UPPSALA
UNIVERSITET

Automatic annotation of Latin vowel length

Johan Winge

Uppsala University
Department of Linguistics and Philology
Språkteknologiprogrammet
(Language Technology Programme)
Bachelor's Thesis in Language Technology

June 4, 2015

Supervisor:
Joakim Nivre

Abstract

This thesis describes a system for annotating Latin texts with vowel lengths using a PoS tagger, a lemmatizer, and the Latin morphological analyzer Morpheus. Three taggers, HunPos, RFTagger, and MATE tools, were optimized and evaluated for this task, and trained on two different corpora, the Latin Dependency Treebank (LDT) and PROIEL. The best tagging results were achieved by RFTagger with 86.73% accuracy on the full morphological PoS tags in LDT, and by MATE tools with 90.42% accuracy when predicting the corresponding combined plain PoS and morphological descriptors of PROIEL. With RFTagger trained on either of the two corpora, the complete system was tested on four different texts and was found to agree with the human annotator on about 98% of the vowel lengths.

Summārium

Litterās singulās dīligenter perscrūtārī omnēsque vōcālēs longās līneolīs ōrnāre molestissimum labōrem esse, ac paene ad īnsāniam adigentem, nēmō est quīn sciat, sī id vel semel temptāverit. Quid ergō melius quam istud onus computātrīs trādere, utpote quae māchinae paene īnfīnītā patientiā industriāque praeditae sint? Ut cognōscātur quae vōcālēs longae sint in verbō quaeque brevēs, opus est dēcernere prīmum quō modō, deinde ex quō vocābulō sīve "lēmmate" dēclīnātum sit; quibus enim rēbus cognitīs expedītē in indice verbōrum apta fōrma inveniātur līneolīs īnstrūcta. Hāc in commentātiōne systēma dēscrībitur, quod ex contextū verbōrum jam annotātōrum computātiōne (nec tamen vērā ratiōne animī, quae nūlla adhūc est māchinīs) discit distinguere inter flexiōnēs similēs, atque ad vērum lēmma verbum redūcere, et sīc longitūdinēs vōcālium dīvīnāre. Perīculō factō comprobātum est hoc systēma ab jūdiciō hominis dē longitūdine fermē quīnquāgēsimae quaeque vōcālis dissentīre.

Contents

Acknowledgements **4**

1 Introduction **5**

2 Background **6**
2.1 Latin vowel lengths 6
2.2 Latin resources 6
2.2.1 The Morpheus morphological analyser 6
2.2.2 Annotated corpora 7
2.3 Tagging techniques and programs 10
2.3.1 HMM tagging with HunPos 10
2.3.2 RFTagger 12
2.3.3 MATE tools 12
2.4 Previous work 13
2.4.1 The Māccer macronizer 13
2.4.2 Latin PoS tagging 14

3 Methodology **16**
3.1 System overview 16
3.2 Tokenization 18
3.3 PoS tagging 18
3.4 Lemmatization 20
3.5 System evaluation 21

4 Results **22**
4.1 Tagger optimization 22
4.1.1 HunPos 22
4.1.2 RFTagger 24
4.2 Tagger comparison 25
4.3 Lemmatization 26
4.3.1 Optimization of PurePos 26
4.3.2 Lemmatizer comparison 26
4.4 System evaluation 27

5 Discussion and conclusion **29**

Bibliography **31**

A Tag sets **33**

Acknowledgements

First and foremost I would like to thank my supervisor, Joakim Nivre, who on several occasions pushed me in the right direction and provided much appreciated criticism. I am also indebted to Marco Passarotti, György Orosz, and Bernd Bohnet, who generously gave assistance whenever I reached out to them. Thanks also to Daniel, for his inspiring enthusiasm and Latin proofreading, and to Gunilla, for her constant patience, support, and encouragement. Finally, I want to extend my gratitude to Gregory Crane for creating the morphological analyzer Morpheus, without which this thesis would not have been possible.

1 Introduction

Classical Latin (a label conventionally given to the language from roughly 80 BC to 180 AD) had a duplicate vowel system, whereby each of the six short vowel phonemes /a, e, i, o, u, y/ also had a corresponding long variant. In the Latin alphabet, however, the vowel letters had to represent both the short and the long sounds. While the Romans during some periods would occasionally make the lengths explicit in writing, either by doubling the letters or by marking long vowels with a diacritic called the *apex* (similar to an acute accent), in the usual standardized orthography which is now used in regular editions of Latin texts, no such distinctions are made.

The main exceptions are dictionaries and textbooks, where long vowels are usually marked with the *macron* diacritic, a small straight horizontal bar: *ā*, *ē*, *ī* etc.; short vowels are then either left unmarked or, in order to make the short length explicit, marked with a small semicircle, the *breve*: *ă*, *ĕ*, *ĭ* etc. Apart from the general interest of historical authenticity, knowing the vowel quantities gives the ability to correctly place the stress on the correct syllables, and to better appreciate classical poetry.

Not having vowel lengths marked results in a large number of false homonyms, i.e. word forms which are homographs but not actually homophones when pronounced using a restored classical pronunciation. This can occur when two different inflections of the same lemma are spelled the same, such as *mensa* ('table') which can be either in the nominative (*mēnsă*) or in the ablative case (*mēnsā*), or when the spellings of two different lemmas happen to coincide, such as *lĕvis* 'light' (adj.) and *lēvis* 'smooth', or when both the lemmas and the inflections are different, e. g. *rĕgī* 'to be ruled' and *rēgī* 'king' (dat.). To correctly mark all the vowel lengths in a text is thus a disambiguation task with respect to both lemmatization and morphological classification.

Since marking vowel lengths by hand is a repetitive and error prone process, it would be beneficial, both as a paedagogical tool in itself and when editing textbooks, to be able to perform an automatic annotation of the vowel lengths in a text. The purpose of this thesis is to explore how the available resources in the form of common NLP tools and annotated Latin corpora can be adapted and combined to form an automatic vowel length annotation system, and in particular to investigate how automatic part-of-speech (PoS) tagging can be used to improve the result.

With regard to terminology, it should be noted that the term "PoS tagging" throughout this work is generally used in a less restricted sense, encompassing tagging with the plain part of speech (noun, verb etc.) as well as with more detailed morphological attributes (case, number, tense etc.). A text with marked vowel lengths is referred to as being "macronized".

2 Background

2.1 Latin vowel lengths

The classical Latin language had few phonotactic restrictions on the distribution of short and long vowels. In initial, medial, and final syllables alike, accented or not, both long and short vowels may be found. As vowel lengths are not indicated in the normal Latin orthography, our knowledge of which vowels are long and which are short depend on other sources, first and foremost classical Latin poetry, which was bound by quantitative meters, i.e., a regular alternation of heavy and light syllables.

Light syllables are those which end in a short vowel, (C)V; all other syllable structures count as heavy: (C)VV, (C)VVC, or (C)VC. Thus, if a metrically heavy syllable does not end with a consonant, we can deduce that the vowel is long; otherwise, the quantity is "hidden", meaning that the vowel can be either long or short. In that case, evidence has to be gathered from etymology, development in the Romance languages, borrowings into other languages, and so on (Allen 1989). When the length is uncertain, it is not uncommon to see the vowel marked differently in different textbooks and lexica.

2.2 Latin resources

2.2.1 The Morpheus morphological analyser

The Morpheus morphological analyser was developed as part of the Perseus Digital Library Project,[1] a large collection of resources pertaining to the history, literature and culture of the Greco-Roman world (Crane 1991). Its database is derived from the Latin–English dictionary by Lewis and Short (1879).

Given a list of word forms as input, it will, for each individual word, give all possible morphological analyses that it manages to find, based on its lexicon and built-in inflectional grammar. Each analysis consists of a series of keywords that describe the word inflection, together with the corresponding lemma and the word form with vowel quantities marked up. (Macrons are then represented with underscores, while breves are written as freestanding circumflexes.) For example, given the word form *regi*, Morpheus will give three possible analyses:

```
<NL>N re_gi_,regius  masc/neut gen sg          ius_ia_ium</NL>
<NL>V re^gi_,rego  pres inf pass               conj3</NL>
<NL>N re_gi_,rex  masc dat sgx_gis</NL>
```

[1] `http://nlp.perseus.tufts.edu/`

The human readable keywords can of course quite easily be analysed and converted to a compact morphological tag, such as those used in the Latin Dependency Treebank (see below), or to any similar format.

The Morpheus program can thus serve two purposes: either as a mapper from word forms to possible morphological tags (which can be used to create a lexicon for use during tagging to restrict the set of possible tags for each token), or as a mapper from word form and tag pairs to macronized word forms, to be used during post-processing of the automatically tagged text.

2.2.2 Annotated corpora

Extensive Latin corpora with manually annotated morphosyntactic information exist primarily in the form of dependency treebanks, of which three are freely available, all under a Creative Commons Attribution-NonCommercial-ShareAlike license (McGillivray 2013, p. 8):

- The Latin Dependency Treebank (LDT) (Bamman and Crane 2011).
- The Latin part of the Pragmatic Resources in Old Indo-European Languages (PROIEL) treebank (Haug and Jøhndal 2008).
- The Index Thomisticus Treebank (IT-TB) (Passarotti 2010).

A fourth corpus, the LASLA (Laboratoire d'Analyse Statistique des Langues Anciennes) database, is bigger than any of these by a large margin, consisting of around 2 000 000 annotated Latin tokens, encompassing text from a large number of classical authors, and some neo-Latin texts as well. Unlike the other three it is not a treebank, and, while the database can be queried online, it is not available for public use in its entirety (Piotrowski 2012, p. 114).

