DIGITAL CLASSICS ONLINE

Opera Graeca Adnotata: Building a 40M+ Token Multilayer Corpus for Ancient Greek

Giuseppe G. A. Celano

Abstract: In this article, the beta version 0.2.0 of *Opera Graeca Adnotata* (*OGA*), the largest open access multilayer corpus for Ancient Greek (AG), is presented. *OGA* consists of 1,999 literary works and 40M+ tokens sourced from the *canonical-greekLit*, *First1KGreek*, and *PatristicTextArchive* GitHub repositories, which together host AG texts ranging from approximately 900 BCE to 1400 CE. The texts have been enriched with nine annotation layers: (i) tokenization; (ii) sentence segmentation; (iii) lemmatization; (iv) morphology; (v) dependency structure; (vi) dependency function; (vii) IPA transcription; (viii) composition date; and (ix) CTS structure. The layers are described by highlighting the main technical and annotation-related issues encountered. The corpus is released in the standoff formats PAULA XML and its derivative LAULA XML and is queryable online through ANNIS.

1. Introduction[1]

Multilayer corpora contain a variety of annotations modelled as independent layers. Unlike corpora with inline annotations, multilayer corpora have the unique advantage of scalability, as a potentially infinite number of annotation layers can be added using a standoff approach, where layers are interconnected through references to base texts in a graph structure.[2] An example of an open access multilayer corpus for a modern language is the National Corpus of Polish,[3] which is encoded in a standoff format according to the P5 TEI formalism: Polish texts mostly sourced from newspapers and magazines are tokenized, sentence-segmented, and annotated for morphosyntax, named entities, and word sense disambiguation.

A number of historical language corpora have also been annotated with different layers of linguistic information, such as Coptic Scriptorium[4] and RIDGES Herbology,[5] which are both provided in standoff PAULA XML. RIDGES Herbology, for example, contains German herbal texts ranging from 1478 to 1870, annotated with three different transcription layers, namely a diplomatic one, which is the closest to the original text, and two normalization layers called 'clean' and 'norm', respectively: the former aims to address the issue of a few character variations of the diplomatic transcription, while the latter

1 Acknowledgements: This work has been supported by the German Research Foundation (DFG project number 408121292).

2 While Zeldes (2018) proposes a definition of 'multilayer' with reference to independent annotation types, I adopt a definition where independence simply refers to formally separate standoff annotation layers, regardless of how content-wise independent layers are.

3 Przepiórkowski et al. (2011).

4 Schroeder / Zeldes (2016).

5 Odebrecht et al. (2017).

offers a higher-level normalization according to the principles of modern German orthography. Besides morphosyntactic annotation, it is noteworthy that the standoff nature of the RIDGES corpus allows the addition of, for example, lexical layers, such as the one specifying a token's language (e.g., whether it is German or Latin) and the one containing a person's full name, both of which would be difficult to add to syntactic annotation inline.

In the present article, I describe the beta version 0.2.0 of *Opera Graeca Adnotata* (*OGA*),[6] a multilayer corpus for Ancient Greek (AG),[7] focusing on its design as well as the technical and annotation-related issues encountered while working with a large dataset consisting of 1,999 texts and 40,105,221 tokens, which is currently by far the largest open access annotated corpus for AG.[8]

The paper is organized as follows: in Section 2, related work is presented, while Section 3 introduces the standoff formalism of PAULA XML and its derivative LAULA XML. Section 4 and its subsections describe the original texts (Section 4.1) and the annotation layers: tokenization (Section 4.2); sentence segmentation (Section 4.3); morphosyntax and lemmatization (Section 4.4); IPA transcription (Section 4.5); composition date (Section 4.6); and Canonical Text Services (CTS) structure (Section 4.7). Section 5 concludes the article with a brief summary and final remarks.

