

Research Article

Open Access

Uwe Springmann*, Helmut Schmid, and Dietmar Najock

LatMor: A Latin Finite-State Morphology Encoding Vowel Quantity

DOI 10.1515/opli-2016-0019

Received Feb 29, 2016; accepted May 18, 2016

Abstract: We present the first large-coverage finite-state open-source morphology for Latin (called LatMor) which parses as well as generates vowel quantity information. LatMor is based on the Berlin Latin Lexicon comprising about 70,000 lemmata of classical Latin compiled by the group of Dietmar Najock in their work on concordances of Latin authors (see Rapsch and Najock, 1991) which was recently updated by us. Compared to the well-known Morpheus system of Crane (1991, 1998), which is written in the C programming language, based on 50,000 lemmata of Lewis and Short (1907), not well documented and therefore not easily extended, our new morphology has a larger vocabulary, is about 60 to 1200 times faster and is built in the form of finite-state transducers which can analyze as well as generate wordforms and represent the state-of-the-art implementation method in computational morphology. The current coverage of LatMor is evaluated against Morpheus and other existing systems (some of which are not openly accessible), and is shown to rank first among all systems together with the Pisa LEMLAT morphology (not yet openly accessible). Recall has been analyzed taking the Latin Dependency Treebank¹ as gold data and the remaining defect classes have been identified. LatMor is available under an open source licence to allow its wide usage by all interested parties.

Keywords: morphology; finite state methods; Latin; historical linguistics

1 Introduction

Morphological analysis is an important step for automatic processing of natural languages. Many tools such as part-of-speech (POS) taggers and parsers require or profit from information about possible part-of-speech tags of wordforms produced by a morphological analyzer.

For many widely spoken modern languages, morphological analyzers (Beesley, 1996; Çöltekin, 2010; Schmid et al., 2004, to name just a few) are readily available, which have usually been implemented with finite state transducers. For Latin, the situation is different: The two best-known and publicly available analyzers, namely William Whitaker's Words tool² and Morpheus from the Perseus Digital Library project³ (an analyzer for both Latin and Ancient Greek: Crane, 1991, 1998), have been written in ADA and C, respectively. Only in the last few years there have been some implementations within a transducer framework such as

Article note: This paper belongs to the special issue on Treebanking and Ancient Languages, ed. by Giuseppe G.A. Celano and Gregory Crane.

***Corresponding Author: Uwe Springmann:** Centrum für Informations- und Sprachverarbeitung, Ludwig-Maximilians-Universität München; Institut für deutsche Sprache und Linguistik, Humboldt-Universität zu Berlin; e-mail: firstname@last-name.net


Helmut Schmid: Centrum für Informations- und Sprachverarbeitung, Ludwig-Maximilians-Universität München, email: last-name@cis.uni-muenchen.de

Dietmar Najock: Institut für Griechische und Lateinische Philologie, Freie Universität Berlin

1 https://github.com/PerseusDL/treebank_data

2 <http://archives.nd.edu/whitaker/words.htm>

3 <http://www.perseus.tufts.edu/hopper/morph?lang=la>

 © 2016 U. Springmann et al., published by De Gruyter Open.

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License.

Bereitgestellt von | provisional account
Unangemeldet
Heruntergeladen am | 06.12.18 16:38

Parsley and PROIEL (see Sect. 2), and they are up to 1,000 times faster than previous methods, being able to analyse several 100,000 wordforms per second.

In this paper, we present the first large-coverage finite-state open-source morphology for Latin (called LatMor) which parses as well as generates vowel quantity information. This transducer runs in both analysis and generation mode and can analyze both normal and vowel quantity marked-up text. It can also analyze normal text (without vowel marks) and output all possible matching analyses in a marked-up form, which provides necessary information (up to the needed disambiguation) for a Text-To-Speech (TTS) system⁴ or for the analysis of metrical patterns in poetry and prose.

