Lemmatization and morphological analysis for the Latin Dependency Treebank

Giuseppe G.A. Celano

Abstract

The present article presents some challenges posed by lemmatization and PoS tagging of Latin, with reference to the ongoing work to revise the Latin Dependency Treebank. Current options available for lemmatization and morphological analysis of Latin are reviewed and discussed. The pipeline to annotate the morphological layer of the Latin Dependency Treebank is shown to consist in three main steps: (i) tokenization/ sentence split, which is performed via a documented rule-based algorithm, (ii) prepopulation by means of COMBO, a state-of-the-art joint lemmatizer, PoS tagger, and parser trained on the data of the Latin Dependency Treebank 2.1, and (iii) manual error correction informed by the attempt to identify and document lemmatization and morphology annotation rules.

Keywords: Latin Dependency Treebank, lemmatization, PoS tagging.

1. *Introduction*

Lemmatization is, in computational linguistics, a task which is commonly considered part of the morphological layer of annotation because of the strong interrelationship between it and PoS tagging (including morphological features identification)[1], all of them concerning the forms a given word can take on the basis of its function in a clause[2].

[1] A note on the terminology I use in the paper. The expression 'morphological analyzer' is used to mean a program outputting morphological analyses for tokens out of context (e.g. the neuter noun *bellum* receives three analyses, the nominative, accusative, and vocative, which share the same word form). The expression 'morphological analysis' is usually used with reference to a morphological analyzer. On the contrary, I use 'PoS tagger' to mean a statistical tagger outputting one single analysis for each token depending on the context; 'PoS tagging' can however also be used in a general sense, i.e. with or without reference to a PoS tagger. The term 'lemmatizer' is used for programs providing lemmas for tokens both in context and out of context.

[2] Both lemmatization and PoS tagging crucially depend on tokenization, as is shown in Section 4.2.

Received: January 2020
Accepted: May 2020

More precisely, lemmatization can be defined as consisting in the assignment of an 'ID word form' to a set of word forms sharing the same 'base' or 'root' and the same 'part of speech'. An example is the Latin verb *collaboro* ("I collaborate") as the lemma for all the verb word forms sharing the base *collabor*, such as *collaboravit*, *collaboravissemus*, *collaborare*, *collaboratum*, and so forth. Similarly, the Latin noun *dux* is the lemma for all the noun word forms whose base/root is *duc*, such as, for example, *duci*, *ducem*, or *duces*.

Very often, morphologically related word forms share the same 'root', but have different bases. For example, third conjugation verbs, such as *cado*, *is*, *cecidi*, *casum*, *cadere*, typically present different stems. Latin nouns, such as *artifex*, *icis*, can also show vowel changes between the nominative and all other cases.

It is important to note that the choice of a given word form as the ID for its morphologically related word forms is arbitrary/conventional. In Latin, for example, the first person singular of the indicative present is chosen as a lemma (such as *collaboro* above), even though any other related verb word form could in theory be chosen. Indeed, the infinitive form of a verb is commonly used, for example, in Italian, to serve the same ID function as the indicative present in Latin.

It is as important to note that the inventory of parts of speech is also, at least to a certain extent, rather arbitrary/conventional[3]. In Latin, for example, the participle could be considered as an independent part of speech because of its peculiar morphosyntactic proprieties, which, as the etymology of the name itself reveals (*particeps*, i.e. it takes part in the nature of both verb and noun) set it apart from other verb forms.

In lexicography/traditional grammar, lemmas correspond to dictionary entries. Crucially, such entries commonly correspond to more than a single word form: for example, the dictionary entry/lemma for the above mentioned verb *collaboro* is *collaboro*, *collaboras*, *collaboravi*, *collaboratum*, *collaborare*, i.e. it contains, besides the first person singular of the indicative present, also the second person singular of the indicative present, the first person singular of the indicative perfect, the supine, and the infinitive. All these forms provide full morphological information about the verb, because all relevant verb stems are provided, which allow analysis/identification of any word form of the verb.

Dictionary lemmas are most useful because Latin verbs belonging especially to the third conjugation can show unpredictable verb stems: for ex-

[3] How problematic definition of parts of speech is is made particularly clear in typological studies (see, among many others, Haspelmath, 2012 and Sasse, 2001).

ample, the dictionary entry for *capio* (“I take”) is *capio*, *capis*, *cepi*, *captum*, *capere*, where the stems for the perfect, supine, and even the infinitive are not as regular as, for example, those of most verbs of the first conjugation. The lemma provided in a dictionary entry is, therefore, aimed not only to function as an ID for the set of its morphologically related word forms, but also to provide full information for its conjugation/declension.