2. Related Work

There are a few noteworthy corpora of literary AG works. *Thesaurus Linguae Graecae*[9] is the largest non-open access corpus for AG (110M+ words); its texts are accessible only via a query interface with limited functionality, primarily supporting word form and lemma search. The open access counterpart of *TLG* is represented by *Perseus Digital Library* (*PDL*)[10] and its derivative *Scaife Viewer* (*SV*),[11] whose collection of literary AG texts largely coincides with that of *OGA*: both websites aim to offer a reading environment for literary texts (with limited search capability). *PhiloLogic4*[12] and *Diogenes*[13]—which are based (also) on the texts of *PDL*/*SV*—are also open access resources designed as reading environments, with integration of morphological and lexical information. *eAQUA*[14] offers a number of tools to extract information from classical texts. A pioneering corpus for fragmentary authors is *Digital Fragmenta Historicorum Graecorum*:[15] it provides the texts of 636 fragmentary Greek historians, along with their translations and commentaries, all of which can be searched for word forms.

Among the ever-growing number of open access resources, treebanks, such as the *Ancient Greek Dependency Treebank* (*AGDT*),[16] are particularly worth mentioning, as they contain a variety of texts

6 A previous release, *OGA v0.1.0*, is available on Zenodo at https://doi.org/10.5281/zenodo.8158675 (last access 11.07.2025) and is described in the preprint Celano (2024a), on which the present paper is based.

7 Celano (2024b).

8 The data is available on Zenodo at https://doi.org/10.5281/zenodo.14206061 (last access 11.07.2025) and can be queried online at https://annis.varro.informatik.uni-leipzig.de (last access 11.07.2025). Note that this latter website hosts the latest version of the corpus, which is meant to change over time as new releases become available.

9 https://stephanus.tlg.uci.edu (last access 11.07.2025).

10 https://www.perseus.tufts.edu/hopper (last access 11.07.2025). The website is a legacy one, the work continuing on the *SV*.

11 https://scaife.perseus.org (last access 11.07.2025).

12 https://perseus.uchicago.edu (last access 11.07.2025).

13 https://d.iogen.es (last access 11.07.2025).

14 http://www.eaqua.net (last access 20.01.2026).

15 https://www.dfhg-project.org (last access 11.07.2025); Berti (2021).

16 See Celano (2019) for an overview of existing treebanks.

manually annotated for morphosyntax, which many other resources, including *OGA*, rely on. The *Diorisis corpus*[17] offers lemmatization and morphological analyses of 820 texts and 10M+ tokens.[18] More recently, the *GLAUx* project aims to provide a larger morphosyntactically and semantically annotated corpus, i.e., the *GLAUx* corpus:[19] the latest version[20] consists of 20M+ morphosyntactically annotated tokens.[21] The above-mentioned annotated corpora share the characteristic of being encoded in a project-specific format, which is meant to provide consumable, but hardly extensible, data.

3. The Formats: Paula XML and LAULA XML

PAULA XML (Potsdamer AUstauschformat Linguistischer Annotationen) is an established open access standoff format for linguistic annotation,[22] which was inspired by LAF (Linguistic Annotation Framework).[23]

In PAULA XML, a base text is directly or indirectly referenced by identifiers contained in annotation layers, each of which is stored in a separate file.[24] Altogether, the files form an acyclic graph. A base text embeds the transcription of an original text within a shallow XML structure, so that it can typically be referenced by at least one annotation layer, i.e., the tokenization layer, which identifies tokens by referencing character offsets. Each thus identified token is associated with an ID, which can in turn be referenced in other annotation layers.

17 https://doi.org/10.6084/m9.figshare.6187256.v1 (last access 11.07.2025). *Diorisis* does not seem to be a currently maintained resource.

18 Vatri / McGillivray (2018).

19 Keersmaekers (2021).

20 https://glaux.be/; https://github.com/alekkeersmaekers/glaux (last access 11.07.2025).

21 More recently, an attempt to annotate Greek Papyri is detailed in Keersmaekers / Van Hal (2024).

22 https://github.com/korpling/paula-xml (last access 11.07.2025).

23 https://www.iso.org/standard/37326.html (last access 11.07.2025).

24 Standoff annotation is typically performed by using separate files for each annotation layer, and indeed, this is the PAULA XML model. However, it is to be noted that standoff annotation is primarily defined by a referencing mechanism keeping markup data and text to markup separate, without this necessarily implying separation in different files. However, for scalability purposes, separate files are typically used.