The Latin Dependency Treebank

Just as Morpheus, the Latin Dependency Treebank (LDT) was developed as part of the Perseus Project. It is a collection of a number of different Latin texts by some of the most well known classical authors:[2]

Author	Work	Sentences	Tokens
Caesar	*De Bello Gallico* (selections)	71	1 488
Cicero	*In Catilinam* 1.1–2.11	327	6 229
Sallust	*Catilina*	701	12 311
Petronius	*Cena Trimalchionis*	1 114	12 474
Jerome	*Vulgate: Apocalypse*	405	8 382
Propertius	*Elegies:* Book 1	361	4 857
Ovid	*Metamorphoses:* Book 1	316	4 789
Vergil	*Aeneid* (Book 6 selections)	178	2 613
	Total:	3 473	53 143

[2] Only Jerome is a post-classical author, writing in the late fourth century.

Table 2.1: Format of the PoS tag in the Latin Dependency Treebank.

Pos.	Feature	Values
1	PoS	All possible values are listed in tables A.1 and A.2.
2	Person	1st (`1`), 2nd (`2`), 3rd (`3`)
3	Number	singular (`s`), plural (`p`)
4	Tense	present (`p`), imperfect (`i`), future (`f`), perfect (`r`), pluperfect (`l`), future perfect (`t`)
5	Mood	indicative (`i`), subjunctive (`s`), imperative (`m`), infinitive (`n`), participle (`p`), gerund (`d`), gerundive (`g`), supine (`u`)
6	Voice	active (`a`), passive (`p`)
7	Gender	masculine (`m`), feminine (`f`), neuter (`n`)
8	Case	nominative (`n`), vocative (`v`), accusative (`a`), genitive (`g`), dative (`d`), ablative (`b`), locative (`l`)
9	Degree	comparative (`c`), superlative (`s`)

The texts are syntactically annotated using a dependency grammar closely modelled on that of the Prague Dependency Treebank, which means that each token is annotated with a link to its head, the token it syntactically depends on, together with a label describing the type of syntactic relation. Each token is also annotated with a morphological PoS tag describing the inflection, as well as the lemma of which the token is an inflected form (Bamman et al. 2007).

The framework developed for annotating and exploring Latin texts in the Perseus project builds upon the dictionary by Lewis and Short (1879). Thus, as in Morpheus, the lemmas, and in particular the numbering scheme used to disambiguate homographic lemmas, correspond to the naming of the dictionary entries. However, it seems that manual editing of the lemma tags during the development of the treebank has introduced some inconsistencies in the way the lemmas are named, such as different spellings of assimilated prefixes (an unstable feature of Latin orthography in general).

In LDT, enclitic particles have been separated, and placed *before* the words they were attached to (both in the sequence of tags in the XML source files and in the numbering of the id-attribute). This process is not always possible to reliably reverse, because the interrogative particle *-nĕ* ends up identical to the negative subjunction *nē*: frequently they are erroneously marked as belonging to the same lemma.

The PoS tags, encoding information about the inflection of the word form as well as the part of speech proper, are all nine characters long, and each position gives information about a separate inflectional feature, as presented in table 2.1. Overall, 436 unique tags are included in the corpus. However, at closer inspection it becomes apparent that the way the different positional features are utilized is somewhat unsystematic. 24 tags which unexpectedly lack values for one or more features occur only once; due to space constraints, these have not been included in tables A.1 and A.2, which describe the existing PoS tags of the treebank. Some similarly deficient tags occur more than once, however; for example, the seven verbs with the tags `v-sp-an*-` are really gerunds and should thus have the value `d` in the fifth position.

PROIEL

An important result of the Pragmatic Resources in Old Indo-European Languages (PROIEL) project at the University of Oslo is a parallel treebank of translations of the New Testament into classical Indo-European languages (Haug and Jøhndal 2008). Apart from the translation by Jerome (the *Vulgate*), the Latin part of the treebank also contains a couple of other texts, by Caesar and Cicero.[3]

Author	Work	Sentences	Words
Jerome	*Vulgate*	9 034	80 532
Caesar	*De Bello Gallico*	1 154	22 408
Cicero	*Litterae ad Atticum*	3 596	40 161
Cicero	*De Officiis*	236	4 230
	Total:	14 020	147 331

Additionally, there is a fifth text: *Peregrinatio Aetheriae*, consisting of 921 sentences and 17 554 words. However, following the example of Skjærholt (2011b), I have excluded it from the following analysis and any future experiment, considering that this 5th century work is written in a Vulgar Latin which exhibits many idiosyncrasies compared to the other works.

Punctuation in PROIEL is handled differently from LDT: instead of being represented as regular tokens and being included in the dependency trees, they are encoded as additional attributes to the regular word tokens. Enclitic particles are separated similarly to the practice in LDT, except that the order of the enclitic and its head word is preserved.

Each token in the PROIEL corpus is annotated with the lemma, a PoS tag denoting the plain part of speech of the lemma, and a morpho-syntactic descriptor (MSD), which encodes information about the morphology of the word form. The PoS tags are two characters wide: the first gives the plain part of speech, with similar values as the PoS attribute in LDT; the second character gives a more fine-grained subdivision of some of the parts of speech, primarily different pronouns.

The MSD consists of ten fields, as described in table 2.2. As can be seen there, the morphological tags encode basically the same information as the PoS tags in LDT, except that the gender attribute is more detailed: if a form is ambiguous with regard to gender, the gender attribute gets a value that covers all alternatives (even if an ambiguous adjective depends on a noun with a certain gender).

Tables A.3 and A.4 give an overview of all PoS and MSD combinations that are used in the corpus. In total there are 959 unique PoS-MSD combinations.

Different but homographic lemmas seem to be disambiguated using the lemma attribute in combination with the part-of-speech tag: i.e., only when the part-of-speech tags differ are two lemmas distinguished (with a numerical index in the lemma tag). Because of this differing practice, there exists no readily apparent way to map the PROIEL lemmas to those used by LDT or Morpheus.

[3] I have been working with source files available from the development platform of PROIEL, `http://foni.uio.no:3000/pages/public_data` (accessed 2015-03-17). That version is slightly extended compared to the one available from `http://proiel.github.io/`.

Table 2.2: Format of the morpho-syntactic descriptor in the PROIEL treebank.

Pos.	Feature	Values
1	Person	1st (`1`), 2nd (`2`), 3rd (`3`)
2	Number	singular (`s`), plural (`p`)
3	Tense	present (`p`), imperfect (`i`), future (`f`), perfect (`r`), pluperfect (`l`), future perfect (`t`)
4	Mood	indicative (`i`), subjunctive (`s`), imperative (`m`), infinitive (`n`), participle (`p`), gerund (`d`), gerundive (`g`), supine (`u`)
5	Voice	active (`a`), passive (`p`)
6	Gender	masculine (`m`), feminine (`f`), neuter (`n`), m/n (`o`), m/f (`p`), m/f/n (`q`), f/n (`r`)
7	Case	nominative (`n`), vocative (`v`), accusative (`a`), genitive (`g`), dative (`d`), ablative (`b`)
8	Degree	positive (`p`), comparative (`c`), superlative (`s`)
9	(Unused)	—
10	Inflection	inflecting (`i`), non-inflecting (`n`)

The Index Thomisticus Treebank (IT-TB)

IT-TB (Passarotti 2010) is a treebank consisting of 265 000 annotated tokens from the works of the 13th century author Thomas Aquinas, syntactically parsed according to the same principles as LDT. The morphological tags are however in a very different format, compared to LDT or PROIEL. With its considerable size it is obviously a valuable resource for the Latin language; however, because it is restricted to a single author, and only covers mediaeval Latin, it has not been used in this study.

2.3 Tagging techniques and programs

2.3.1 HMM tagging with HunPos

The part-of-speech tagger HunPos was developed by Halácsy et al. (2007) as an open source replacement of the well known tagger TnT (Trigram 'n' Tags) (Brants 2000), which is only available in binary form and under a restrictive license. Seeking to improve the tagging of Hungarian, which (like Latin) is a highly inflecting language, they felt the need to extend TnT's functionality to optionally also make use of a morphological dictionary.

Like TnT, HunPos is a supervised stochastic tagger, which implements a hidden Markov model (HMM) to assign a sequence of tags to a sequence of tokens. In other words, a sequence of tags is assigned to a new text using statistics collected during the training phase, where the tagger learns from a manually annotated corpus. Given a sequence of tokens, $w_1, w_2, \ldots, w_T$, the most likely sequence of tags $t_1, t_2, \ldots, t_T$ is generated. To make the calculation of the probability of a tag sequence possible, a HMM tagger operates under the assumption that the probability for any given tag depends on the surrounding tags and corresponding tokens within a small window, and furthermore that the

probabilities are all independent. For each tag, the likelihood that it is correct is estimated as the product of two probabilities:

First, the probability that we would see the word given the tag, $P(w_i|t_i)$. This emission probability (also called lexical or output probability) is estimated based on the relative frequency, which is calculated during training: out of all instances of the tag t in the training data, how many times is it paired with the word w? In HunPos (unlike TnT) this behaviour can be tuned to also take the preceding token(s) into consideration; by default, HunPos operates with the probability $P(w_i|t_{i-1}, t_i)$, but even larger context windows can be used, if so desired, by setting the parameter e (a value of 1 means bigrams, 2 trigrams, and so on; in other words, the current token is not counted).

Second, the probability of the tag given preceding tag(s), the so-called transition (or context) probability. By default, HunPos looks at the two preceding tags, $P(t_i|t_{i-2}, t_{i-1})$; in other words, the tags are handled in groups of three, trigrams. This too is configurable, using the parameter t, and depending on the availability and nature of the training data it may prove better to work with bigrams instead ($t = 1$) – or longer n-grams. Similar to the emission probabilities, the transition probabilities are estimated using the relative frequencies seen in the training set.

Since the training data is likely to be relatively sparse, in that not all possible trigrams are sufficiently represented, if at all, the probabilities are estimated as a weighted sum of the relative frequencies for unigrams, bigrams and trigrams:

$$P(t_i|t_{i-2}, t_{i-1}) = \lambda_1\hat{P}(t_i) + \lambda_2\hat{P}(t_i|t_{i-1}) + \lambda_3\hat{P}(t_i|t_{i-2}, t_{i-1}),$$

where $\hat{P}$ signifies the maximum likelihood estimate. The weights λ_n are calculated using the deleted interpolation method; they are calculated once, and hence do not depend on the particular trigram (Brants 2000).