On the contrary, a lemma in treebanking conventionally consists only in the first word form of its corresponding dictionary entry. This has a significant impact on further automatic processing of a given token. For example, the lemmas for the word forms *lupum* and *exercitum* are *lupus* and *exercitus*, respectively. Without knowing their corresponding dictionary lemmas (i.e. *lupus*, *lupi* and *exercitus*, *exercitus*), it is impossible to correctly decline them, even if one takes their morphological analyses into account: indeed, they are both masculine, singular, and accusative nouns.

The information concerning their kind of declension (i.e. I decl. vs IV decl.), which is necessary to correctly decline them, is simply missing in the annotation available within treebanks[4]. This deficiency is even more apparent when it comes to verbs: it is not possible to infer all verb stems from a given word form such as, for example, *ausum* (whose lemma would be *audeo*). Apart from most first conjugation verbs, there is no way to automatically infer all verb stems from single word forms, many of them being potentially able to belong to different conjugations.

As is well known, lemmatization is of crucial importance for many natural language processing tasks, such as summarization, topic modeling, and, more in general, any kind of semantics-oriented research, in that it allows reduction of the variety of word forms available in a text, with consequent increase of machine learning algorithms’ performance[5].

In the present article, I will review (some of) the resources available for Latin lemmatization and morphological analysis in Section 2. In Section 3, I draw attention to a few challenges in Latin lemmatization and morphological analysis. In Section 4, I show the current approach to lemmatization and morphological analysis/PoS tagging for the Latin Dependency Treebank. Section 5 contains some concluding remarks.

[4] For an introduction on the treebanks I will mention in the present article, see Celano (2019b) and references therein.

[5] An interesting, recent example of the potential of lemma information for Latin research is presented in Sprugnoli *et al.* (2019).

2. *An overview of Latin lemmatizers and morphological analyzers*

There exist many lemmatizers/morphological analyzers for Latin nowadays, and their number is likely to grow due to the increasing availability of digitized texts/corpora and accessibility of machine learning techniques. I will show in the present section (some of) the most known/remarkable ones[6].

Lemmatizers/morphological analyzers can be evaluated along different dimensions. For example, their coverage of the Latin vocabulary varies. A systematic comparison of all of them is missing, but Springmann *et al.* (2016) provide evidence[7] that *LatMor*[8] (Springmann *et al.*, 2016) and *LemLat*[9] (Passarotti *et al.*, 2017) can recognize many more types/tokens of Classical and Medieval Latin than *PROIEL*[10], Parsley[11], Words[12], and Morpheus[13].

Some lemmatizers/morphological analyzers seem to have been primarily created for human, rather than machine, consumption. For example, both Words and Collatinus[14] can be queried via HTML interfaces or desktop applications, which make them useful especially for traditional scholarship. Words could also be queried automatically because word forms to analyze are contained in URLs[15]: however, the output is a simple HTML page providing no structure for its morphological analyses, so automatic extraction is not immediate. The sources for its more than 39,000 entries seem to derive from the *Oxford Latin Dictionary* and Lewis and Short[16].

Collatinus[17] is based on: Lewis and Short (1879), Gaffiot (2016), Du Cange (1883), Georges (1913-1918), Jeanneau (2017), Gaffiot (1934),

[6] Gleim *et al.* (2019) have recently trained a few PoS taggers and lemmatizers for Latin, using data from *PROIEL* and Capitularia. They run a number of interesting experiments, including testing how well a model can perform on a different kind of corpus.

[7] These results are in line with the ones in Gleim *et al.* (2019: 19).

[8] See *http://www.cis.uni-muenchen.de/~schmid/tools/SFST/*.

[9] I always refer to *LemLat* 3.0: *http://www.lemlat3.eu/*.

[10] See *https://github.com/mlj/proiel-webapp/tree/master/lib/morphology*.

[11] See *https://github.com/goldibex/parsley-core*.

[12] See *http://archives.nd.edu/words.html*.

[13] See *https://github.com/tmallon/morpheus*.

[14] See *https://outils.biblissima.fr/en/collatinus*.

[15] An example for *amoris* is *https://archives.nd.edu/cgi-bin/wordz.pl?keyword=amoris*.

[16] I could not find more precise references for the dictionaries on *https://mk270.github.io/whitakers-words/plan.html* [accessed on 30.11.2019].

[17] The references for the source dictionaries which follow coincide with the bibliographically incomplete ones given on the website *https://outils.biblissima.fr/en/collatinus/#downloads* [accessed on 30.11.2019]. To interpret them, the reader is referred to the weblink, from where the relevant resources can be downloaded.

