BRILL

RESEARCH DATA JOURNAL FOR THE HUMANITIES AND SOCIAL SCIENCES (2018) 1-11

BRILL OPEN

brill.com/rdj

The Diorisis Ancient Greek Corpus

Linguistics and Literature

A. Vatri
The Alan Turing Institute & University of Oxford, UK
alessandro.vatri@wolfson.ox.ac.uk

B. McGillivray
The Alan Turing Institute & University of Cambridge, UK
bm517@cam.ac.uk

Abstract

The Diorisis Ancient Greek Corpus is a digital collection of ancient Greek texts (from Homer to the early fifth century AD) compiled for linguistic analyses, and specifically with the purpose of developing a computational model of semantic change in Ancient Greek. The corpus consists of 820 texts sourced from open access digital libraries. The texts have been automatically enriched with morphological information for each word. The automatic assignment of words to the correct dictionary entry (lemmatization) has been disambiguated with the implementation of a part-of-speech tagger (a computer programme that may select the part of speech to which an ambiguous word belongs).

Keywords

digital corpus – Greek – lemmatization – annotated corpus – classics – historical linguistics

– Related data set "Diorisis Ancient Greek Corpus" with DOI https://www.doi.org/10.6084/m9.figshare.6187256 in repository "figshare".

1. Introduction

The Diorisis Ancient Greek Corpus was created in the context of the project "Computational models of meaning change in natural language texts" (SF042) funded by The Alan Turing Institute. The project aimed at developing Bayesian learning models of semantic change in Ancient Greek texts and therefore required a large diachronic corpus of Ancient Greek as a basis for the statistical modelling. In this article we describe the main features of the Diorisis Ancient Greek Corpus and how it was designed and created. The corpus, aimed at classics and historical linguistics scholars, is the largest of its kind and can be used as an evidence basis for a wide range of studies on the Ancient Greek language.

2. Context

The computational model we have been developing for the purpose of our project requires that the texts in the corpus be input as text files in which each sentence is stored in one and only one line. Each line should begin with the year in which the text was composed, and this should be separated from the sentence by a tab. Sentences should appear as sequences of lemmas; that is, all the inflected forms should be converted into the corresponding dictionary entry. High-frequency words (such as forms of the verb 'to be' or function words such as 'the') need to be filtered out, as they do not provide useful information to the model and only generate noise in the data. The preparation of input in this form requires the texts to be annotated with information on the dictionary entry of each word-form (lemmatization). Available collections of lemmatized Ancient Greek (AG) texts are very small in size and number: the Ancient Greek Dependency Treebank created and maintained by the Perseus Project (Bamman and Crane, 2011, http://perseusdl.github.io/treebank_data) only contains thirty-three texts (557,922 word-tokens, including punctuation marks), and the annotated Greek texts included in the PROIEL treebank (Haug and Jøhndal, 2008, https://proiel.github.io) only include Herodotus' *Histories* and the New Testament (225,837 word-tokens). These resources are too limited for the purposes of our project and, furthermore, lack the date and text-type (i.e. literary genre) metadata that our model needs to take into account.

For these reasons, we have complied a large AG corpus from open-access sources, lemmatized it automatically, and manually added metadata. One of the challenges of this task stems from the fact that we sourced the data from resources in different digital formats:

- Text Encoding Initiative (TEI) XML (with or without namespace specification)
- Non-TEI XML
- HTML
- Microsoft Word files

TABLE 1 *Possible encodings of the Greek character ᾆ*

Greek Character	Unicode (UTF - 8)	Unicode (hex reference)	Beta Code
ᾆ	ᾆ	&#1F86;	A)=|

Greek characters were originally encoded either as Beta Code (https://www.tlg.uci.edu/encoding) or as UTF-8 Unicode. In certain HTML pages, UTF-8 characters were encoded as HTML hexadecimal references (see for instance Table 1).

All these discrepancies needed to be brought to uniformity.