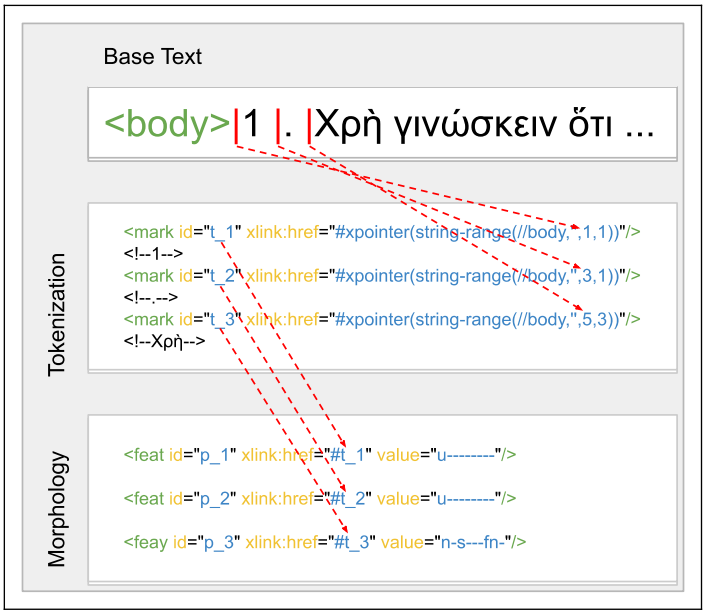

Fig. 1: Standoff annotation layers in PAULA XML.

As shown in fig. 1, the XPointer expressions within the tokenization layer reference the base text by specifying the start offset of a token – numbering starts with 1 and not 0 – and its length. The XPointer expressions are associated with IDs, which can then be used, for example, in the morphological layer to associate each of them with the corresponding morphological annotation contained in the `value` attribute.

Such a model has the advantage of offering an elegant solution to the issue of overlapping markup. For example, a layer for prosody annotation could reference tokens that differ from morphosyntactic tokens. Ensuring that different tokenization layers reference the same base text guarantees that their schemes can be compared.

Currently, *OGA* contains the following annotation layers: (i) tokenization; (ii) sentence segmentation;[25] (iii) lemmatization; (iv) morphology; (v) dependency structure; (vi) dependency function; (vii) IPA transcription; (viii) composition date; and (ix) CTS structure.

Notably, PAULA XML requires a base text to be inserted in the `<body>` element of an XML file, without any XML markup within it. PAULA XML is part of a set of technologies developed for annotation of multilayer corpora, which includes ANNIS,[26] a query engine, and its related technologies: Salt, a meta model for manipulating linguistic data, Pepper,[27] a converter between different annotation formats, and Hexatomic, an annotation editor.

One disadvantage of PAULA XML is that the size of a corpus tends to grow significantly as new texts and/or annotation layers are added. For example, the directory containing the PAULA XML files of *OGA 0.2.0* is as large as about 23GB (unzipped). For this reason, *OGA* files are processed using a lighter and more efficient XML structure, called LAULA XML (*Leipziger AUstauschformat Linguistischer Annotationen*), which retains the logic of PAULA XML, but its repeating element and attribute names are shortened to one character (e.g., `<mark>` becomes `<m>` and `<feature>` `<f>`); moreover, information that in PAULA XML can only be added inside XML comments is conveniently encoded, in LAULA XML, within XML attributes, whose contents can typically be processed by XPath parsers much more efficiently.

25 As explained below, this layer is available in LAULA XML, but not in PAULA XML.

26 Krause / Zeldes (2014); https://corpus-tools.org/annis (last access 11.07.2025).

27 https://corpus-tools.org/pepper (last access 11.07.2025).

More importantly, LAULA XML allows original TEI XML files to be directly referenced, without the need of modifying texts, a particularly convenient property for corpora such as *OGA*, whose main texts consist solely of TEI XML files, sometimes with heavy paratext markup. This means that, for example, if there is an interest in connecting tokens to the critical apparatus encoded in original files, LAULA XML—but not PAULA XML—supports this. A sentence segmentation layer is also provided in LAULA XML, but not in PAULA XML, in that the latter is used for conversion into relANNIS, the file format for ANNIS, which only allows token- and text-based queries.