The total probability for a certain tag sequence coupled with a given token sequence is finally calculated as the product of all emission and transition probabilities, and the tag sequence that has the largest probability is chosen. With the default window sizes, we get the following formula:

$$\underset{t_1 \ldots t_T}{\operatorname{argmax}} P(t_{T+1}|t_T) \prod_{i=1}^{T} P(t_i|t_{i-2}, t_{i-1})P(w_i|t_{i-1}, t_i)$$

The optimal tag sequence, which maximizes the compound probability, is found dynamically using the Viterbi algorithm. To speed up the processing time of the algorithm, a beam search is used, whereby some states are discarded according to a certain threshold. In theory, this means that the algorithm is not guaranteed to find the optimal solution, but with a proper tuning of the width of the beam search, the actual difference is negligible (Brants 2000).

Unseen words, i. e. words not found in the training data, pose a problem, seeing that their emission probabilities can not be estimated using the procedure described above. They can be handled in different ways; the solution adopted in TnT is to assume that an unseen word has a tag distribution similar to *rare* words with the same suffix as the unseen word. In HunPos, the threshold for what is considered a rare word, as well as the maximum suffix length that is considered, are both configurable.

Additionally, HunPos adds the possibility to use the information in an external morphological lexicon. HunPos will then restrict the set of possible tags for an unseen word to those given in the lexicon. The tags are then weighted as before, using statistics from words with similar suffix.

2.3.2 RFTagger

RFTagger (Schmid and Laws 2008) is an HMM tagger which was developed with the aim to improve on the problem with sparse data associated with large tag sets. With detailed morphological tags, the number of tag n-grams naturally increases dramatically, compared to the simpler tag sets commonly used for more isolating languages like English. As a consequence of this, a training set of a given size will cover a smaller percentage of all the tag n-grams naturally occurring in the language, which in turn means that the tagger will be less likely to produce the right tags when run on new data. Smoothing can only partly overcome this difficulty. The adopted solution is to decompose the tags into feature vectors, instead of regarding them as atomic units. The transition probability factor in the HMM, which in HunPos is conditioned on the preceding tags, is then replaced by a product of attribute probabilities.

The other concept that sets RFTagger apart from HunPos is the way these morphological attribute probabilities are calculated: instead of conditioning them on all the features of the closest preceding tags, the most relevant combination of features are chosen by the creation of decision trees.

A separate binary decision tree is built for each value of every attribute. For example, the decision tree for estimating the probability of the current tag having the value "ablative" for the case attribute may hypothetically first check whether or not the plain part of speech of the preceding tag is "preposition", then, if not, check if the preceding case attribute is "ablative" as well, and so on, until ending up on a leaf giving the probability.

Each test in the decision tree checks a feature from one of the preceding tags within a certain, configurable context (c); default is 2 (trigrams) but Schmid and Laws report improvements with contexts as large as 10.

Without restricting the growth of a tree during the training phase, there is the risk that the tree may include increasingly specific tests, based on increasingly rare evidence, and thus overfits the model to the training data. To avoid this, a pruning strategy is implemented whereby the number of samples a test node is based on is multiplied by the information gain of the test (see Schmid and Laws (2008) for details); if the resulting value falls below a certain threshold p (by default set to 6), the node is pruned.

Amongst other parameters can be mentioned that RFTagger, like HunPos, can make use of a supplementary lexicon, providing information about unseen words, or additional tags to words present in the training data. By using the argument `-s`, unseen sentence initial words will be treated as their corresponding lower-case forms. A couple of other parameters can be set, mostly relating to different levels of smoothing, but these have not been used in this work.

2.3.3 MATE tools

Because of Latin's relatively free word order and rich morphology, there is, as pointed out by Lee et al. (2011) amongst others, a considerable interaction between the morphology and the syntax, such that it often is the case that a sentence has to be syntactically parsed in order to successfully disambiguate between different morphological analyses of a word: an ambiguous word may agree with (depend on) a distant word, and the relationship between them can not satisfactorily be captured by tagging techniques operating on a smaller context. Unfortunately, automatic parsing often relies on having tagging done in advance, which causes a kind of chicken-and-egg problem: if a word is erroneously tagged, the parser may not be able connect it to the word it agrees with. The solution that would benefit both tasks is to perform the tagging and the parsing simultaneously; this was successfully demonstrated with Latin by Lee et al. (2011) (see section 2.4.2 for more details on their experiments).

MATE tools is a publicly available suite of NLP software, which include an experimental transition-based dependency parser performing joint parsing and morphological tagging (Bohnet and Nivre 2012; Bohnet et al. 2013). In transition-based dependency parsing, the future nodes (each labeled with a token) are first placed in a queue, and a stack is initialized with a root node. The dependency tree is then built by the application of transition rules, which change the parser configuration in different ways: the Right-Arc transition creates a labelled directional arc from the node on top of the stack to the second topmost, and removes it from the stack; the Left-Arc transition makes an arc in the opposite direction, and thus removes the second topmost node; the Shift transition moves the frontmost node from the queue to the stack. To these is added the Swap transition which can reorder the nodes, so that non-projective trees can be built (Nivre 2009). At the end, the queue is empty, and all nodes have been incorporated in the tree.

The probability for a transition to occur is based on a wide range of features from any part of the current parser configuration, weighted according to a weight vector learned from the training data. The best sequence of transitions is found using a beam search.

To perform the joint tagging in MATE tools, the shift transition is modified to also tag the node. Bohnet et al. (2013) describe several different operating models for the parser, but according to the configuration that gave the best results on most of the tested languages, the plain PoS tags and the morphological descriptors are selected independently. The morphological tags themselves are however regarded as atomic, to avoid risking creating inconsistent descriptors.

2.4 Previous work

2.4.1 The Māccer macronizer

Māccer, an online application for macronizing Latin text developed by Felipe Vogel, was made public in January 2015.[4] It does not perform any morphological analysis, but instead relies on a word list of annotated forms, collected from

[4] `http://fps-vogel.github.io/maccer/`

various macronized texts available online. Unknown words are clearly marked as such in the output.

There is a rudimentary disambiguation of homographic forms in place, based on the frequency of the different macronized forms in the source texts. If the difference in frequency is not large enough, the differing vowels are marked as ambiguous, being left for the user to correct manually. From the information available on the website, the system does not seem to take any contextual information into consideration. Plans for the future allegedly include expanding the word key with automatically generated inflected entries.

2.4.2 Latin PoS tagging

Using the TreeTagger program trained and evaluated (with 10-fold cross-validation) on an early version of the Latin Dependency Treebank consisting of 30 457 tokens, Bamman and Crane (2008) achieved 95% accuracy when resolving the plain parts of speech, and 83% accuracy in assigning the full morphological tags. When analysing the separate morphological features, case and gender proved most difficult to get right.

Poudat and Longrée (2009) performed an extensive examination of two taggers, MBT and TnT, using different parts from the LASLA corpus, particularly with the aim to test the different taggers' sensitivity to stylistic, diachronic, generic and discursive variations. A third tagger, TreeTagger, was also considered, but was discarded since it could not handle the extremely large tag set of LASLA, consisting of 3 732 tags. For all experiments, the taggers were evaluated on the same five test sets, each from a separate author (Caesar, Sallust, Quintus Curtius, Cicero, and Catullus). A large number of increasingly larger training sets were used, with the following general results: TnT had consistently better accuracy than MBT, surpassing it by on average 4.8 percentage points; training on texts by the same author as that of the test set gave better results than when testing on other authors, but, when training and testing on the same author, the results increased if the training set also encompassed additional authors – in other words, more data is always beneficial.

Further processing of the results of Poudat and Longrée (2009), by calculating the average accuracy over all the five separate test sets (weighted according to their respective size), reveals that the best accuracy when using a training set of moderate size (138 000 tokens) was 78.99%, achieved by TnT trained on a collection of historians from the first century BC. With a larger training set, 352 000 tokens, including also later historians, the accuracy increased to 84.04%. Extending the training data further with a large part of the Ciceronian corpus, reaching a total of 607 000 tokens, increased the average accuracy of TnT marginally to 84.09%.

Passarotti (2010) briefly reports on an experiment where HunPos achieved 96.78% accurate PoS tags when trained and tested on the IT-TB (61 024 tokens), and 89.90% accuracy on the full morphological tags.

With their system performing joint morphological tagging and dependency parsing, Lee et al. (2011) achieved a slight increase in the accuracy of the morphological tags compared to how the system performed when the tagging was done separately. Especially adjectives and participles benefited from the joint approach. Lee et al. present their accuracy scores only for the individual

attributes: on average, the scores (from the joint model) are higher than those achieved by Bamman and Crane (2008). This was especially the case with gender (1.0 percentage point increase) and number (0.75 p. p. increase), while the accuracies of tense, mood and voice were slightly lower (by 0.4, 0.4 and 0.3 percentage points, respectively).

Skjærholt (2011a) ran a large number of experiments using various parts of the PROIEL corpus and two different taggers: TnT and Wapiti, the later being a sequence labeller based on so-called conditional random fields (CRF). Amongst the results can be noted 96.87% accuracy on plain PoS tags with TnT, using cross-validation on *De Bello Gallico* and the *Vulgate*. The corresponding accuracy of the MSD tagging was 88.9%. The CRF models did not give substantially better results than HMM tagging with TnT; Skjærholt however speculates that they would benefit from more training data.

3 Methodology

3.1 System overview

To perform automatic annotation of Latin vowel lengths, I propose a system based on common NLP modules (a tokenizer, a lemmatizer, and a PoS tagger) in combination with the Morpheus morphological analyser. Either the Latin Dependency Treebank (LDT) or PROIEL can serve as training data to the tagger; for the lemmatizer, only LDT can be used, because the lemmas found in PROIEL are incompatible with those in Morpheus.