3. Methods

3.1. *Selection of Texts*

We designed the Diorisis corpus in order for it to be representative of a fair number of Ancient Greek genres (see section 4. Data below). We decided not to include anthological collections of texts from different periods, such as the Greek Anthology; however, we did include texts that contain a large number of quotations, such as Athenaeus' *Deipnosophists* (second century AD) and the rhetorical works of Dionysius of Halicarnassus (first century BC). Texts were sourced from:

(1) the Perseus Canonical Greek Literature repository (752 texts, XML format, licensed under a Creative Commons Attribution-ShareAlike 3.0 United States License, https://www.github.com/PerseusDL/canonical-greekLit);
(2) "The Little Sailing" digital library (8 texts, Microsoft Word, http://www.mikrosapoplous.gr/en/texts1en.html);
(3) the Bibliotheca Augustana digital library (60 texts, HTML format, http://www.hs-augsburg.de/~harsch/augustana.html#gr).

3.2. *Metadata*

All texts have been converted into TEI-compliant XML. The TEI headers of Perseus source files have been included in the destination files (in the element fileDesc/sourceDesc/biblFull). The following metadata have been added to all texts:

(1) the approximate or exact (when known) date of composition of each text, sourced from the most up-to-date literature on each AG author or work (stored in the element profileDesc/creation in the TEI header);
(2) the text-type (literary genre and sub-genre) of each text (stored in the elements xenoData/genre and xenoData/subgenre);

(3) a reference to the URL of the source files (in the element fileDesc/sourceDesc/ref);
(4) the identificators of AG authors and works from the TLG canon (http://stephanus.tlg.uci.edu/canon.php), which are adopted as a standard by the Perseus Project as well (stored in fileDesc/titleStmt/tlgAuthor and fileDesc/titleStmt/tlgId);
(5) the names and roles of the persons involved in the preparation of the corpus (fileDesc/editionStmt/respStmt element and subelements).

All materials not belonging to the text body (footnotes, critical apparatuses, other annotations) have been removed during the conversion of source files, with the exception of the following information:

– the location of each sentence in the text (line or book/chapter/section numbers), when available, has been preserved and stored as an attribute of each sentence node (see Data section below);
– if a sequence was marked as a quotation (through the tag <quote> in the Perseus XML files), words extracted from such sequence contain the attribute @isquote with the value 'True');
– if a word of part of a word was supplied by a modern editor (in fragmentary texts; element <add> in the Perseus XML files), words consisting of, or containing, such additions are marked with the attribute @lacuna with the value 'True'.

3.3. *Character Encoding*

All Greek characters have been converted to Beta Code, in order to adopt a uniform and consistent encoding and with a view to automatic parsing and lemmatization. For these purposes, Beta Code was chosen because of its flexibility and ease of use in the following look-up operations:

– Word-forms to be automatically analysed and annotated may or may not start with a capital letter; in order to be matched to entries in a digital dictionary, forms should be converted to the formats corresponding to the entries. Greek lowercase and uppercase letters are encoded as different characters in the Unicode table (e.g. the lower-case letter α corresponds to UTF-8 code 0391, the upper-case letter A corresponds to UTF-8 code 03B1), which would require an ad-hoc conversion for each character between its lower-case and upper-case versions. Beta Code simply encodes capitalization through the juxtaposition of an asterisk (*) character (lower-case α is

encoded as A, and upper-case A is encoded as *A), which can be easily added or removed in the look-up process.
- Diacritics such as the Greek diaeresis (¨) may or may not appear in dictionary entries (for instance, editors may add them to Greek words to mark hiatuses in metrical texts). Greek characters containing the diaeresis (alone or in combination with other diacritic marks) all have different UTF-8 codes (e.g. ϊ = 03CA, ΐ = 0390, ῒ = 1FD2, ῗ = 1FD7), whereas Beta Code encodes the diaeresis through the juxtaposition of a plus sign (+; e.g. ϊ = I+, ΐ = I/+, ῒ = I\+, ῗ = I=+). This makes it very easy to process diacritics in the look-up process.
- In AG orthography, the grave accent (`) is only used to mark the alteration of the pitch normally marked by an acute accent in connected speech; thus, it never appears in dictionary entries (which only contain acute or circumflex accents). Whereas Unicode has different codes for Greek characters with an acute or a grave accent, Beta Code encodes such diacritics as forward (/) and backward (\) slashes, respectively; this makes grave accents easy to convert into acute accents in the look-up process.

