences, and with the recent extensions, the overall structure of the vocabulary has been considerably extended. Taking the results of both papers together, we cover two major strands of use cases for an OntoLex Morphology module, so that after more than five years of development within the W3C Community Group Ontology-Lexica, the OntoLex-Morph vocabulary can now be considered relatively mature and stable.

Despite the advanced state of affairs after five years of development in this community, there are some limitations as pointed out in Sect. 4 that we plan to address in the next months. After having demonstrated that we cover requirements from both lexicography and language technology, we will work on consolidating the OntoLex-Morph vocabulary in order to prepare its final publication, probably in 2023. The primary goal of this paper and our presentation is two-fold: On the one hand, it documents the recent extensions, and on the other hand, it aims to elicit feedback from reviewers and audience to take into account before publishing it as a W3C vocabulary in the form of a community report of the W3C Ontology-Lexica Community Group.

[7] The most significant change in the overall model is that we now define morph:Morph as a subclass of lexical entry rather than as an independent concept, so that the existing ontolex:Affix class can now be interpreted as a subclass of morph:Morph and that the redundancy between ontolex:Affix and morph:AffixMorph is eliminated.

6. Acknowledgements

The research described in this paper was conducted in the context of the Cost Action CA18209 *Nexus Linguarum. European network for Web-centred linguistic data science*, with partial support from the Research Group *LiODi. Linked Open Dictionaries* (funded by the German Federal Ministry of Education and Science, BMBF, 2015–2022), from the ERC Horizon 2020 research and innovation programme via the projects Prêt-à-LLOD (grant agreement no. 825182), and Linking Latin (grant agreement no. 769994) and from "APOLLONIS: Greek Infrastructure for Digital Arts, Humanities and Language Research and Innovation" (MIS 5002738), (co-financed by Greece and the EU). Moreover, the authors would like to thank Bettina Klimek and Maxim Ionov for coordinating the development of OntoLex-Morph in the last years, as well as all OntoLex-Morph contributors.

7. Bibliographical References

Anagnostopoulou, D., Desipri, E., Labropoulou, P., Mantzari, E., and Gavrilidou, M. (2000). Lexis - Lexicographical Infrastructure: Systematising the Data. In *Proceedings of the International Workshop on Computational Lexicography and Multimedia Dictionaries (COMPLEX 2000)*, Patras, Greece.

Bosque-Gil, J. and Gracia, J. (2019). The Ontolex Lemon Lexicography Module. Final Community Group Report. Technical report, W3C.

Chiarcos, C., McCrae, J., Cimiano, P., and Fellbaum, C. (2013). Towards Open Data for Linguistics: Linguistic Linked Data. In *New Trends of Research in Ontologies and Lexical Resources*, pages 7–25. Springer.

Chiarcos, C., Fäth, C., and Ionov, M. (accepted). Unifying Morphology Resources with Ontolex-Morph. a Case Study in German. In *13th International Conference on Language Resources and Evaluation (LREC-2022)*, Marseille, France, June.

Chiarcos, C. (2010). Towards robust multi-tool tagging. an OWL/DL-based approach. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 659–670.

Cimiano, P., Buitelaar, P., McCrae, J., and Sintek, M. (2011). Lexinfo: A declarative model for the lexicon-ontology interface. *Journal of Web Semantics*, 9(1):29–51.

Cimiano, P., McCrae, J., and Buitelaar, P. (2016). Lexicon Model for Ontologies: Community Report. Technical report, W3C.

Cimiano, P., Chiarcos, C., McCrae, J. P., and Gracia, J. (2020). *Linguistic Linked Data*. Springer.

Eckle-Kohler, J., McCrae, J. P., and Chiarcos, C. (2015). lemonUby – a large, interlinked, syntactically-rich lexical resource for ontologies. *Semantic Web*, 6(4):371–378.

Klimek, B., McCrae, J. P., Bosque-Gil, J., Ionov, M., Tauber, J. K., and Chiarcos, C. (2019). Challenges for the Representation of Morphology in Ontology Lexicons. *Proceedings of eLex*, pages 570–591.

Litta, E. and Passarotti, M. (2019). (When) inflection needs derivation: a word formation lexicon for Latin. In Nigel Holmes, et al., editors, *Lemmata Linguistica Latina. Volume 1. Words and Sounds*, pages 224–239. De Gruyter, Berlin, Boston, December.

McCrae, J., Spohr, D., and Cimiano, P. (2011). Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In *Extended Semantic Web Conference*, pages 245–259. Springer.

Parole Consortium. (1996). Morphosyntactic specifications : Language Specific Instantiations. Technical report, LE-PAROLE report.

Passarotti, M., Mambrini, F., Franzini, G., Cecchini, F. M., Litta, E., Moretti, G., Ruffolo, P., and Sprugnoli, R. (2020). Interlinking through lemmas. The lexical collection of the LiLa knowledge base of linguistic resources for Latin. *Studi e Saggi Linguistici*, 58(1):177–212.

Pellegrini, M., Litta, E., Passarotti, M., Mambrini, F., and Moretti, G. (2021). The Two Approaches to Word Formation in the LiLa Knowledge Base of Latin Resources. In *Proceedings of the Third International Workshop on Resources and Tools for Derivational Morphology (DeriMo 2021)*, pages 101–109.

Racioppa, S. and Declerck, T. (2019). Enriching Open Multilingual Wordnets with Morphological Features. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019).*

Schmid, H. (2005). A Programming Language for Finite State Transducers. *Finite-State Methods and Natural Language Processing FSMNLP 2005*, page 50.

Thornton, A. M. (2019). Overabundance: A Canonical Typology. In *Competition in Inflection and Word-Formation*, pages 223–258. Springer.

Vidra, J., Žabokrtskỳ, Z., Ševčíková, M., and Kyjánek, L. (2019). Derinet 2.0: Towards an All-in-One Word-Formation Resource. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 81–89.

Zielinski, A., Simon, C., and Wittl, T. (2009). Morphisto: Service-Oriented Open Source Morphology for German. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 64–75. Springer.