The text which shall be annotated is thus first tokenized according to the standard of the corpus used to train the statistical tools. The set of unique tokens is then sent to Morpheus, which for each token reports the possible analyses, in the form of triples consisting of a morphological tag, a lemma, and the corresponding word form with vowel lengths marked. The tokenized text is then fed to the PoS tagger and the lemmatizer, which may make use of the Morpheus analyses to improve their results. For each token, the predicted morphological tag and lemma are then matched to one of the analyses given by Morpheus for the token. The resulting series of macronized words is finally detokenized, to produce a copy of the original text with marked long vowels. The whole workflow is described schematically in figure 3.1.

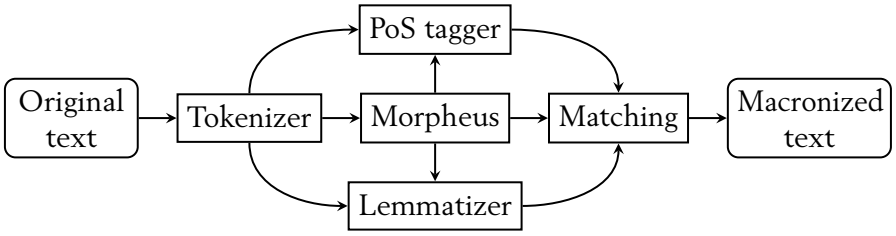

Figure 3.1: Workflow of the vowel length annotating system.

The system thus ultimately depends on Morpheus to provide the right vowel lengths. While it, to the best of my knowledge, is the best open source resource for Latin morphological analysis and corresponding vowel lengths, it was unfortunately found that the reported macronized forms could not always be relied upon. In particular, vowel lengths in inflectional endings were often insufficiently marked. Many errors had also crept in during the automatic conversion of the original dictionary (Lewis and Short 1879); for example, it was not uncommon to find erroneously segmented stems. Thus, as a prerequisite to our vowel length annotation system, the database and inflectional rules of Morpheus had to be reviewed and amended: 25 vowels in inflectional endings

were corrected, and 638 individual word forms or stems were modified in some way. The resulting patched version, which the rest of this thesis is based upon, is available on GitHub.[1]

In order to compare and map the output from the PoS tagger to the Morpheus analyses, a conversion between their respective formats has to be performed. This functionality was implemented in the form of a Python script, which parses the human readable Morpheus analyses and stores them in an intermediate format, which in turn can be converted to tags compatible with either corpus.

To ensure the soundness of the conversion script, the extent to which the converted Morpheus analyses cover each corpus was evaluated using a metric borrowed from Skjærholt (2011a, p. 47): for each unique word form in the target corpus that is known to Morpheus (i. e. that can be analysed in at least one way), the set of tags applied to that word form throughout the corpus is collected; if a tag from this set is not found in the converted Morpheus analyses, it is counted as missing. The total number of missing tags is compared to the total number of unique word form and tag pairs in the corpus (again disregarding those word forms unknown to Morpheus). Using this method, 5% of the tags in LDT are missing from the converted Morpheus analyses. The corresponding ratio for the PROIEL corpus is 9%. This could be compared to the 11% reported by Skjærholt, who performed a similar conversion of tags in the LDT format to the PROIEL corpus, using a series of manually applied rules.

Obviously, it will not always be the case that a predicted tag and lemma pair is amongst the Morpheus analyses of a given word. The possible reasons for this are many: it could be because of an erroneous tag or lemma, or a deficiency in the morphological analyser or its lexicon, or a problem with the conversion between the output from Morpheus and the tag format used in the corpus. In any case, the analysis with the most similar morphological tag is chosen, being the best available suggestion. The predicted lemma is taken into consideration only when two or more analyses have equally similar tags. If the word form is unknown to Morpheus, no vowels are marked.

The similarity between two tags is calculated in a fairly straightforward way: for each attribute (position in the tag) the two values are compared, and the differing attributes are counted. Participles are a notable exception to this principle: because of their close relation to adjectives (and nouns!), and consequent inconsistencies in the corpus tags and the Morpheus analyses, it seemed best to modify the calculated distance in that case: using LDT tags as an example (cf. table 2.1), the distance between the participle `v-sppamg-` and the adjective `a-s--ng-` is regarded as being two steps, one for the differing part of speech, and one for the differing gender (masculine vs. neuter); in other words, when participles are compared with adjectives or nouns, the tense, mood and voice attributes are ignored. In practice though, it might be doubtful whether this ad hoc exception has any actual influence on the annotation result.

If it is necessary to select between analyses on the basis of the lemma, the one with the most similar lemma is chosen, according to the editing distance. This is to accommodate to different spelling conventions and occasional spelling errors.

[1] `https://github.com/Alatius/morpheus/releases/tag/thesis`

3.2 Tokenization

Tokenization in Latin is for the most part an uncomplicated matter; the only real difficulty lies in correctly separating the enclitic particles, mainly *-que*, *-ne* and *-ve* (Lee et al. 2011; Skjærholt 2011a). They can be postfixed to any word, without any apostrophe or other indication that the resulting word form is a composite. Occasionally, this may give rise to genuinely ambiguous forms, such as *domine*, which can be either the noun *domĭnus* inflected in the vocative case ('O master') or the adverb *domī* ('at home') joined with the interrogative particle *-ne*.

This has been addressed in a rudimentary fashion: if a word that ends in what looks like an enclitic particle is a correct Latin word in itself (according to Morpheus) it is as a rule not divided. A small class of border cases (notably *neque*, 'nor'), which are divided in the corpora but included as single words in Morpheus and lexica, are excepted from this rule, and are thus divided.

This has the effect that *domīne* can never become marked as such (with long *ī*). In practice though, this is no big problem, as an analysis of PROIEL reveals that this situation is very rare: out of 147 331 tokens, 1 587 are enclitics, and only once does an enclitic combine with the preceding word to form an ostensibly correct form in itself.[2]

3.3 PoS tagging

With a large range of different PoS taggers at our disposal, the question naturally arises which of them is best suited as the tagging module in our vowel annotating system. A couple of PoS taggers were selected for evaluation based on their availability, license, ease of use, speed, interesting features, or documented good performance with large morphological tag sets: HunPos, RFTagger, and the experimental MATE tools.

With regard to training resources, there are, as previously mentioned, two corpora which are suitable for our purposes: LDT and PROIEL. Each has things speaking for and against it: LDT, while smaller, is representative for a larger range of classical literature, and its lemmas and tags are more similar to those used by Morpheus. PROIEL on the other hand is almost three times as big, but consists mainly of the less classical Vulgate. Instead of settling for one of them, I have opted to evaluate the taggers on both corpora in parallel.

The sentences in each corpus were first randomly mixed, before being partitioned into training, development, and test sets. There are both pros and cons with this approach: on the one hand, the corpus becomes unnaturally uniform, which means that the percentage of unknown words in the test set will be smaller than in a real case scenario (Brants 2000); on the other hand, the risk is minimized that an unfortuitous random choice of test set skews the final result.

[2] From Cicero, *Ad Atticum* 1.17.6: *In pūblicāne rē?* ('In public affairs?'). Here, *pūblicā* is an ablative adjective, but *publicane* looks either like a vocative (*pūblicānĕ*) or an adverb (*pūblicānē*). If tagged as the first, the vowels happen to come out correct anyway; in the other case, the final *-e* is erroneously marked as long. Of course, in either case the faulty tokenization may have repercussions on the tags of the surrounding words.

The shuffled corpora were divided in the following way: 10% was set apart as the test set; the remaining 90% was then divided into ten equally large parts, which were used to perform 10-fold cross-validation when optimizing the parameters of the taggers. Once the best configuration for a tagger was found, it was then trained on all ten parts combined, and tested on the as of yet unseen test set, to give a fair estimation of the tagger's performance.

The results from the tagging experiments were evaluated both intrinsically (in terms of tagging accuracy, TA) and extrinsically, i. e., with respect to how much the predicted tags help in annotating the vowel lengths when plugged into the proposed vowel annotating system. In order to isolate the effect the taggers have on the end result, this was done by letting the system mark vowels in each development/test set, and then counting the percentage of macronized words that are identical to the annotation we get when using the morphological tags from the gold standard instead (all other things being equal). Because Morpheus compatible lemmas are not available in the PROIEL corpus, all vowel length annotations (both when using corpus tags and predicted tags) are done without information about lemmas: if several analyses match the tag, the first one reported by Morpheus is chosen. In this way the vowel length accuracy based on tags (VLA_t) is calculated. To set the scores into perspective, we can compare them with the results from a baseline tagger, which automatically tags punctuation correctly, and for the rest of the tokens simply chooses the first Morpheus analysis; unknown words are then automatically counted as wrongly tagged.

When evaluating the performances of different taggers, the difference between the final results were tested for statistical significance using McNemar's test, as suggested by Dietterich (1998): Let a denote the number of tokens tagged correctly by tagger A but mistagged by B, and, vice versa, let b be the number of tokens correctly tagged by B but not by A. The null hypothesis is that A and B perform equally well, in which case a would be more or less equal to b. McNemar's test is used to estimate whether the difference is statistically significant using the following test statistic:

$$\chi^2 = \frac{(|a - b| - 1)^2}{a + b}$$

Provided that the null hypothesis is correct, the probability p that $\chi^2 > 3.84$ is less than 0.05, which is a commonly accepted threshold. Thus, if $\chi^2 > 3.84$ the null-hypothesis is rejected, and we conclude that the difference between the two taggers is statistically significant. In the same way the significance of differences between vowel length annotations is estimated.

Along with the optimization of the first tagger, HunPos, the two corpora were preprocessed in a couple of different ways, to establish which versions should be used for the best results. The following modifications were tested:

- Punctuation is a feature that is treated differently in the two corpora; while LDT includes punctuation marks as individual tokens, in PROIEL they are only represented as additional attributes to the regular word tokens. The question then is whether punctuation marks should be included in the input to the taggers.

- As noted in chapter 2.2.2, the Latin Dependency Treebank includes a couple of non-standard tags, which lack one or more expected morphological attributes. Is the presence of these tags in the training data detrimental to the vowel length annotation? To answer that question, an alternative version of the treebank was developed, with 76 modified tags.[3]

The variants of the corpora which were found to give the best results together with HunPos were then used in all the following experiments, under the assumption that the results would carry over also to the other taggers.