The remainder of this paper is organized as follows: Sect. 2 describes previous work on Latin morphology, Sect. 3 gives information on the Berlin Lexicon, in Sect. 4 we give details on our finite state transducers, and in Sect. 5. we evaluate our morphology with respect to coverage and recall. The paper concludes with future work and a summary.

2 Previous work

Among Latin morphological analyzers written for scientific use (i.e., not just programs with limited capabilities and typically a small lexicon covering some basic vocabulary addressing the needs of learners at schools or universities) the following additional approaches (apart from Whitaker's Words and Morpheus) are known to us:

LEMLAT⁵ (Passarotti, 2004) was written at ILC-CNR in Pisa (this analyzer is not publicly available due to copyright issues). The LatLem system of Najock and Morgenroth is based upon a lemma lexicon (see Sect. 3) that was expanded into full forms together with their morphological tags by a Pascal program written by Hermann Morgenroth (see Rapsch and Najock, 1991, p. IX-XII). The resulting file consisting of about 2 million full forms was used for the lemmatization of classical Latin texts by a table-lookup method. The PROIEL Latin morphology⁶ was compiled for the "Pragmatic Resources in Old Indo-European Languages" (Haug and Jøhndal, 2008). Morpheus was first developed for Ancient Greek in 1985 and later extended to support Latin in 1996. Parsley⁷ is a 2013 reimplementation of Morpheus (Latin analyzer) by Harry Schmidt reusing its stem and endings tables and applying the SFST toolbox of Helmut Schmid (Schmid, 2006).

While Words, Morpheus, and LEMLAT have been implemented using traditional general purpose programming languages, the newer systems PROIEL, Parsley, and our LatMor system all use the SFST toolbox. Morpheus, Parsley, and LatMor are all capable of analyzing wordforms into their morphological representations including vowel quantity, but LatMor is the only one which both analyzes as well as generates wordforms with vowel quantities.

3 The Berlin Latin Lexicon

This section describes the Berlin Latin Lexicon, a lemma lexicon which was compiled in the 1980s in the group of Dietmar Najock at Freie Universität Berlin, with substantial contributions of Peter Rosumek, the main editor of the concordance to Pliny the Elder (Rosumek and Najock, 1996). This lexicon contains about 70.000 lemmata and was mainly built from the entries of Georges' Handwörterbuch (Georges, 1913) with ad-

⁴ In Latin, the stressed syllable depends on syllable length which for open syllables in turn depends on vowel length: Open syllables with a short vowel are short, all other syllables are long. The penultimate syllable is stressed if long, otherwise the syllable before that if present.

⁵ <http://www.ilc.cnr.it/lemlat>

⁶ <https://github.com/mlj/proiel-webapp/tree/master/lib/morphology>

⁷ <http://parsley.goldibex.com/>, <https://github.com/goldibex/parsley-core>

ditional proper names from Lewis and Short (1907). In order to cover vowel quantity (distinction between long and short vowels), the entries were checked against the lexicon by Menge et al. (1983). An expanded list of wordforms together with their morphological tags was generated and used for lemmatizing classical Latin texts in order to build concordances. When new words showed up in the course of this work, they were incorporated into the lexicon.

Morphological information printed in lexica is usually not in a form that can unambiguously be parsed by a machine. Georges prints the following third declension nouns as *pater*, *tris m.*; *mare*, *is n.*; and *ovis*, *is f.* The computer needs to know that the genitives are *patris*, *maris*, *ovis* and not **patertris*, **mareis*, **ovisis*. Therefore, a “/” has been inserted in the lemmata at exactly the place where the ending needs to be put: *pa/ter*, *mar/e*, *ov/is*. A typical entry therefore looks like *su serv/us, i: m*, giving the part-of-speech, lemma with “stem” division, genitive (the colon indicates a long vowel) and gender. We are currently working to devise rules which will automatically find these stems directly from the information in printed lexica which may become available electronically as transcriptions or via OCR, making it easier to expand the lexical basis.