4. The *OGA* Pipeline

Due to the high number of texts and annotation layers to process, *OGA* is created automatically with minimal human inspection: it is the output of several scripts, whose content is summarized in the following subsections. Because of its large size, the corpus, a few related resources, and its documentation are made available on Zenodo;[28] for the latest updates, the reader is referred to the associated GitHub repository.[29]

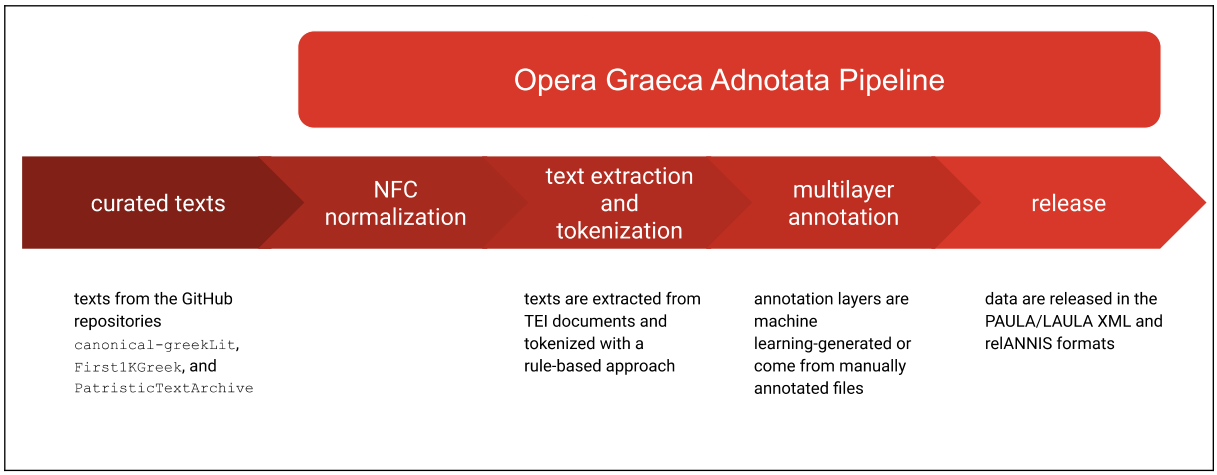

Fig. 2: Pipeline for the creation of *OGA*.

4.1. The Texts

The main texts of *OGA 0.2.0* are sourced from three independently managed GitHub repositories: (i) *canonical-greekLit*;[30] (ii) *First1KGreek*;[31] and (iii) *PatristicTextArchive*.[32] While the first repository contains Classical Greek literary texts, which mostly coincide with the ones available in PDL 4.0,[33] the second one aims to complement *canonical-greekLit* by adding one edition of any Greek literary work composed until about 250 CE. *PatristicTextArchive* is a more recent effort, which aims to create digital critical editions of patristic texts.

Since at least 2017, the texts in *canonical-greekLit* and *First1KGreek* have been edited actively for correction of OCR errors and, more in general, compliance with more recent standards. In particular, there has been an ongoing effort to transition older XML files into EpiDoc P5 TEI XML files and

28 https://doi.org/10.5281/zenodo.14206061 (last access 11.07.2025).

29 https://github.com/OperaGraecaAdnotata/OGA (last access 11.07.2025).

30 https://github.com/PerseusDL/canonical-greekLit (last access 11.07.2025).

31 https://github.com/OpenGreekAndLatin/First1KGreek (last access 11.07.2025).

32 https://github.com/PatristicTextArchive (last access 11.07.2025).

33 https://www.perseus.tufts.edu/hopper (last access 11.07.2025).

provide each text with CTS structure. New releases of both repositories are issued on a frequent basis. *PatristicTextArchive* encodes texts in TEI XML files with the specification of the same CTS structure found in *canonical-greekLit* and *First1KGreek*. Of the texts of these repositories, *OGA 0.2.0* contains 1,999, with a total of 40M+ tokens. *OGA* is conceived in such a way that, when new releases of the above-mentioned repositories become available, the corpus, along with its annotation layers, can be re-built from scratch, so as to incorporate newly added or modified texts.

It is to be noted that, to the best of my knowledge, an evaluation of the accuracy of the digitized texts (especially those in *canonical-greekLit* and *First1KGreek*) is still lacking.[34] While the digitized texts, in general, seem to represent the main texts of the original print editions accurately,[35] there still are various errors and inconsistencies that are going to affect tokenization and sentence segmentation (see section 4.2).