HunPos and RFTagger were both optimized with regard to the parameters that were judged to have the most influence on the tagging results, primarily the sizes of the contexts used to estimate the conditional probabilities in the HMM models. Additionally, for HunPos the rare words frequency threshold was optimized; for RFTagger the tree pruning threshold.

Unfortunately it was not possible to optimize MATE tools in the same way as HunPos and RFTagger, due to the very long training times. Instead it was run with the configuration that gave the best results on Czech, German and Hungarian according to Bohnet et al. (2013), namely with soft lexical constraints in the form of the Morpheus lexicon, and with 800 word clusters generated from a plain text corpus of about 10 000 000 tokens created by concatenating the Latin texts available from the Perseus Project.[4]

3.4 Lemmatization

Lemmatization is a necessary component in the system in order to successfully disambiguate between identical word forms with identical tags. Similar to how different taggers were evaluated both intrinsically and extrinsically, we can evaluate a lemmatizer by calculating both the lemma accuracy (LA) and the accuracy of the macronized words we get with the help of the lemmas, VLA_l.

The way the VLA_l measure is calculated is mostly analogous to how the VLA_t score is defined, though with some important differences: the vowel length gold standard that is used as a basis for the accuracy count is now generated from the LDT corpus only, by matching, for each token, *both* the tag and the lemma to the Morpheus analyses. For those tokens where such a matching is possible, the lemmas from the lemmatizer are used in conjunction with oracle tags to select a macronized word form according to the principles described above. VLA_l is then defined as the percentage of the macronized words that are identical to the corresponding macronized words in the generated gold standard.

Three lemmatization strategies were evaluated:

1. A baseline lemmatizer that marks punctuation correctly, and for the rest of the tokens chooses the lemma from the first Morpheus analysis. If unknown to Morpheus, the word form itself is chosen as lemma.

2. A basic lemmatizer that chooses the lemma which is most commonly seen in the training data together with the word form. If however the

[3] `https://github.com/Alatius/treebank_data/releases/tag/thesis`

[4] `http://www.perseus.tufts.edu/hopper/opensource/download`

word form is unseen, it is analysed by Morpheus, and out of the proposed lemmas the one is chosen that is most common in the training data (regardless of word form). If none of the lemmas occurs, the lemma of the first analysis is chosen. If no analysis is available, the word form itself is copied as the lemma.

3. PurePos, a reimplementation of HunPos in Java, with added lemmatizer functionality (Orosz and Novák 2013).

PurePos was optimized using 10-fold cross-validation on 90% of LDT, in the same way as in the tagger optimization. Afterwards, the three lemmatization approaches were evaluated on the unseen test set.

3.5 System evaluation

In order to assess the complete system's performance in a real usage scenario, four texts with marked vowel lengths were chosen: *Fabulae Faciles* (Ritchie 1903), *Alicia in Terra Mirabili* (Carroll 2011), the first book of Tacitus' *Annales* (Winge 2008), and *Life of Hannibal* by Cornelius Nepos (Mulligan 2013). The first two are neo-Latin texts, written in a fairly simple language, with short uncomplicated sentences; the other two are authentic classical works.

These experiments also give us an opportunity to finally decide which of the two corpora that works best as training data for the PoS tagger: for this purpose, the system was run with the tagger trained on either of the two corpora. A couple of other systems were also tested, implementing some less sophisticated methods for solving the vowel length annotation problem:

- A baseline system, which for each word simply gives the vowel lengths of the first reported Morpheus analysis.
- A basic system, which looks up each word in Morpheus, and chooses the macronized word form that is most common in LDT (i. e., in the macronized text gotten from LDT by matching tokens, tags, and lemmas to Morpheus analyses). This simple statistical method is comparable to the strategy used by the Māccer macronizer (see chapter 2.4.1).
- An oracle system, which chooses the macronized word form from Morpheus that is most similar to the one in the annotated text. This gives the upper bound for what the system can achieve with optimal[5] lemmatization and tagging, without improvements done to the Morpheus lexicon.

For the evaluation of these final results, a more straightforward measure has been used, namely the plain vowel length accuracy (VLA), i. e., the percentage of vowels that were correctly marked. By considering individual vowels (instead of counting whether whole words are correctly macronized or not) we get a more fine-grained measure, which better adapts to different conventions, such as different treatments of hidden quantities (cf. chapter 2.1). It also directly measures the amount of editing that has to be performed to correct the automatically annotated text.

[5] Optimal in a pragmatic sense, not necessarily optimal from a linguistic perspective.

4 Results

4.1 Tagger optimization

4.1.1 HunPos

With all the tested taggers, the user has the option to provide a lexicon, which for each unknown token (i. e., not present in the training data) in some way restricts the range of available tags that can be applied. Since the purpose of the tagging module in the system is to find tags which are to be matched with the Morpheus analyses, it makes perfect sense to use the output from Morpheus as lexicon. To make sure that doing so indeed boosts the performance, HunPos was run with and without this feature (with default settings otherwise). Finally, the tagging was also performed using the corrected version of LTD. The collected results for the two corpora are presented in table 4.1. The positive effect of a lexicon is unmistakable. Using the corrected corpus benefits the tag accuracy, although the increase in vowel length accuracy is negligible. In the following experiments, the corrected version of LDT is used, unless otherwise noted.

To choose between the tags provided by the lexicon, HunPos uses statistics from rare words with the same suffix as the word to be tagged. The maximum frequency that a word may have in the training data to be regarded as rare, f, is set to 10 by default, but can be modified; figure 4.1 shows how the performance depends on this frequency threshold. The value resulting in the best VLA_t on both corpora, 80, was then chosen for the following experiments. Different values for the maximum suffix length (s) was also tested in tandem, but that parameter did not affect the results in any substantial way, and has thus not been included in the diagrams.

Punctuation in Latin generally gives information about the syntactic structure of the sentence, which might be valuable when tagging. On the other hand, maybe a possible positive effect is negated by increased sparseness in the data? To test this, HunPos was trained on two versions of each corpus, with or

Table 4.1: The effects of a external morphological lexicon and corrected corpus when tagging with HunPos, using 10-fold cross-validation on 90% of each corpus.

Lexicon	Corrections	LDT		PROIEL	
		TA	VLA_t	TA	VLA_t
–	–	79.49	97.46	87.84	98.91
+	–	84.66	97.56	89.45	98.96
+	+	84.73	97.57	—	—

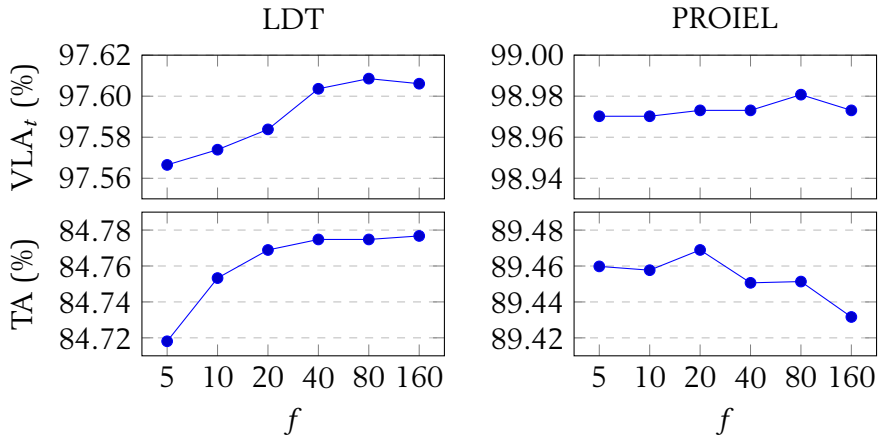

Figure 4.1: HunPos with varying rare words frequency threshold (f), using 10-fold cross-validation on 90% of each corpus.

without punctuation present. For this experiment, only the accuracy of tags belonging to non-punctuation (TA$_w$) was evaluated.

Optimization of the emission (e) and transition context (t) parameters were performed in tandem with the punctuation evaluation, since it seemed reasonable to suspect that the absence or presence of punctuation may influence the optimal choice of these parameters. The results encompassing the highest values are presented in figures 4.2 and 4.3. Larger values for e and t were also tested, but did not give any improvements.

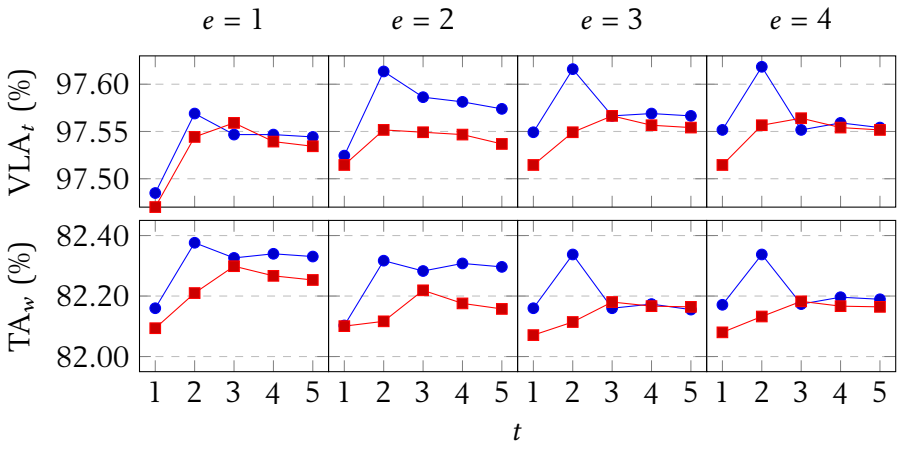

Figure 4.2: HunPos on LDT, with (—●—) and without (—■—) punctuation, with varying transition (t) and emission (e) contexts.

Including the punctuation from PROIEL thus proves beneficial to both the tagging and the vowel length annotation.