Calonghi (1898), Valbuena (1819), Quicherat (1836). Its last version (11.2) is claimed to contain more than 80,000 lemmas. Notably, Collatinus[18] also outputs information for syllable length and lemmas are provided in their full form, i.e. in the way they can be found in printed dictionaries (the latter feature is present also in Words). The underlying data is available on GitHub[19], but an API for computer consumption is not provided.

Some lemmatizers/morphological analyzers, such as Morpheus and *LemLat* are especially known for their use in treebanking[20]. Morpheus is the morphological analyzer/lemmatizer used for the Ancient Greek and Latin Dependency Treebank (it will be introduced in Section 4).

LemLat shares the same annotation scheme with the Index Thomisticus Treebank. Even though *LemLat* is one of the oldest lemmatizers/morphological analyzers for Latin, its source code and data have been made open much later[21] (which impacted its exploitation in other projects). It consists in a rule-based morphological analyzer, which depends on a MySQL database containing the data for lemmas/morphological forms.

The internal workings are described in the corresponding documentation[22]. It can be queried within a standalone application, which outputs morphological analyses and lemmas for each word form given as an input. Notably, *LemLat* provides a segmentation for each word analyzed, which distinguishes bases from endings (this information is provided also in Words). The possibility to download the entire database as a MySQL dump guarantees even more query flexibility[23]. The database is based on Georges (1913-1918), Glare (1968-1982), and Gradenwitz (1904), which together amount to 40,014 lemmas, and *Totius Latinitatis Onomasticon* (26,415 lemmas; see Passarotti *et al.*, 2017 for more details).

LatMor is a finite-state morphological analyzer which parses Latin words and returns their morphological analyses, lemmas, and, notably, even vowel quantities. It is accessible at the command line, after the SFST

[18] I refer to the version available online [accessed on 30.11.2019].

[19] See *https://github.com/biblissima/collatinus/tree/master/bin/data*.

[20] A backoff Latin lemmatizer based on the data of the Latin Dependency Treebank is available in *CLTK*: *http://docs.cltk.org/en/latest/latin.html*.

[21] More precisely in 2016, if one follows the date of creation for the corresponding GitHub repository: *https://api.github.com/respos/circse/lemlat3*.

[22] See *https://github.com/CIRCSE/LEMLAT3*.

[23] Because of the complexity of the rules governing the merging of the morphological forms contained in the many MySQL tables, a desideratum for the future is rearranging the content of the database and publish it also in other formats.

tools have been installed. It is based on the lemmas found in Georges (1913-1918) and additions from Lewis and Short (1907); it contains about 70,000 lemmas.

There are a few problems affecting all the above mentioned lemmatizers/morphological analyzers. All of them cannot communicate among each other without proper conversion of morphological labels, since they are all different[24]. The annotation schemes are, in general, similar, but there are still differences, which require attention. For example, *ubi* is classified as 'invariable' in *LemLat*, but as 'adverb' or 'conjunction' in *LatMor*.

Another remarkable problem is that each lemmatizer/morphological analyzer joins together lemmata of more than one dictionary on the assumption that there is consistency across all the resources as to the criteria employed to identify lemmata. This probably holds true in general (also because of the known interdependencies among the original printed editions), but it is still unknown to what extent exactly.

One technical limitation of all the lemmatizers/morphological analyzers is that they cannot analyze multiword expressions, such as passive forms: for example, the expression *amatus est* cannot be given as an input and analyzed as a perfect passive indicative, but it has to be split into *amatus* (i.e. 'perfect passive participle') and *est* ('present indicative'). This is unfortunately an unsolved problem also affecting treebanking, where tokenization typically allows splitting but not merging of two graphic words, and therefore multiword tokens such as *amatus est* can be annotated only by means of specific syntactic labels[25].

Lastly, none of the lemmatizers/morphological analyzers provides a community-based mechanism allowing editing of the databases, which could guarantee corrections and extension. Most resources do not make the underlying database open or easily accessible; when the database is available (such as those of *LemLat* or Collatinus), their formats do not allow editing easily.

[24] Differences in orthography may also apply.

[25] The inability of properly analyzing multiword expressions in treebanking heavily depends on the fact that tokenization and morphosyntactic annotation are not commonly added standoff: inline annotation makes it difficult to express splitting and merging of graphic words, while trying to keep markup in a file relatively simple and easy to understand (and process). For a proposal of standoff annotation for Latin see Celano (2019a).

3. *Challenges for (Latin) lemmatization and morphological analysis*

There exist challenges concerning lemmatization and morphological analysis for Latin (as well as other languages), which especially pertain to the computational nature of these tasks.