Although correction of the original texts is outside the scope of *OGA*, some automatic encoding normalization can be performed. More precisely, NFC normalization is applied to address the issue of characters such as 'epsilon with an acute accent', which can be encoded as the Unicode codepoint U+03AD or U+1F73, the former being in the "Greek and Coptic" chart, while the latter in the 'Greek Extended' one: only the codepoint U+03AD is, however, the NFC-normalized codepoint, and therefore NFC normalization ensures uniform encoding of this character.

4.2. Text Extraction and Tokenization

Since the original texts are encoded as TEI XML texts, preprocessing is needed to separate the text of a work, which annotation layers reference, from its paratext, which is not annotated.[36] This is a non-trivial task with heavily marked-up literary texts, because the distinction between (main) text and paratext is signalled in TEI documents via use of many different XML elements that serve different functions.

For example, the element `<note>`, as the name itself suggests, contains a note, and therefore its content can be safely identified as paratext. Similarly, the contents of elements such as `<app>` and `<bibl>` can be discarded, in that they unambiguously identify paratext related to critical apparatus and bibliography, respectively.

The content of other TEI elements is, however, part of a text: for example, `<foreign>` contains a foreign language term, while `<add>` signals a text addition by an editor, which should arguably be considered as part of the main text.

In a few cases, the semantics, and consequently the structure, of a TEI element are complex. For example, `<choice>` presents a number of alternative readings for a specific passage: it can contain `<sic>` to highlight a certain word form whose correction is given in its sibling element `<corr>`; or it can contain the sibling elements `<abbr>` and `<expan>` for an abbreviation and its expansion, respectively.

After a main text has been extracted, a rule-based tokenization is applied to it, as there is almost a one-to-one correspondence between graphic words and morphosyntactic tokens in AG. The most notable exception to this occurs in the case of a few conjunctions,[37] such as οὐδέ, and crasis, i.e., the phonological phenomenon whereby two words can be univerbated: for example, the word κἀκεῖνος consists of

34 An attempt to evaluate OCR for Ancient Greek is documented in Boschetti et al. (2009).

35 However, accuracy seems to greatly vary from text to text.

36 Currently, a way to keep a reference to paratext in PAULA XML would be to convert paratext TEI elements into annotation layers, as in the GUM corpus (see Zeldes [2017]). However, this solution turns out to be cumbersome, especially for heavily marked-up files. This issue does not arise in LAULA XML, however, in that the tokenization layer references an original file.

the words καὶ (‘and’) and ἐκεῖνος (‘that’): since they belong to different POS, they need to be separated in a morphosyntactic tokenization scheme.

Luckily, most cases of crasis can be unambiguously identified because of the punctuation mark ‘coronis’ (i.e., the Unicode codepoint COMBINING COMMA ABOVE, U+0313) placed on a non-initial vowel. On the basis of this formal criterion, the texts were searched for words with a coronis and a list of them, which is also made available in the *OGA* release, was compiled and used for tokenization. Cases where a coronis is more difficult to identify, as when a smooth breathing is on a word-initial vowel, were left untreated.

On a preliminary test based on 38,710 tokens from 2,234 sentences randomly selected from all texts in *OGA*, 281 erroneous tokens were found (i.e., error rate of about 0.0073): however, all tokenization errors except one[38] were due to OCR/encoding errors in the original texts: for example, paratext content is sometimes encoded as text or tokens contain wrong characters or missing accents.

Notably, XPointer expressions in a PAULA XML tokenization layer reference a base text that only consists of a long string contained in a `<body>` element (see fig. 1), in that no XML markup is allowed in it; on the other hand, since base texts in LAULA XML coincide with the original TEI XML texts, tokenization layers can reference tokens conveniently through XPath expressions following CTS structures.

4.3. Sentence Segmentation

Similarly to tokenization, sentence segmentation is achieved in *OGA* through a rule-based approach. In fact, sentence boundaries are regularly signalled in modern editions of AG texts by the period, semicolon, and middle dot punctuation marks.

The algorithm also addresses the issue of use of parentheses – mostly of editorial meaning – in conjunction with other sentence-final punctuation marks, in that their order is not standardized, but usually follows a modern language’s style rules. It has been found that 9 sentences out of 2,234 (i.e., error rate of about 0.0040) are wrongly segmented because of OCR errors in the original texts.