On LDT, HunPos consistently performed best with $t = 2$ (trigrams). Increasing the emission context past $e = 1$ improved VLA$_t$ slightly when using punctuation, even though the tag accuracy actually decreased. On PROIEL, the results are more regular. Even larger values for e and t were tested, but those did not lead to any improvements.

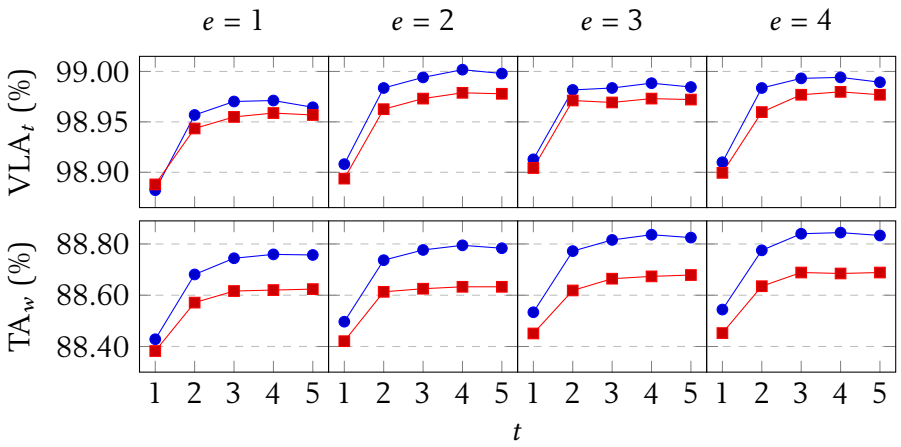

Figure 4.3: HunPos on PROIEL, with (—●—) and without (—■—) punctuation, with varying transition (t) and emission (e) contexts.

4.1.2 RFTagger

The optimization of RFTagger was done with regard primarily to two parameters: the size of the context window, c (corresponding to the transition context t of HunPos), and the tree pruning threshold, p. The results at different values of these parameters are reported in figure 4.4.

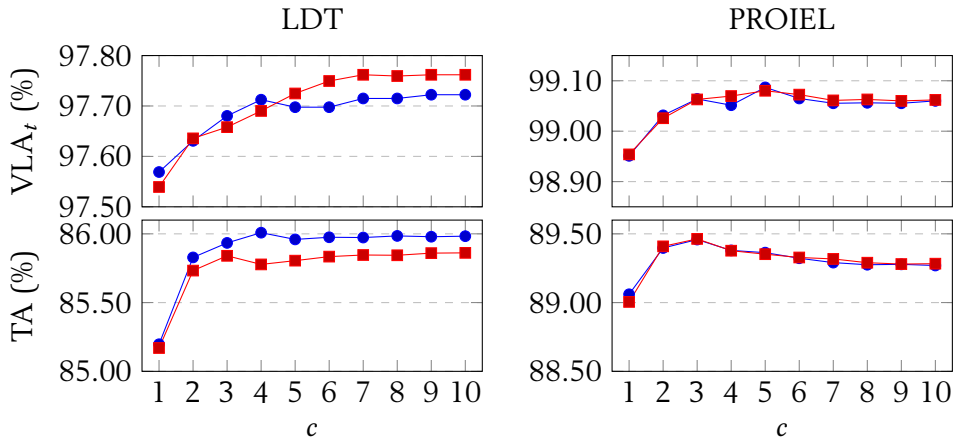

Figure 4.4: RFTagger with different context sizes (c) and two different tree pruning settings, $p = 6$ (—●—) and $p = 7$ (—■—).

The default value of $p = 6$ seems well balanced, but with larger context windows the vowel length annotation of LDT benefited from slightly more aggressive pruning. Even higher and lower values, $p = 5$ and $p = 8$, were tested, but did not improve the performance.

Preliminary tests revealed that using the `-s` argument to treat unseen sentence initial words in the same way as the corresponding lower-case forms consistently increased both VLA$_t$ and TA slightly (the later by about 0.03 percentage points); that setting has therefore been used in all the experiments.

4.2 Tagger comparison

For the comparison of the different taggers, the configurations that during the optimization gave the highest VLA_t scores on each corpus were chosen. The taggers were then trained on the 90% of each corpus that previously had been used for cross-validation, and were tested on the as of yet unseen test set. The resulting performances of the three taggers, plus those of the baseline tagger, are reported in table 4.2.

Since MATE tools works with treebank data, and punctuation is not included in the dependency trees in PROIEL, the TA score for MATE tools on PROIEL had to be adjusted so that it could be compared with the scores from the other taggers: similarly to the baseline tagger, the punctuation was then automatically counted as correctly tagged.

Table 4.2: The tagging and vowel length annotation performances of HunPos, RFTagger, and MATE tools.

Corpus:		LDT			PROIEL	
Tagger	Settings	TA	VLA_t	Settings	TA	VLA_t
Baseline	—	64.68	87.87	—	51.49	91.46
HunPos	$e = 4, t = 2$	85.43	97.48	$e = 2, t = 4$	90.02	99.01
RFTagger	$c = 7, p = 7$	**86.73**	97.75	$c = 5, p = 6$	89.68	**99.13**
MATE tools	—	85.13	**98.24**	—	**90.42**	99.06

However, while suggestive, it should be noted that not all of these results can be demonstrated to be significantly different, according to McNemar's test. The significant results are collected in table 4.3 (excluding comparisons with the baseline tagger). Note that none of the VLA_t scores on PROIEL are proven to be significantly different.

Table 4.3: Statistically significant differences between the taggers.

Corpus	Measure	Better tagger	Worse tagger	Significance level
LDT	TA	RFTagger	HunPos	$p < 0.01$
LDT	TA	RFTagger	MATE tools	$p < 0.001$
LDT	VLA_t	MATE tools	HunPos	$p < 0.01$
LDT	VLA_t	MATE tools	RFTagger	$p < 0.05$
PROIEL	TA	MATE tools	RFTagger	$p < 0.01$

Both RFTagger and MATE tools thus compare favourably with HunPos. A strange outlier is however the low TA of MATE tools on LDT. Thankfully, this was compensated with a very good VLA_t score.

While our study is not concerned with dependency parsing per se, it may be interesting to note that MATE tools achieved 65.85 UAS (unlabelled attachment score) on LDT, and 56.80 LAS (labelled attachment score); these far from stellar results are further indications of the problems MATE tools had with this corpus. On PROIEL the corresponding scores are more reasonable, with 80.07 UAS and 73.05 LAS.

4.3 Lemmatization

4.3.1 Optimization of PurePos

The best settings for emission (e) and transition contexts (t) when using PurePos to predict lemmas were found using the same setup as in the tagger optimization. Two settings for the rare word frequency threshold were also tested, the default value $f = 10$, and the value that gave the best VLA_t, $f = 80$. The later gave consistently better results, and is the only one that is represented in the diagram.

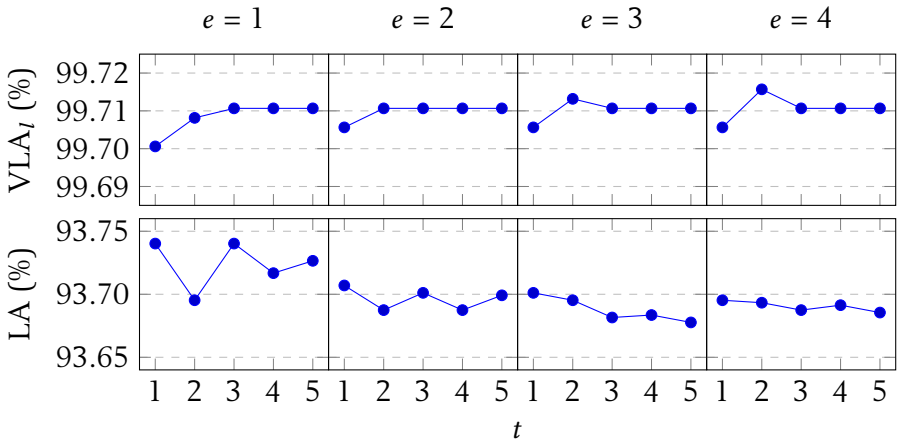

Figure 4.5: Lemmatization with PurePos, using 10-fold cross-validation on 90% of LDT, with varying transition (t) and emission (e) contexts; in all experiments, $f = 80$.

As can be seen in figure 4.5, VLA_t benefited marginally from increased emission context e, at least for $t = 2$. Lemma accuracy (LA), on the other hand, declined both with increasing emission context and increasing transition context. These general results are in many ways reminiscent of the the results from the tagging experiments with HunPos on LDT (with punctuation): cf. figure 4.2. One notable difference is the low LA when $t = 2$, compared to the other measures.

4.3.2 Lemmatizer comparison

As in the tagger comparision, the evaluation of different lemmatization strategies was carried out by training on 90% of the corpus (LDT), and testing on the test set. PurePos was run with the settings that had given the best VLA_t during the optimization, i. e., $e = 4, t = 2, f = 80$. The results for PurePos and the two reference lemmatizers are reported in table 4.4:

Table 4.4: Lemmatizer performances on the LDT test set.

Lemmatizer	LA	VLA_l
Baseline	81.86	97.34
Basic	**94.48**	99.81
PurePos	93.98	**99.86**

PurePos just about beat the Basic lemmatizer on VLA_l, with two more correctly macronized words (out of 4 246), which of course is not statistically significant; nor can the difference between their LA scores be shown to be significant using McNemar's test.

It may nevertheless seem surprising that PurePos got lower LA than the Basic lemmatizer. Further investigation into the differences between the two lemmatizers reveals that this is because PurePos relies more on the lemmas in the lexicon, which not always are spelled exactly as those in the corpus. Because the macronized word forms are selected by matching lemmas using the Levenshtein edit distance measure, those spelling differences are inconsequential when VLA_l is calculated.

In view of these results, it seems doubtful whether it is worth the trouble to invoke PurePos as an external lemmatizer, seeing that the easily implemented basic strategy gives more or less the same end result, and that in a shorter amount of time.