As was shown in Section 2, most lemmatizers/morphological analyzers rely on information contained in more than one printed dictionary. This raises the question of which criteria were employed (i) to identify lemmas and – for the purposes of morphological analysis – (ii) to assign them a part of speech.

These two problems particularly affect digital resources because they should strive to ensure as much consistency as possible, any automatic data processing crucially relying on it. Indeed, while consistency is also desirable in printed dictionaries, it seems reasonable to claim that the specific purpose for which they were created (i.e. human consumption) may allow for accommodation of a number of 'irregularities', which, on the contrary, impinge on computational resources derived from them, but designed for machine consumption.

This is particularly clear when it comes to deciding about the part of speech for a given word: for example, *hiberna* could be analyzed as a substantivized adjective and therefore subsumed under the adjectival lemma *hibernus*, *a*, *um* or considered as a separate noun, and therefore assigned the separate entry *hiberna*, *orum*. Some printed dictionaries opt for the second solution, but it is not completely clear why: on the one hand, *hiberna* seems to occur so frequently as to be able to be recognized as belonging to an independent – although related – lemma; on the other, neuter adjectives can be regularly substantivized in Latin, but many/most of them are subsumed under their corresponding adjectival lemmas (similarly, *Romani* is, for example, found under *Romanus*, *a*, *um*).

Strictly connected to the question of substantivized adjectives is that of participles. Participles are regularly assigned the part of speech 'verb', even though, as is well known, they can serve different functions within a clause. Classification of participles in printed dictionaries is different and not even always consistent within the same dictionary.

For example, the *Oxford Latin Dictionary* (Glare, 1968-1982) has two different lemmas for *amans*, the former being an adjective and the latter a noun. However, *subiectus* is there presented only as an adjective lemma, with its function as a noun being a subcategorization of it. On the contrary, Georges (1913-1918), has only one lemma for *amans* (as an adjective and a noun), but two for *subiectus* (the adjective function being kept separate from the noun one). Both dictionaries, however, do not consider *laborans* a lemma, even though it is also attested with the meaning of "the one who works".

In *LemLat* ('BASE LES' function) *amans* is analyzed as a noun (not as an adjective), while *subiectus* (i.e. "a subordinate") as a verb. On the contrary, *LatMor* keeps the adjective and the noun lemmas separate both for *amans* and *subiectus*. Another interesting example is *florens*, which is analyzed as a verb (i.e. participle) in *LemLat*, but as an adjective and a verb in *LatMor*.

A classification issue similar to that of participles is posed by infinitives. They are normally analyzed as verbs, but one should note that infinitives functioning as nouns are also classified as verbs and therefore their lemmas correspond with that of the corresponding verbs: this clearly challenges the annotation scheme's consistency/uniformity, in that the category 'noun', which is acknowledged, for example, for *studium*, should/could in principle also apply, for example, to *studere* in *studere bonum est*.

Likewise, the gerundive and gerund are problematic because of their nature at the interface between 'verb' and, respectively, 'adjective' and 'noun'. This becomes evident at the syntactic level, in that it is questionable whether they should get adjectival/nominal or verbal syntactic labels.

Rather idiosyncratic is also the category 'pronoun', which does not distinguish pronouns used as adjectives (e.g. *horum amicorum*) from those used as nouns (e.g. *horum*). PoS tagging for relative adverbs such as *ubi*, *quo*, and *qua* can fluctuate between 'adverb' and 'conjunction': in *LemLat ubi* is 'invariable', while *quo* and *qua* are 'pronominal'; in *LatMor ubi* and *quo* are both 'adverb' and 'conjunctions', but *qua* is only 'adverb'.

All the above mentioned uncertainties arising in lemmatization/morphological analysis are ultimately due to lack of (clear) definitions for morphological categories. This is a long-standing problem in linguistics. However, while such classification inconsistencies in printed dictionaries can usually be accommodated by readers because lemmas and the corresponding PoS labels primarily serve the purpose of pointers to word meanings, they impact lemmatizers/morphological analyzers much more severely, in that their function is supposed to be that of providing reliable lemmatization/morphological classification.

Moreover, printed dictionaries can much better cope with spelling issues. It is well known that over centuries Latin showed spelling variants, which lexicographers often try to account for by using internal references. If one looks up *adpono* in, for example, the *Oxford Latin Dictionary* (Glare, 1968-1982), a reference to *app-* is given the reader for all words starting with *adp-*. This system is also used for 'grammatical' references: in the *Thesaurus Linguae Latinae*, for example, *florens* refers to the verb *floreo*.