Our update to this lexicon consisted in the emendation of some known erroneous or incomplete entries. A systematic evaluation of possible transcription errors for vowel quantities is currently under way. We will also incorporate information on vowel quantities gained since the time of publication of Georges (1913) and Menge et al. (1983) as compiled by Allen (1989).

4 Finite-State Analyzer

Our finite-state analyzer uses the Berlin Latin Lexicon. Because its format cannot be processed directly by the finite state program, it is first converted by means of a Perl script. The entry *su serv/us, i: m*, for instance, is replaced by *servus<N><base><NMasc-o>*. This LatMor lexicon entry explicitly encodes an inflection class (here *NMasc-o*) which was implicit in the original entry. The implementation of the conversion rules was a time-consuming manual process, but once it has been done, new lexicon resources written in the standard format can easily be integrated.

LatMor was implemented with the finite-state transducer toolkit SFST (Schmid, 2006). It creates the full set of inflectional endings for each inflectional class and attaches them to the respective word stems which are obtained by removing the standard inflectional ending (e.g. *-us* in the above example). A set of morphophonological rules is applied to generate the correct surface forms. They insert, for instance, the letter “u” in *audiunt* and shorten the long “a” vowel of *laudāre*⁸ in the form *laudant*. The development of the inflection module was based on the inflection tables provided by Rubenbauer et al. (1995). Greek nouns and adjectives present in the lexicon are inflected according to the paradigms given in Leumann et al. (1977).

Words with highly irregular inflection (e.g. pronouns such as *tibi*) are directly mapped to their analysis (*tu<PRO><pers><2><sg><dat>*) instead of employing the inflection mechanism described before. Otherwise, we would have to define inflection classes which comprise a single member. Exceptions can be specified explicitly. The form *bene*, e.g., replaces the regularly formed (but incorrect) adverb **bone* for the adjective *bonus*. This was implemented by first deleting *bone* from the transducer and then adding the irregular form *bene*. Similarly, it is possible to specify that a certain wordform generated by the inflectional paradigm of a word does not exist. In this case, no irregular form is added.

⁸ The transducer uses macrons to represent long vowels.

Table 1: Coverage of different morphological analyzers on three Latin texts. A wordform counts as analyzed if either the wordform itself or its lowercased form receives an analysis. The best results are underlined.

	Caesar		Nepos		Godfrey	
all	type	token	type	token	type	token
PROIEL	70.0	51.6	69.4	47.9	63.1	50.6
Parsley	89.5	95.2	90.0	94.3	86.7	91.7
Words	90.5	96.6	88.1	93.3	93.0	95.4
Morpheus	92.5	93.8	89.0	92.7	87.6	92.7
LEMLAT	<u>98.2</u>	99.0	<u>98.1</u>	99.1	91.0	94.9
LatMor	97.5	<u>99.1</u>	<u>98.1</u>	<u>99.2</u>	<u>96.4</u>	<u>97.5</u>

5 Evaluation

The compiled morphology currently analyzes 2,206,464 different wordforms⁹ with 2.5 analyses per wordform on average. If we ignore ambiguity due to syncretism (e.g. *servīs* can be either dative or ablative plural), the average ambiguity reduces to 1.05 analyses per wordform.¹⁰ The disambiguation of the set of possible morphological analyses to the single analysis that is correct within a given sentence context is not a task for a morphology; however, a morphological disambiguator such as the MarMoT tagger (Müller et al., 2013) could be used for such a context-dependent disambiguation.