4.4 System evaluation

For the final evaluation, the system was configured to run with RFTagger as the PoS tagger, trained on either LDT or PROIEL, and set to mark long vowels in the four chosen texts. The vowel length accuracies (VLA) for these two setups, along with the accuracies achieved by the more rudimentary reference systems are reported in figure 4.6. MATE tools was also considered as tagger, but preliminary testings revealed no improvements to the VLA scores, and therefore the dramatic difference in tagging time between the two programs made the balance tip in favour of adopting RFTagger instead. For lemmatization, the Basic approach was chosen, considering its fully satisfactory results and easier integration, and also because PurePos was quite slow in comparison.

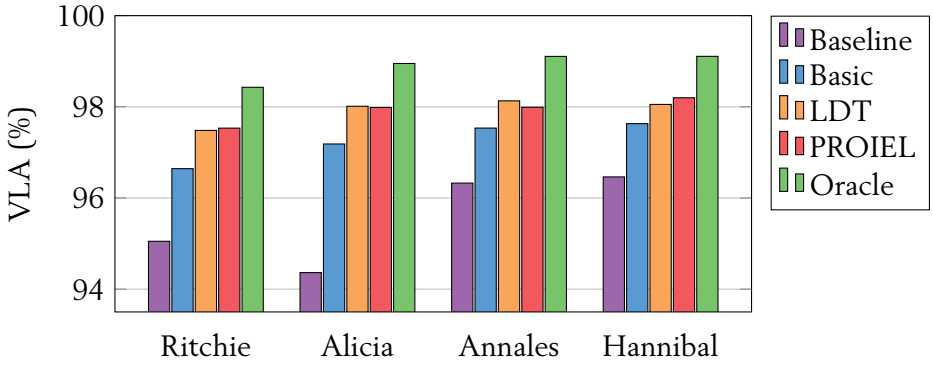

Figure 4.6: Vowel length accuracy using LDT vs. PROIEL to train RFTagger, compared to various reference systems.

The first thing to notice is that already the basic system gives quite good results, with at least 97% VLA on most texts. Both the basic and the baseline system are slightly better at predicting the vowel lengths in the classical texts (*Annales* and *Hannibal*) than in the neo-Latin texts (*Ritchie* and *Alicia*).

With the help of lemmatization and PoS tagging (trained on either LDT or PROIEL), the average increase of VLA on the two neo-Latin texts, compared to the performance of the basic statistical system, was 0.8 percentage points. On the classical texts the improvement was smaller, about 0.5 p. p. on average. This may be because they are more difficult to tag, or, perhaps more probable, because the already good results from the basic system give less room for improvements: discounting the *Ritchie* text, the accuracy is fairly stable at or slightly above 98%.

The overall lower scores on *Ritchie* can be explained by the fact that its vowel lengths were annotated almost a century ago: since then, progress in historical linguistics have changed our conception of which vowels are short and which are long in a number of cases. Making up for these changes, the results are very similar to those of *Alicia* (except for the baseline system).

As for the different corpora, the choice seems not that important. The difference in performance between the system trained with LDT and the one trained with PROIEL is statistically significant only on *Annales* ($p < 0.05$), where LDT comes out slightly ahead of PROIEL (98.13% vs. 97.99% VLA).

To gain a better understanding of the nature of the errors that prevent the system from achieving the same results as the human annotators, the shortest text, *Hannibal*, was picked out for a more careful investigation. Each vowel that had got a different length ascribed to it (using PROIEL as training data) than in the human created gold standard was analysed, and the reason for the discrepancy was counted as belonging to one of five categories:

- 33 vowels had got their lengths erroneously classified because of an incorrect tag.
- 9 were due to faulty lemmatization.
- 20 were incorrectly marked in Morpheus' lexicon, or belonged to words that were not known at all.
- 45 discrepancies could be attributed to different vowel length marking conventions.
- 3 had in fact got the correct length, but were incorrectly marked in the gold standard.

The choice between some of these categories, particularly the last three, is naturally somewhat arbitrary. Cases where the vowel was in a closed syllable, and the vowel length is hidden and thus more difficult to know for sure, have been counted as being due to different conventions.

5 Discussion and conclusion

With the help of morphological PoS tagging and lemmatization, the accuracy of our vowel length annotation system has been shown to increase by between 0.5 and 0.8 percentage points, compared to the performance of a basic statistical system (figure 4.6). While this may not seem much, it results in a decrease of the necessary manual post-editing by at least 20% (up to 50% with an improved lexicon).

The results from the tagging experiments (see table 4.2) are roughly on par with those reported in the literature (chapter 2.4.2), or even slightly better. On average, the HunPos tagger falls slightly behind both RFTagger and MATE tools on morphological tagging, and especially when used for annotating vowel lengths. The former has the advantage of predicting each morphological attribute separately; the later benefits from the syntactic analysis that is performed together with the tagging. In light of this, it seems concievable that a combination of these two features would prove beneficial to the VLA score.

Between LDT and PROIEL, no clear winner could be determined; while PROIEL supposedly has an advantage due to its larger size, it was found that training on LDT instead gave a better result on a text by Tacitus. While not proven statistically significant, the opposite seems to be true for Cornelius Nepos. It might be that we here see the effect of different domains (cf. Skjærholt (2011a) and Poudat and Longrée (2009)): while neither author is represented in any of the corpora, Nepos is commonly regarded as an easy author, and it may be that his language in many respects is similar to that of Caesar and the *Vulgate*, which make up a large part of PROIEL. The more eclectic composition of LDT may on the other hand make the system better prepared to tackle Tacitus, who is known as a more difficult author, with terse and varied syntax.

The optimization experiments revealed some cases where the default tagger parameters are not optimal for Latin. First, increasing the rare word frequency cutoff in HunPos proved beneficial in most cases (figure 4.1). This confirms the suspicion of Skjærholt (2011a, p. 51) that the frequency cutoff should be modified, seeing that an unknown word may just as well be an unseen inflectional form of a common word, rather than a genuinely rare lemma.

The default context windows size in RFTagger is by default very conservative. This parameter (c) can with advantage be increased to 3 or 4 or even more (cf. figure 4.4), as also confirmed by Schmid and Laws (2008) in their own experiments on German.

The experiments also revealed some interesting differences between the intrinsic and extrinsic evaluation. Particularly when using RFTagger (see figure 4.4), it was found that VLA_t benefited more from larger context windows than does TA. With HunPos, this tendency could not be as clearly demonstrated,

though the small decrease in tagging accuracy on LDT, together with the concurring increase in VLA_t as the emission context increased (figure 4.2) may hint at a similar phenomenon.

One hypothesis for why this may be the case is that a larger context window may improve inflectional agreement between separated words, and that the decrease in tag accuracy (supposedly resulting from the increased sparseness) mainly concerns tags (or rather individual morphological attributes) that are inconsequential to the choice of different vowel lengths. An analysis of the differences would have to be carried out to answer that conclusively.

A number of ways can be envisioned to improve the performance. The idea was briefly entertained to merge the two corpora into one, and thus get a considerable increase in the available training data. This poses a challenge due to their various peculiarities and different tag sets, but if done carefully it may prove beneficial. At the same time, it must be remembered that the tagging accuracy is already quite good, and that a consolidation of powers thus may lead to only a marginal improvement.

The lemmatizer which has been used is quite basic, but, from the very limited investigation into the matter, it seems that this is not an overwhelming problem, as even dedicated tools have problems exactly where reliable lemmatization is most needed: when the lexicon provides several lemma alternatives to a word form and tag pair. Some not so uncommon homographs make for a real challenge to a lemmatizer, such as the pair *lĕvis* 'light' (adj.) and *lēvis* 'smooth' mentioned in the introduction; in cases such as these, a more advanced semantic analysis is probably needed.

However, the single module that, if improved, would have the most dramatic effect on the performance is probably the Morpheus lexicon. As shown in figure 4.6, there is a considerable gap between the accuracy of the oracle system and a perfect result. With a full coverage of the classical vocabulary (a grand vision, to be sure, but not impossible), the vowel length accuracy on classical texts may reach 99%, or thereabouts. In lieu of this, an improvement of the vowel lengths in unknown words could be gained by attempting to mark vowels in suffixes, based on the predicted tag and known inflectional endings.

Bibliography

Allen, W.S. (1989). *Vox Latina*. Second edition. Cambridge University Press. ISBN: 9780521379366.

Bamman, David and Gregory Crane (2008). "Building a dynamic lexicon from a digital library". In: *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*. ACM, pp. 11–20.

Bamman, David and Gregory Crane (2011). "The ancient Greek and Latin dependency treebanks". In: *Language Technology for Cultural Heritage*. Springer, pp. 79–98.

Bamman, David, Marco Passarotti, Gregory Crane, and Savina Raynaud (2007). *Guidelines for the Syntactic Annotation of Latin Treebanks (v. 1.3)*.

Bohnet, Bernd and Joakim Nivre (2012). "A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing". In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pp. 1455–1465.

Bohnet, Bernd, Joakim Nivre, Igor Boguslavsky, Richárd Farkas, Filip Ginter, and Jan Hajič (2013). "Joint morphological and syntactic analysis for richly inflected languages". *Transactions of the Association for Computational Linguistics* 1, pp. 415–428.

Brants, Thorsten (2000). "TnT: a statistical part-of-speech tagger". In: *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, pp. 224–231.

Carroll, L. (2011). *Alicia in Terrā Mīrābilī*. Ed. by Johan Winge. Evertype. ISBN: 9781904808695.

Crane, Gregory (1991). "Generating and parsing classical Greek". *Literary and Linguistic Computing* 6.4, pp. 243–245.

Dietterich, Thomas G (1998). "Approximate statistical tests for comparing supervised classification learning algorithms". *Neural computation* 10.7, pp. 1895–1923.

Halácsy, Péter, András Kornai, and Csaba Oravecz (2007). "HunPos: an open source trigram tagger". In: *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*. Association for Computational Linguistics, pp. 209–212.

Haug, Dag TT and Marius Jøhndal (2008). "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*, pp. 27–34.