LemLat has an internal converter for spelling variations: for example, it automatically converts *v* into *u*. This feature is not present in *LatMor*: if a word has a different spelling, it is simply not recognized. Contrary to *LemLat*, *LatMor* adopts the distinction between consonantal *u* (spelled as *v*) and vocalic *u*.

4. *The Latin Dependency Treebank: Towards guidelines for morphological annotation*

Strange as it may sound, there are no guidelines for morphological annotation for the Latin Dependency Treebank (as well as for the other Latin treebanks). Annotation of morphology may at first sight seem less difficult/problematic than that of syntax, and admittedly many studies have been produced for Latin morphology over the centuries, which have reached a consensus on many key points.

Notwithstanding, accounts for Latin morphology vary and, as was shown in Section 3, there exist a number of open questions that need to be addressed before performing corpus annotation. In the light of that, the Latin Dependency Treeebank is currently under revision[26]: in the present section, I outline the current pipeline to annotate lemmas/morphology in it and the challenges faced to foster consistency.

I will discuss the problem of orthography and tokenization in Section 4.1 and Section 4.2, respectively. I will present the morphological analyzer Morpheus in Section 4.3 and the COMBO lemmatizer/PoS tagger/parser in Section 4.4.

[26] *DFG* project 408121292: *https://gepris.dfg.de/gepris/projekt/408121292?context=projekt&task=showDetail&id=408121292&*.

4.1. *Orthography*

An underestimated annotation problem is that of orthography. Latin, as is known, has been written differently over the centuries, and such variations are sometimes recorded in critical editions.

Among the most well-known variations is that between the letters *u* and *v*, and *i* and *j*. Classical Latin had only one letter for both /ʊ uː/ and /w/ and one letter for both /ɪ iː/ and /j/. The two oppositions between the consonant and vowel sounds were introduced in writing later. Other well-known variants – just to mention a few – are the groups *adp-/add-*, *adn/ann-*, or vocalic alternations such as that in *seruos/seruus*.

Such variants pose a challenge for text digitization, lemmatizers/morphological analyzers, and PoS taggers. In the Latin Dependency Treebank, digitized texts preserve the Latin spelling found in critical editions. A normalization layer is, however, planned to be added standoff to each text, so that texts with different spellings can be queried easily and efficiently.

The normalization layer relies on Brambach's rules (McGabe, 1877)[27], which promote use of Latin orthography of the Silver Age. In offering clear guidelines, Brambach's system has already been adopted by many editors[28].

4.2. *Tokenization (and sentence split)*

Tokenization[29] consists in identifying the minimal units for a given analysis/annotation. It is fair to say that the tokenization task for Latin has received much less attention than it deserves. Tokenization represents, *stricto sensu*, the first kind of annotation a text receives.

It is not clear how to exactly define what a token should be in morphosyntactic analysis[30]. In Latin, for example, the negation *non* and the conjunction *et* are recognized as (separate) tokens, but in some treebanks *nec* (i.e. *et non*) represents a single token. Another example are multiword

[27] See *https://archive.org/details/aidstolatinortho00bramrich/page/n6*.

[28] Notably, Brambach sometimes offers more than one option. For more information on how these cases are dealt with, see *https://git.informatik.uni-leipzig.de/celano/latinnlp/blob/master/guidelines/01_orthography.md*.

[29] 'Tokenization' is here used to describe the processes that are sometimes referred to by some scholars as 'tokenization' and 'word segmentation'.

[30] This is of course related to the well-known open question of definition of 'word' (see, for example, Simone, 2008: 150 ff. for a few examples of its heterogeneous nature).

expressions, such as *res publica*: they are commonly treated as two tokens, even though they function syntactically as one-word tokens, such as *Roma* or *mare*.

The problem seems to be even more challenging when it comes to finding a definition of token that applies crosslinguistically: the function of prepositions in a language can be, for example, expressed by cases in another language. One attempt to mitigate some of the irregularities of current tokenization schemes is to account for them at the syntactic level via the use of specific syntactic labels.

Clearly, such a strategy, which seems to be dictated by convenience[31], is questionable on a theoretical level. It is also untested what the impact of such a strategy is on, for example, PoS taggers/syntactic parsers.

For the Latin Dependency Treebank a new rule-based algorithm[32] has been developed to tokenize texts. After whitespace-based tokenization, if a token ending with a punctuation mark is not recognized as an abbreviation (via the use of a word list and a regular expression), the punctuation mark is separated. The same token is then analyzed to see if it matches one of the members in a list containing tokens which need to be split by *ad-hoc* rules: this holds true, for example, for *mecum* or *nequis*.