Different characters are used as quotation marks in the source files: single straight quotes ('), single curly quotes (‘’), double straight quotes ("), double curly quotes (“”), angle quotes («»). These have all been converted to double straight quotes, with the exception of single straight/curly quotes, which may be used as apostrophes (marking prodelision at word beginning and elision at word end). Single curly quotes used as apostrophes have been converted to straight quotes.

3.4. *Linguistic Pre-processing*

We have conducted a series of automatic linguistic pre-processing steps on all text files in the corpus via Python scripts (published on https://www.github.com/alevatri/diorisis). We performed sentence segmentation based on strong punctuation marks, i.e. Greek full stop (.) middle dot (·), and question mark (;). We performed word tokenization based on white spaces. Words divided (and hyphenated) at line ends have been joined into a single word node. Punctuation marks have been tokenized and assigned to special nodes (see section 4. Data below). The tokenized files are available from https://www.figshare.com/articles/Diorisis_Corpus_-_Preprocessed_files/7229162.

Lemmatization has been performed using a dictionary based on the parsed word-form list included in Diogenes (https://community.dur.ac.uk/p.j.heslin/Software/Diogenes/), a tool for searching AG and Latin corpora distributed under the GNU General Public License. The path to the original list within the software package is /Resources/perl/Perseus_Data/greek-analyses.txt. The list was provided by the Perseus Digital Library under Creative Commons licensing and contains all possible morphological analyses for 911,840 AG word forms.

One important step was handling ambiguous forms. In the Diogenes list, 364,028 word forms admit more than one analysis; 93,248 of them may be parsed as forms of different lemmas (see below for an example). Assigning the correct lemma to a word form in its context is crucial for the purpose of our project and is also required in a number of linguistic analyses. The dictionary was able to recognize and provide possible analyses for all except 152,274 words in our corpus (1.49%, see section 4. Data below on the size of the corpus). Word tokens that may be analyzed as forms of different lemmas amount to 2,020,004 (19.79%). One approach for selecting a single lemma in such cases would consist in picking the first (or an otherwise random) possible parse from the dictionary. The Classical Language Toolkit (CLTK) lemmatizer (http://docs.cltk.org/en/latest/greek.html#lemmatization) selects lemmas based on their overall frequency in Greek. Our approach consists in assigning a part-of-speech (PoS) to each form in the texts and then we assign the lemma based on the PoS. This allows to disambiguate those forms that correspond to different lemmas with different PoS values. For instance, an AG word like πράξεις admits the following analyses:

- lemma: πράσσω; POS: verb; morphology: second person singular, active future indicative;
- lemma: πρᾶξις; POS: noun; morphology: nominative or accusative plural.

A PoS tagger would output whether the word-form πράξεις should be interpreted as a noun or as a verb in context, which would entail that we may select the lemma πρᾶξις or the lemma πράσσω as its headword. The effectiveness of this approach is limited by the fact that certain words may be analysed as forms of lemmas belonging to the same headword. For instance, the form βασιλειῶν is either:

- the genitive plural of βασίλεια (noun, 'queen'), or
- the genitive plural of βασιλεία (noun, 'kingdom');
- or the masculine or neuter nominative singular of the present participle of the verb βασιλειάω.

In such cases, the 'verb' output of a PoS tagger corresponds to only one candidate. Conversely, if a PoS tagger ouputs 'noun', two candidate lemmas will be selected, and one of them should still be picked randomly with a confidence score for the disambiguation corresponding to the inverse of the number of possible candidate lemmas (e.g., one of two nouns would be selected with 0.5 confidence).

The PoS tagger we have trained and used for this purpose is TreeTagger (http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger, Schmid, 1994 and 1995). This tool was trained on annotated AG texts available from the Perseus Ancient Greek and Latin Dependency Treebank (http://perseusdl.github.io/treebank_data/) and from the PROIEL project (https://proiel.github.io). 7 out of 33 texts available from the Perseus treebank were excluded from the training set and used as a test set. The accuracy score of this TreeTagger model (calculated as the number of correct PoS tags out of all assigned PoS tags) was found to amount to 91% (see Celano et al., 2016 for a comparison with the performance of other PoS taggers).