Lee, John, Jason Naradowsky, and David A Smith (2011). "A discriminative model for joint morphological disambiguation and dependency parsing".

In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Volume 1*. Association for Computational Linguistics, pp. 885–894.

Lewis, Charlton Thomas and Charles Short (1879). *A Latin Dictionary Founded on Andrews' Edition of Freund's Latin Dictionary. Revised, Enlarged and in Great Part Rewritten by CT Lewis and Charles Short*. Clarendon Press.

McGillivray, Barbara (2013). *Methods in Latin Computational Linguistics*. Brill.

Mulligan, Bret (2013). *Nepos: Life of Hannibal*. URL: `http://dcc.dickinson.edu/nepos-hannibal/preface` (visited on 2015-05-12).

Nivre, Joakim (2009). "Non-projective dependency parsing in expected linear time". In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1*. Association for Computational Linguistics, pp. 351–359.

Orosz, György and Attila Novák (2013). "PurePos 2.0: a hybrid tool for morphological disambiguation." In: *RANLP*, pp. 539–545.

Passarotti, Marco (2010). "Leaving behind the less-resourced status. The case of Latin through the experience of the Index Thomisticus treebank". In: *Proceedings of the 7th SaLTMiL Workshop on the creation and use of basic lexical resources for less-resourced languages, LREC 2010, La Valletta, Malta*.

Piotrowski, Michael (2012). "Natural Language Processing for Historical Texts". *Synthesis Lectures on Human Language Technologies* 5.2, pp. 1–157.

Poudat, Céline and Doninique Longrée (2009). "Variations langagières et annotation morphosyntaxique du latin classique". *Traitement Automatique des Langues* 50.2, pp. 129–148.

Ritchie, Francis (1903). *Ritchie's fabulae faciles: a first Latin reader*. Ed. by J.C. Kirtland. Longmans, Green, and Co. URL: `http://www.gutenberg.org/ebooks/8997` (visited on 2015-05-12).

Schmid, Helmut and Florian Laws (2008). "Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging". In: *Proceedings of the 22nd International Conference on Computational Linguistics: Volume 1*. Association for Computational Linguistics, pp. 777–784.

Skjærholt, Arne (2011a). "Ars flectandi". MA thesis. Norway: University of Oslo.

Skjærholt, Arne (2011b). "More, Faster: Accelerated Corpus Annotation with Statistical Taggers." *JLCL* 26.2, pp. 151–163.

Winge, Johan (2008). *Tacitus: Annales book I*. URL: `http://web.comhem.se/alatius/latin/tacann01.html` (visited on 2015-05-12).

A Tag sets

As a preparation for the development of a tag conversion script, it was necessary to gain familiarity with the two treebanks and the composition of their morphological tags. The tags of LDT are presented in tables A.1 and A.2, and tables A.3 and A.4 give the corresponding data for the PROIEL corpus.

Table A.1: Tag set of the Latin Dependency Treebank, excluding verbs. Tags occurring only once are not included. The asterisk * stands for any possible value from the corresponding field of the PoS tag, as given in table 2.1.

Part of speech	PoS-tag									Freq.
	PoS	P	N	T	M	V	G	C	D	
undefined	-	-	-	-	-	-	-	-	-	231
	-	-	*	-	-	-	*	*	-	4
adjective	a	-	-	-	-	-	-	-	-	4
	a	-	*	-	-	-	-	*	-	11
	a	-	*	-	-	-	-	*	*	5
	a	-	*	-	-	-	*	*	-	5 342
	a	-	*	-	-	-	*	*	*	256
conjunction	c	-	-	-	-	-	-	-	-	5 648
adverb	d	-	-	-	-	-	-	-	-	4 056
	d	-	-	-	-	-	-	-	*	4
exclamation	e	-	-	-	-	-	-	-	-	116
interjection	i	-	-	-	-	-	-	-	-	1
numeral	m	-	-	-	-	-	-	-	-	272
noun	n	-	-	-	-	-	-	-	-	10
	n	-	-	-	-	-	*	-	-	2
	n	-	*	-	-	-	-	*	-	19
	n	-	*	-	-	-	*	-	-	32
	n	-	*	-	-	-	*	*	-	13 930
pronoun	p	-	-	-	-	-	-	-	-	8
	p	-	*	-	-	-	*	*	-	4 614
preposition	r	-	-	-	-	-	-	-	-	2 824
punctuation	u	-	-	-	-	-	-	-	-	4 561

Table A.2: Tag set of verbs in the Latin Dependency Treebank. Tags occurring only once are not included.

Part of speech	PoS tag									Freq.
	PoS	P	N	T	M	V	G	C	D	
	v	-	s	p	-	a	n	*	-	7
	v	-	s	p	d	a	n	*	-	17
	v	-	-	f	n	a	-	-	-	16
	v	-	-	p	n	a	-	-	-	1 003
	v	-	-	p	n	p	-	-	-	285
	v	-	-	r	n	a	-	-	-	104
	v	-	s	p	g	a	m	a	-	2
	v	-	*	p	g	p	*	*	-	138
	v	-	*	f	p	a	*	*	-	69
	v	-	*	p	p	a	*	*	-	533
	v	-	*	p	p	a	*	*	*	5
	v	-	*	r	p	a	*	*	-	17
verb	v	-	p	r	p	d	m	n	-	2
	v	-	*	r	p	p	*	*	-	1 569
	v	*	*	-	i/s	a	-	-	-	10
	v	*	*	p	i/s	*	-	-	-	3 527
	v	*	*	p	m	*	-	-	-	258
	v	*	*	i	-	p	-	-	-	2
	v	*	*	i	i/s	*	-	-	-	1 148
	v	*	*	f	i	*	-	-	-	422
	v	*	*	f	m	a	-	-	-	10
	v	*	*	r	-	a	-	-	-	3
	v	*	*	r	i/s	a	-	-	-	1 646
	v	*	*	l	i/s	a	-	-	-	338
	v	*	*	t	i	a	-	-	-	38

Table A.3: Tag set of PROIEL, excluding verbs. The asterisk * stands for any possible value from the corresponding field of the MSD, as given in table 2.2.

PoS meaning	PoS	Morpho-syntactic descriptor										Freq.
		P	N	T	M	V	G	C	D	-	I	
adjective	A-	-	-	-	-	-	-	-	-	-	n	14
	A-	-	*	-	-	-	*	*	*	-	i	7 011
conjunction	C-	-	-	-	-	-	-	-	-	-	n	11 057
adverb	Df	-	-	-	-	-	-	-	-	-	n	12 244
	Df	-	-	-	-	-	-	-	*	-	i	1 554
relative adverb	Dq	-	-	-	-	-	-	-	-	-	n	698
interrogative adverb	Du	-	-	-	-	-	-	-	-	-	n	523
foreign word	F-	-	-	-	-	-	-	-	-	-	n	459
subjunction	G-	-	-	-	-	-	-	-	-	-	n	4 932
interjection	I-	-	-	-	-	-	-	-	-	-	n	449
cardinal numeral	Ma	-	-	-	-	-	-	-	-	-	n	480
	Ma	-	*	-	-	-	*	*	-	-	i	747
ordinal numeral	Mo	-	*	-	-	-	*	*	-	-	i	365
common noun	Nb	-	-	-	-	-	-	-	-	-	n	55
	Nb	-	*	-	-	-	*	*	-	-	i	27 876
proper noun	Ne	-	-	-	-	-	-	-	-	-	n	568
	Ne	-	*	-	-	-	*	*	-	-	i	5 219
reciprocal pronoun	Pc	-	*	-	-	-	*	*	-	-	i	4
demonstrative pronoun	Pd	-	*	-	-	-	*	*	-	-	i	4 795
interrogative pronoun	Pi	-	*	-	-	-	*	*	-	-	i	900
personal reflexive pron.	Pk	*	*	-	-	-	*	*	-	-	i	799
personal pronoun	Pp	*	*	-	-	-	*	*	-	-	i	8 757
relative pronoun	Pr	-	*	-	-	-	*	*	-	-	i	4 056
possessive pronoun	Ps	*	*	-	-	-	*	*	-	-	i	2 024
possessive reflexive pron.	Pt	*	*	-	-	-	*	*	-	-	i	828
indefinite pronoun	Px	-	-	-	-	-	-	-	-	-	n	11
	Px	-	*	-	-	-	*	*	-	-	i	2 822
preposition	R-	-	-	-	-	-	-	-	-	-	n	11 250

Table A.4: Tags for verbs in PROIEL.

PoS meaning	PoS	Morpho-syntactic descriptor										Freq.
		P	N	T	M	V	G	C	D	-	I	
	V-	-	-	-	-	-	-	-	-	-	n	1
	V-	-	-	-	d	-	-	*	-	-	i	181
	V-	-	-	-	u	-	-	*	-	-	i	11
	V-	-	-	p	n	*	-	-	-	-	i	3 463
	V-	-	-	r	n	a	-	-	-	-	i	310
	V-	-	*	-	g	-	*	*	-	-	i	416
	V-	-	*	p	p	a	*	*	-	-	i	2 298
	V-	-	*	f	p	a	*	*	-	-	i	310
	V-	-	*	f	p	p	*	*	-	-	i	1
	V-	-	*	r	p	a	*	*	-	-	i	13
verb	V-	-	*	r	p	p	*	*	-	-	i	3 687
	V-	*	*	p	i/s	*	-	-	-	-	i	10 858
	V-	*	*	p	m	*	-	-	-	-	i	1 160
	V-	*	*	i	i/s	*	-	-	-	-	i	4 395
	V-	*	*	f	i	*	-	-	-	-	i	1 793
	V-	*	*	f	m	a	-	-	-	-	i	61
	V-	*	*	r	i/s	a	-	-	-	-	i	6 165
	V-	*	*	r	i	p	-	-	-	-	i	1
	V-	*	*	l	i/s	a	-	-	-	-	i	1 242
	V-	*	*	t	i	a	-	-	-	-	i	466
	V-	*	*	t	s	a	-	-	-	-	i	2