In order to avoid inconsistencies in the treatment of expressions such as *postquam* and *post quam* or *etiamnunc* and *etiam nunc*, the above mentioned list also contains those tokens that are recognized to have the same function/meaning but can be written as one or two tokens[33]. The split is preferred over the univerbated variant for two reasons: the split variant (i) (typically) antedates the univerbated form and (ii) it is easier to formalize splitting than merging, in that the parts of a split token such as *postquam* could not be adjacent in a clause.

Finally, a graphic word is split into two tokens if it contains the enclitics *que*, *ve/ue*, and *ne*, including *neque*, *nec*, *neve*, *neue*, and *neu*. These latter were sometimes treated as single tokens in the past. They are however split today according to the principle whereby a token needs to be identified if it is required in order to build a correct syntactic tree. For example, if *neque* were not split, one could not correctly build the tree for a sentence

[31] Tokenization asymmetries seem to be related to lack of standoff annotation.

[32] For full documentation, including the actual algorithm, see *https://git.informatik.uni-leipzig.de/celano/latinnlp/blob/master/guidelines/02_tokenization.md*.

[33] This holds true especially for texts of the Golden/Silver Age, which are currently the focus of the Latin Dependency Treebank.

such as the following (I have abbreviated the sentence to focus on the issue at hand):

(1) *Omnes Belgarum copias* [...] *ad se venire vidit neque iam longe abesse* [...] *cognovit.* "He saw all troops of the Belgae [...] were approaching toward him and learned that they [...] were then not far distant."[34]

(Caes. *De Bello Gallico* 2.5.4)

The conjunction *que* coordinates *vidit* and *cognovit*, but the negation *ne-* applies to *abesse* (and not to *cognovit*).

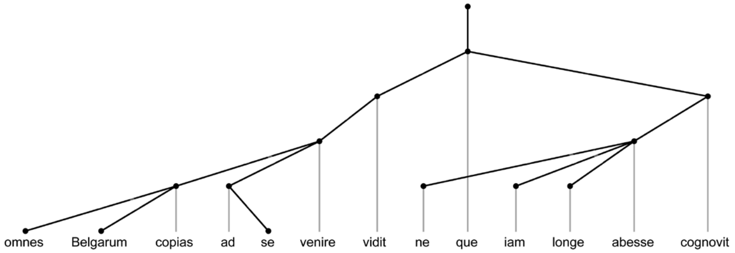

Figure 1. *Parse tree for Caes.* De Bello Gallico *2.5.4.*

Like tokenization, sentence split is currently performed rule-based via a simple algorithm which identifies the major punctuation marks, i.e. full stop, colon, semicolon, question mark, and exclamation point[35].

4.3. *The morphological analyzer Morpheus*

Morpheus (Crane, 1991) is available on the Perseus website[36], via a web API[37], and as a MySQL dump downloadable from the Perseus website[38] (it is also integrated into the annotation tool Arethusa)[39]. These instances serve different purposes. The Perseus website allows easy human interrogation, with morphological analyses been also connected to other resources such as the Lewis and Short (1879) dictionary.

[34] The translation follows *http://data.perseus.org/texts/urn:cts:latinLit:phi0448.phi001.perseus-eng1.*

[35] See *https://git.informatik.uni-leipzig.de/celano/latinnlp/blob/master/guidelines/04_sentence_split.md.*

[36] See *https://www.perseus.tufts.edu/hopper/morph?l=amoris&la=la.*

[37] See *https://www.perseus.tufts.edu/hopper/xmlmorph?lang=lat&lookup=cepissem.*

[38] See *https://www.perseus.tufts.edu/hopper/opensource/download.*

[39] See *https://sosol.perseids.org.*

The web API is designed to automatically parse Latin word forms. The API returns an XML document containing as many <analysis/> elements as the number of possible analyses for a given word form. For example, two analyses for *donis* are given, in that this word form corresponds to the dative plural and ablative plural of the lemma *donum*.

Each <analysis/> element contains a number of child elements describing the morphology of the word form. Among these are the <lemma/> element and <pos/> element (i.e. part of speech), as well as other elements describing morphological features, such as <number/> and <gender/>.

It is possible that the above mentioned versions slightly vary from each other. In the MySQL dump the hib_lemmas table contains 17,573 Latin lemmas. The hib_parses table contains possible morphological forms for each lemma in the hib_lemmas table (466,748). Joining the two tables via the lemma_id field easily allows getting all the word forms and their analyses for a given lemma.

Latin Morpheus is based on the Lewis and Short (1879) dictionary entries. The format of its morphological analyses coincides with the one used in the Latin Dependency Treebank. It is therefore used, for example, to suggest possible morphological analyses during annotation in Arethusa.