Running TreeTagger on the whole corpus gave the following results:

- 1,130,786 word tokens were disambiguated in an unequivocal way (i.e. TreeTagger output a PoS corresponding to one and only one lemma);
- residual ambiguity was cut down to 8.71% of the word tokens;
- confidence scores of words for which TreeTagger output PoS tags corresponding to multiple lemmas sum up to 45,446.27. If this figure is summed to the 1,130,786 unequivocally-disambiguated word tokens, the total disambiguation score would amount to 1,176,232.27 out of 2,020,004 words, and the residual ambiguity is further reduced to 8.26%.

4. Data

- **Diorisis Ancient Greek Corpus deposited at figshare** – **DOI:**10.6084/m9.figshare.6187256
- **Temporal coverage:** ca 7th century BC - 5th century AD

The corpus consists of 820 texts spanning between the beginnings of the AG literary tradition (Homer) and the fifth century AD, and it counts 10,206,421 word tokens. Each work is stored in a separate XML file; filenames have the following structure: *author name* (*TLG Author ID*) - *work title* (*TLG Work ID*).

The corpus includes samples from a number of genres and subgenres. These have been encoded as metadata in the XML TEI header (see Method section above) as detailed in Table 2.

TABLE 2 *Genres and subgenres included in the Diorisis Corpus*

Genres	Subgenres
Poetry	Bucolic
	Didactic
	Epic
	Epigrams
	Erotic
	Choral
Comedy	Comedy
Tragedy	Tragedy
Philosophy	Philosophy
Essays	Essay
	e.g. works of Plutarch and Lucian
	Miscellanea
	The Varia Historia of Aelian and Athenaeus' *Deipnosophists*
Letters	Letters
Narrative	Biography
	Novel
	Mythology
	Church history
	History
Oratory	Oratory
Religion	Homily
	Hymns
	Pagan hymns
	Narrative
	Septuaginta and New Testament
	Protreptics
	Christian protreptics
	Psalms
	Theology
	Christian theology
Technical	Art history
	Geography
	Grammar
	Horsemanship
	Hunting
	Mathematics
	Medicine
	Military
	Natural history
	Politics
	Rhetoric, poetics, criticism
	Science
	Aristotelian treatises

The number of words per genre per century is displayed in Table 3.

TABLE 3 *Word counts of texts in the Diorisis corpus per genre and century*

	Century										
Genres	7BC	5BC	4BC	3BC	2BC	1BC	1AD	2AD	3AD	4AD	5AD	**Total**
Comedy		78,949	15,805									94,754
Essays			3,827				475,169	263,270	361,213	23,965		1,127,444
Letters			9,566	1,333					9,654	164,388		184,941
Narrative		334,791	208,921	311,307		661,459	961,707	482,568	411,061	98,382		3,470,196
Oratory		57,542	529,374				184,783	295,583	2,855	55,683		1,125,820
Philosophy			895,412					112,846	213,493			1,221,751
Poetry	215,075	21,120		80,783	3,158	7,341		22,752	17,714	60,107	126,892	554,942
Religion	15,788			131,895	463,115		133,864	44,919		17,884		807,465
Technical		104,091	326,746	15,409		385,503	24,056	394,012	157,947	3,886		1,411,650
Tragedy		207,458										207,458
Total	**230,863**	**803,951**	**1,989,651**	**540,727**	**466,273**	**1,054,303**	**1,779,579**	**1,615,950**	**1,173,937**	**424,295**	**126,892**	**10,206,421**

The XML files are structured as follows:

```
<TEI.2>
  <teiHeader />
  <text>
    <body>
      <sentence id = "n" location ="N">
        <word form = "form" id = "n" lacuna = "True" isquote = "True">
          <lemma id = "id" entry = "entry" POS = "POS" TreeTagger="true/
          false" disambiguated="n">
            <analysis morph = "morph" />
          </lemma>
        </word>
        <punct mark = "mark" />
      </sentence>
    </body>
  </text>
</TEI.2>
```

<teiHeader> see section 3, Methods, above.

<sentence> nodes have the following attributes:
- @id: progressive integer uniquely identifying the sentence in the file;
- @location: location of the sentence in the text (line, book/chapter/section, etc.), if available.