In a small-scale evaluation, we measured the coverage of our morphological analyzer on three randomly selected Latin texts, two from classical Latin (Gaius Julius Caesar: *De Bello Gallico* and Cornelius Nepos: *Liber de excellentibus ducibus exterarum gentium*) and one from medieval Latin (Godfrey of Winchester: *Epigrammata*, 11th century), all taken from the Latin Library.¹¹

For these texts we also measured the number of possible wordforms differing only in vowel lengths (such as *sequeris*, you follow, and *sequēris*, you will follow). If one wants to prepare texts with vowel quantity markings (for didactic purposes or as input for a TTS system), a list of possible wordforms to choose from (in case of ambiguity) would help the annotator and avoid manual errors. The average number of different vowel-length realizations per token is 1.15 for *De Bello Gallico*, 1.17 for *Nepos* and 1.22 for *Godfrey*.

We compared our morphological analyzer LatMor with these other analyzers: PROIEL, Parsley, Words, Morpheus,¹² and LEMLAT. The texts contain wordforms which have been capitalized (e.g. because they appeared at sentence start). LatMor is able to analyze these wordforms, but not all of the other analyzers. In order to treat them fairly, we consider a wordform as covered by an analyzer if either the original form or a lowercased version of it receives an analysis.

The results of the experiment are shown in Table 1. The table contains the results on all words with the above convention that a word counts as analyzed if either the original form or its lowercased form gets at least one analysis. Token as well as type coverage are given for each text.¹³

LatMor processes over 100,000 wordforms per second on a laptop computer with an Intel Core i5-3320M CPU @ 2.60GHz. An analysis of the 11,420 tokens of Caesars Gallic Wars takes 0.1 sec compared to 6 sec of a local Morpheus installation or 20 min using Morpheus as a web service with Perseids.

⁹ We do not count numerals such as XVIII, wordforms with clitics and capitalized versions of lowercase wordforms, here.

¹⁰ Here we count how many different lemmata a wordform has on average. Lemmata are considered different if either the spelling or the part-of-speech are not identical.

¹¹ <http://www.thelatinlibrary.com/>

¹² Testing was done against the Perseids morphology web service which provided better results than a local installation of Morpheus: <https://sites.tufts.edu/perseusupdates/2012/11/01/morphology-service-beta/>

¹³ The type coverage is computed on the text vocabulary. The token coverage considers all word occurrences of the text and therefore gives higher weight to frequent wordforms.

Analyzers with a comprehensive lexicon (Morpheus, LEMLAT, and LatMor) show the best results. Token coverage is typically higher than type coverage because rare words not in the lexicon have higher weight in the type-based evaluation. The only exception to this is PROIEL where the lexicon misses many function words (*in*, *et*, *ad*, etc). Different Morpheus implementations show different coverage (local installations, Parsley, Perseids morphology service) due to different stem tables.¹⁴ We used the Perseids morphology web service for its consistently high results on lowercase words, although apparently it completely misses proper names. Overall, LatMor achieves the best coverage, being able to analyse both enclitics (e.g. appended *-que*, *-ne*, *-ve*) and Roman numerals (see the difference to other systems on Godfrey). LEMLAT is also very good except for Godfrey and the best performing system for the types of Caesar. If Roman numerals are taken out of account, LEMLAT is in fact the best performing system of all. However, this system is encumbered by property rights as it got patented early in its development and cannot yet be openly released.

Whereas the above evaluation just asks how many of the tokens of a text can be analyzed, we also evaluated recall: How often does our list of analyses contain the correct analysis of a token in its sentence context? As a gold standard to compare against, we used a subset of the Perseus Latin Dependency Treebank (LDT) 2.1¹⁵ consisting of the revised treebank annotations of Caesar (*De Bello Gallico*), Cicero (*In Catilinam*), Propertius (*Elegiae*), and Vergilius (*Aeneis*) as well as the newly added texts of Phaedrus (*Fabulae*), Suetonius (*Life of Augustus*), and Tacitus (*Historiae*).