The annotation scheme for morphology consists in a 9-character long string, each of them always corresponding to a specific morphological category, which can take one of a finite set of values: if a given category does not apply to a word form, a hyphen is used. The first character specifies the part of speech, and can be any of the following: noun, verb, (participle), adjective, adverb, conjunction, preposition, pronoun, numeral, interjection, and punctuation.

In Morpheus it is possible to see participles treated as a part of speech, but in the Latin Dependency Treebank, 'participle' is a mood. The remaining eight characters represent the following morphological categories[40]: person, number, tense, mood, voice, gender, case, and degree. For example, *rumores* can be annotated as 'n-p---ma-', i.e. noun plural masculine accusative.

As showed previously, there are a few issues concerning morphological annotation and lemmatization that require guidelines. For the next release of the Latin Dependency Treebank, all substantivized nouns are lemma-

[40] See for the sets of all values *https://git.informatik.uni-leipzig.de/celano/latinnlp/blob/master/guidelines/03_morphology.md*.

tized under the corresponding adjective lemmas. This is done in that common practice has always been to generally not identify new lemmas for substantivized adjectives (see, for example, *Romani* as "the Romans", which is typically found under *Romanus*, *a*, *um*). This choice is made also because it seems to be in agreement with the treatment of similar phenomena: for example, substantivized participles are also commonly lemmatized under the corresponding verbs, and pronouns are also not distinguished in their adjectival and nominal function.

Relative adverbs, such as *ubi*, *quo*, or *qua* should be tagged as adverbs, even when they are used without an antecedent and their function resembles that of a conjunction. Indeed, the risk in treating them as conjunctions is that, if any of them happens to play the role of an argument, this can correctly be annotated only if the token is tagged as an adverb.

It is probably because of argument structure that sometimes *ubi* meaning "when" is classified as 'conjunction', while *ubi* meaning "where" tends to be considered as a 'relative adverb': the former is typically an adjunct. Similarly, *quo* meaning "to where" is typically an argument and therefore tends to be analyzed as a relative adverb.

Because of the great variety of lemmatization peculiarities which can affect single tokens and because of the fact that dictionaries are not always consistent in and among themselves as to lemmatization/PoS tagging, the best approach in creating digital resources is probably to make available, and regularly update, open lexica (both for human and computer consumption) compiled following documented criteria.

4.4. *The COMBO lemmatizer/PoS tagger/parser*

Currently, texts in the Latin Dependency Treebank are prepopulated both for lemmatization/morphology and syntax using the output of COMBO. After that, they are typically ingested in the Arethusa annotation tool, so that errors can be manually corrected.

COMBO (Rybak and Wróblewska, 2018)[41] is a state-of-the-art joint neural lemmatizer, PoS tagger, and parser which ranked among the best ones in the *CoNLL* 2018 Shared Task. More precisely, it ranked as the 4th best parser for UPoS, 5th for XPoS, 3rd for morphological features, and 7th for all morphological tags (all rank positions concern annotation of the

[41] See *https://github.com/360er0/COMBO*.

Latin data of the *UD* Latin Dependency Treebank). Differently from other parsers, COMBO has been made available online and is relatively easy to retrain.

As the *CoNLL* 2018 Shared Task is based on data annotated in the Universal Dependency annotation scheme, COMBO had to be retrained in order to output annotations according to the annotation scheme of the Latin Dependency Treebank (v. 2.1). Table 1 shows the accuracies for lemmatization and PoS tagging; the models, a REST API, and accuracies for the syntactic annotation are available online[42].

Field	*Accuracy*
LEMMA	0.83
PoS	0.90
XPoS	0.72
FEAT	0.74

Table 1. *Accuracies for Latin.*

The REST API provided for COMBO allows outputting of morphological and syntactic annotation for Latin according to different annotation schemes: Latin Dependency Treebank, *UD* Latin Dependency Treebank, *UD* Index Thomisticus Treebank, *UD PROIEL* Treebank (the *UD* models are available on the COMBO GitHub repository).

5. *Conclusion and prospects*

The present paper has presented some challenges posed by lemmatization and morphological analysis for Latin, with reference to the ongoing work for the revision of the Latin Dependency Treebank. It has been argued that lemmatizers/morphological analyzers mostly depend on digitized dictionaries, which however contain a number of inconsistencies in lemma identification and PoS tagging.

Indeed, printed dictionaries have been created primarily to provide definitions for Latin words, rather than consistent lemmatization. On the contrary, digital resources, such as treebanks, need to aim to classify Latin

[42] See *https://git.informatik.uni-leipzig.de/celano/COMBO_for_Latin*.

tokens as consistently as possible in order to facilitate automation and query of annotations.