<word> nodes have the following attributes:
- @form: word-form as appears in the text, in Beta Code;
- @id: progressive integer uniquely identifying the word in the sentence;
- @lacuna, @isquote (optional attributes): see section 3. Method above.

<lemma> nodes are children of <word> nodes and contain the lemmatization information for each word. They have the following attributes:
- @id: unique alphanumeric identifier of each lemma in the dictionary;
- @entry: human-readable dictionary entry in UTF-8 format;
- @POS: part-of-speech;
- @TreeTagger: this attribute specifies whether the word-form was disambiguated using TreeTagger (see Method section above); possible values are 'true' or 'false';
- @disambiguated: if the @TreeTagger attribute is set to 'true', this attribute indicates the degree of confidence n in the disambiguation ($0 < n \leq 1$; see Method section above). If @TreeTagger is set to 'false', the value of this attribute is 'n/a'.

<analysis> nodes are children of <lemma> nodes and contain all possible morphological analyses of the word-form. Each <lemma> node may contain multiple <analysis> nodes. These have the following attribute:
- @morph: morphological features of the lemma extracted from the Diogenes word-form list without further processing. The attribute may contain combinations of the values listed in Table 4.

TABLE 4 *Values of @morph attribute (morphological features), explanations in italics*

case	
acc	*accusative*
dat	*dative*
gen	*genitive*
nom	*nominative*
voc	*vocative*

person
1st
2nd
3rd

number	
dual	*dual*
pl	*plural*
sg	*singular*

gender	
fem	*feminine*
masc	*masculine*
neut	*neuter*

degree	
comp	*comparative*
irreg_comp	*irregular com.*
irreg_superl	*irregular sup.*
superl	*superlative*

tense	
aor	*aorist*
fut	*future*
futperf	*future perfect*
imperf	*imperfect*
perf	*perfect*
plup	*pluperfect*
pres	*present*

mood	
imperat	*imperative*
ind	*indicative*
inf	*infinitive*
opt	*optative*
part	*participle*
subj	*subjunctive*

voice	
act	*active*
mid	*middle*
mp	*medio - passive*
pass	*passive*

word class	
adverb	*adverb*
adverbial	*adverbial*
conj	*conjunction*
exclam	*exclamation*
expletive	*expletive*
geog_name	*geographical name*
interrog	*interrogative*
numeral	*numeral*
particle	*particle*
prep	*preposition*

dialect/register	
aeolic	*Aeolic*
alphabetic	*alphabetic Greek*
attic	*Attic*
doric	*Doric*
epic	*epic*
homeric	*Homeric*
ionic	*Ionic*
poetic	*poetic*
prose	*prose*

other	
a_priv	*alpha privativum*
contr	*contracted form*
enclitic	*enclitic*
indec	*indeclinable*
indeclform	*indeclinable form*
iota_intens	*iota intensivum (deictic)*
nu_movable	*nu movable*
parad_form	*paradigm form (only attested in examples from grammarians)*
proclitic	*proclitic*

<punct> nodes encode punctuation marks and have the following attribute:
- @mark: the punctuation mark, in Beta Code.

Acknowledgments

This work was supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and the seed funding grant SF042.

References

Bamman, D., & Crane, G. (2011). The Ancient Greek and Latin Dependency Treebanks. In C. Sporleder, A. van den Bosch, & K. Zervanou (Eds.), *Language Technology for Cultural Heritage. Theory and Applications of Natural Language Processing* (pp. 79–98). Berlin: Springer. doi:10.1007/978-3-642-20227-8_5.

Celano, G. G. A., Crane, G., & Majidi, S. (2016). Part of Speech Tagging for Ancient Greek. *Open Linguistics,* 2, 393–399. doi:10.1515/opli-2016-0020.

Haug, D. T. T., & Jøhndal, M. L. (2008). Creating a Parallel Treebank of the Old Indo-European Bible Translations. In C. Sporleder & K. Ribarov (Eds.), *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data* (*LaTeCH 2008*) (pp. 27–34).

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to German. In *Proceedings of the ACL SIGDAT-Workshop* (pp. 47–50).