For the evaluation, we removed sentences with untagged words from the Perseus data, unless the untagged tokens were punctuation symbols. Features in the LatMor output were mapped to Perseus tags in the following way: Perseus tags for exclamations and interjections were merged because they are not distinguished in LatMor. We ignored punctuation symbols, parentheses, and clitics such as *-que* and *-ne* which are separated from the preceding words in the Perseus tokenization. LatMor is able to analyze clitic word-forms when written as a single word, but we didn't bother to reconstruct them from the Perseus tokenization. In the comparison, we allowed LatMor pronouns to match with Perseus adjectives, because e.g. possessive pronouns are occasionally mistagged as adjectives in Perseus and many indefinite pronouns such as *nullus* are consistently tagged as adjectives. When matching LatMor pronouns with Perseus pronouns or adjectives, we allowed unspecified feature values to match with other feature values: LatMor and Perseus do not always agree on the set of features that should be specified for pronouns. Perseus, for instance, never assigns a person feature to pronouns, but specifies the gender of the personal pronoun *tu*. We also did not count as a mismatch if the voice feature of a participle or supine was undefined in Perseus. In all other cases, a feature mismatch between a LatMor and Perseus analysis results in an analysis mismatch. If none of the LatMor analyses matches the Perseus analysis of a word, we count it as an error candidate in the evaluation.

With these matching rules, LatMor matched 13,162 cases or 95.7% of 13,857 unique analyses contained in the treebank texts. The remaining 695 discrepant entries were analysed individually by hand and can be broken down into the following classes:

1. Different tags (284): This largest group of discrepancies does not necessarily consist of errors but rather reflects different tagging conventions. E.g., adjectives arising from perfect participles always have a verb as lemma in LDT, whereas LatMor lists the adjective (e.g. *divertor* instead of *diversus*). Also, LDT often labels deponential forms as passive, but LatMor as deponential.
2. Missing lemmata (256): Here the largest contribution is from proper names ending in *-ius* such as *Cassius* or *Pompeius* which the Berlin Lexicon only lists as adjectives, a choice that can be traced back to the respective lemmata in Georges (1913). Abbreviations of first names are also not analyzed as names.
3. Spelling differences (56): This group mainly consists of differences arising from non-assimilated composita and *u/v* variations, such as *conlocavit* for *collocavit*.
4. Special forms (52): missing special morphological forms such as *volt* for *vult*, *sequentum* als genitive plural for *sequentium*, *duxti* for *duxisti*, *oreris* for *oriris*.

¹⁴ Crane, priv. comm.

¹⁵ https://github.com/PerseusDL/treebank_data/tree/master/v2.1/Latin

5. LDT errors (47): This smallest group actually consists of errors in our assumed gold data (wrong lemmata, tenses or other incorrect labels).

6 Future Work

The failure groups presented in the previous section suggest directions for further improving the morphology:

- Gaps in the lexicon need to be identified and fixed (e.g., by including the proper names of the 2-volume *Onomasticon* of Forcellini et al., 1940).
- Derivation rules should be implemented to analyze e.g. prefix verbs and to account for phonological assimilation processes giving rise to spelling variation.
- Irregular forms need to be identified and taken into account.

Work along these lines is currently under way. We wrote our new morphology also with the idea of extending it to enable the analysis of the extensive wordformation happening in the Neo-Latin era (e.g. compounds ending in *-logia* such as *botanologia*, *deuterologia*; Ramminger, 2014) so that it can be used to analyse the increasing electronic corpus of the vast amount of Neo-Latin literature becoming available from transcription and OCR efforts. The inclusion of derivation and compounding as morphological processes has been demonstrated by Schmid et al. (2004) in the case of the German morphology SMOR and will be applied to LatMor as well.

7 Summary

We presented the first large-coverage finite-state open-source morphology for Latin which encodes vowel quantity information. The morphology is based on a large Latin dictionary which was compiled from a variety of resources. The morphological analyzer achieves large coverage over a random selection of three texts from classical as well as medieval Latin. It is publicly available under a CC-BY-NC-SA licence at <http://cistern.cis.lmu.de>.

Acknowledgement: We thank Marco Passarotti and Greg Crane for providing us with reference data to test against.