Annotation for the Latin Dependency Treebank currently relies on a rule-based tokenization and sentence-split algorithm, whose output feeds the COMBO lemmatizer, PoS tagger, and parser, used to prepopulate texts. Subsequently, both lemmas and morphological labels are manually corrected. Within the Arethusa annotation tool, the morphological analyzer Morpheus can sometimes help selection of correct alternative labels.

A major goal of the current revision of the Latin Dependency Treebank is to also document annotation choices for lemmatization/morphology via examples/rules to foster consistency: this is work in progress[43].

Acknowledgments

I gratefully acknowledge the support of the *DFG* (Deutsche Forschungsgemeinschaft), which has funded the present research (Project Number: 408121292).

References

Celano, G.G.A. (2019a), *Standoff annotation for the Ancient Greek and Latin Dependency Treebank*, in *DATeCH2019: Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage, Brussels, Belgium*, Association for Computing Machinery, New York, pp. 149-153.

Celano, G.G.A. (2019b), *The Dependency Treebank for Ancient Greek and Latin*, in Berti, M. (2019, ed.), *Digital Classical Philology: Ancient Greek and Latin in the Digital Revolution*, De Gruyter, Berlin, pp. 279-298.

Crane, G. (1991), *Generating and parsing Ancient Greek*, in «Literary and Linguistic Computing», 6, 4, pp. 243-245.

Georges, K.E. and Georges, H. (1913-1918), *Ausführliches Lateinisch-Deutsches Handwörterbuch*, Hahn, Hannover.

Glare, P.G.W. (1968-1982), *Oxford Latin Dictionary*, Oxford University Press, Oxford.

[43] See *https://git.informatik.uni-leipzig.de/celano/latinnlp*.

Gleim, R., Eger, S., Mehler, A., Uslu, T., Hemati, W., Lücking, A., Henlein, A., Kahlsdorf, S. and Hoenen, A. (2019), *Practitioner's view: A comparison and a survey of lemmatization and morphological tagging in German and Latin*, in «Journal of Language Modelling», 7, 1, pp. 1-52.

Gradenwitz, O. (1904), *Laterculi Vocum Latinarum*, Hirzel, Leipzig.

Haspelmath, M. (2012), *How to compare major word-classes across the world's languages*, in «UCLA Working Papers in Linguistics», 17, pp. 109-130.

Lewis, C.T. and Short, C. (1879), *A Latin Dictionary*, Oxford University Press, Oxford.

Lewis, C.T. and Short, C. (1907), *A New Latin Dictionary*, American Book Co., New York.

McGabe, W.G. (1877), *Aids to Latin Orthography by Wilhelm Brambach*, Harper and Brothers, New York.

Passarotti, M., Budassi, M., Litta, E. and Ruffolo, P. (2017), *The* Lemlat *3.0 package for morphological analysis of Latin*, in Bouma, G. and Adesam, Y. (2017, eds.), *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language, Gothenburg, Sweden*, Linköping University Electronic Press, Linköping, pp. 24-31.

Rybak, P. and Wróblewska, A. (2018), *Semi-supervised neural system for tagging, parsing and lemmatization*, in Zeman, D. and Hajič, J. (2018, eds.), *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Brussels, Belgium, October 31 - November 1*, Association for Computational Linguistics, pp. 45-54.

Sasse, H.-J. (2001), *Scales between nouniness and verbiness*, in Haspelmath, M., König, E., Oesterreicher, W. and Raible, W. (2001, eds.), *Language Typology and Language Universals*, De Gruyter, Berlin / New York, pp. 495-509.

Simone, R. (2008), *Fondamenti di linguistica*, Laterza, Roma / Bari.

Springmann, U., Schmid, H. and Najock, D. (2016), LatMor*: A Latin finite-state morphology encoding vowel quantity*, in Celano, G.G.A. and Crane, G. (2016, eds.), *Treebanking and Ancient Languages: Current and Prospective Research*, in «Open Linguistics», 2, 1, pp. 386-392.

Sprugnoli, R., Passarotti, M. and Moretti, G. (2019), Vir *is to* Moderatus *as* Mulier *is to* Intemperans *- Lemma embeddings for Latin*, in Bernardi, R., Navigli, R. and Semeraro, G. (2019, eds.), *Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15*.

Giuseppe G.A. Celano
Abteilung Automatische Sprachverarbeitung
Institut für Informatik
Universität Leipzig
Augustusplatz 10
04109 Leipzig (Germany)
celano@informatik.uni-leipzig.de