Funding: This work was partially funded by Deutsche Forschungsgemeinschaft (DFG) under grant no. LU 856/7-1 and SCHU-1026/7-1.

References

- William Sidney Allen. *Vox Latina (2nd edn., corrected reprint)*. Cambridge University Press, 1989.
- Kenneth R. Beesley. Arabic finite-state morphological analysis and generation. In *Proceedings of the 16th Conference on Computational Linguistics (COLING)*, volume 1, pages 89–94, 1996.
- Çağrı Çöltekin. A freely available morphological analyzer for Turkish. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, pages 820–827, 2010. URL <http://www.lrec-conf.org/proceedings/lrec2010/summaries/109.html>.
- Gregory Crane. Generating and parsing classical Greek. *Literary and Linguistic Computing*, 6(4):243–245, 1991.
- Gregory Crane. New technologies for reading: The lexicon and the digital library. *The Classical World*, pages 471–501, 1998.
- Egidio Forcellini, Giuseppe Furlanetto, Francesco Corradini, and Joseph Perin. *Lexicon totius Latinitatis, t. V-VI: Onomasticon. Typis Seminarii, Patavii*, 1940.
- Karl Ernst Georges. *Ausführliches lateinisch-deutsches Handwörterbuch*. Hahnsche Buchhandlung, Hannover und Leipzig, 1913.
- Dag Trygve Truslew Haug and Marius Jøhndal. Creating a parallel treebank of the old Indo-European Bible translations. In *Proceedings of the Language Technology for Cultural Heritage Data Workshop (LaTeCH 2008), Marrakech, Morocco, 1st June 2008*,

- pages 27–34, 2008.
- Manu Leumann, Johann Baptist Hofmann, and Anton Szantyr. *Lateinische Grammatik: Lateinische Laut- und Formenlehre*. CH Beck, 1977.
- Charlton T Lewis and Charles Short. *A New Latin Dictionary. Founded on the Translation of Freund's Latin German Lexicon Edited By E. A. Andrews, LL D*. Clarendon Press, 1907.
- Hermann Menge, Otto Güthling, and Erich Pertsch. *Langenscheidts großes Schulwörterbuch lateinisch-deutsch*. Langenscheidt, 1983.
- Thomas Müller, Helmut Schmid, and Hinrich Schütze. Efficient higher-order CRFs for morphological tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, 2013.
- Marco Carlo Passarotti. Development and perspectives of the Latin morphological analyser LEMLAT. *Linguistica computazionale*, 20(A):397–414, 2004.
- Johann Ramminger. Neo-Latin: Character and Development. In Philip Ford, Jan Bloemendal, and Charles Fantazzi, editors, *Brill's Encyclopaedia of the Neo-Latin World*, pages 21–36. Brill, 2014.
- Jürgen Rapsch and Dietmar Najock. *Concordantia in Corpus Sallustianum*, 2 vols. Olms, Hildesheim, Zürich, New York, 1991.
- Peter Rosumek and Dietmar Najock. *Concordantia in C. Plinii Secundi Naturalem Historiam*. Alpha - Omega / A. Olms-Weidmann, Hildesheim u.a, 1996. ISBN 9783487100166. URL <http://isbnplus.org/9783487100166>.
- Hans Rubenbauer, Johann Baptist Hofmann, and Rolf Heine. *Lateinische Grammatik, 12. Auflage*. Buchner, Lindauer, Oldenbourg, Bamberg-München, 1995.
- Helmut Schmid. A programming language for finite state transducers. In Anssi Yli-Jyrä, editor, *Finite-State Methods and Natural Language Processing: 5th International Workshop (FSMNLP 2005)*, volume 4002 of *Lecture Notes in Artificial Intelligence*, pages 308–309. Springer, Heidelberg, Germany, 2006.
- Helmut Schmid, Arne Fitschen, and Ulrich Heid. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, volume 4, pages 1263–1266, Lisbon, Portugal, 2004.