The sentence segmentation layer is made available only in LAULA XML; as appears in Section 3, PAULA XML is used as a serialization format for conversion into relANNIS, i.e., the file format for ANNIS, which displays and queries annotations without the help of sentence boundaries.

4.4. Morphosyntactic Annotation and Lemmatization

The morphosyntactic annotation in *OGA 0.2.0* is performed automatically using Trankit.[39] Trankit is a transformer-based parser, which, relying on the pretrained model XLM-RoBERTa, proved to deliver state-of-the art results for UD treebanks.[40] Moreover, it is very time-efficient because only the weights of a few layers are trained, while all the others remain fixed.

Trankit delivered the best results for AG (see tab. 1) when compared to three other parsers,[41] i.e., a baseline LSTM-based parser with randomly initialized character embeddings called Dithrax, and two parsers updating the token embeddings provided by GreBERTa and PhilBERTa,[42] respectively.

37 The list of them can be found at https://github.com/OperaGraecaAdnotata/OGA (last access 11.07.2025), subdirectory *tokenize*.

38 A RIGHT SINGLE QUOTATION MARK used to mean elision is separated from the preceding token.

39 Nguyen et al. (2021).

40 https://trankit.readthedocs.io/en/latest/performance.html (last access 11.07.2025).

41 See Celano (2025) for the detailed analysis of the comparison of the parsers as well as lemmatizers.

42 Riemenschneider / Frank (2023).

Lemmatization was performed using GreTa, an encoder-decoder transformer, which resulted in the best performing model (see Tab. 1), when compared to Dithrax and PhilTa.[43]

POS	XPOS	Feats	AllTags	UAS	LAS	Lemmata
96.41	91.90	94.77	91.56	82.60	77.10	91.41

Tab. 1: F1 Scores for the Trankit parser and the GreTa lemmatizer.

Trankit and GreTa were trained on (i) the AGDT v2.1,[44] (ii) the Gorman Trees,[45] and (iii) the Pedalion Trees.[46] All these treebanks adopt the same annotation scheme (i.e., the AGDT one) and altogether constitute by far the largest morphosyntactically annotated corpus for AG (1.2M+ tokens). Since the corpus comprised texts of different genre and age (from approximately 900 BCE to 400 CE), the Trankit and GreTa models[47] are expected to be able to generalize much better than other existing models, such as, for example, the ones trained on UD treebanks, whose sizes are much smaller – the token count for the UD Perseus Treebank plus the UD PROIEL treebank is about 416K tokens.

Although all the above-mentioned treebanks follow the same annotation scheme, their texts, which were annotated by many different annotators, needed to be made consistent internally, among themselves, and compliant with the *OGA* tokenization scheme. All treebank fields, such as word form, lemma, POS, etc., required a non-trivial normalization because of some clearly erroneous or null values. All apostrophe-looking characters were converted into MODIFIER LETTER APOSTROPHE (U+02BC). Some tokens, such as the coordinate conjunctions οὐδὲ and εἴτε, were tokenized. Sentences with syntactic cycles were corrected or deleted.[48]

The AGDT annotation scheme derives from that of the Prague Dependency Treebank[49] (Hajič et al., 2018) and consists of four annotation layers: (i) morphological layer; (ii) lemma layer; (iii) dependency structure layer; and (iv) dependency function layer. The morphological annotation is represented as a 9-character string, where the first character is the part of speech of a token and the remaining characters its morphological features. The lemma annotation refers to the dictionary entry for a token: notably, the AGDT annotation scheme follows the convention of representing lemmata as single word forms, even if, in traditional grammar, lemmata consist of more word forms, which describe a token more accurately: for example, a lemma for a noun in an AG dictionary consists not only of its nominative form, as in the AGDT, but also of its genitive, which conveys relevant information about its declension.

The syntactic annotation follows a dependency formalism, which consists in the identification of directed relations between a head token and its dependent tokens – a head token can have more than one dependent, but not vice versa. When considered together, all relations form a directed acyclic graph, more commonly described as an (upside-down) syntactic tree. Each syntactic dependency is also typed with a syntactic label expressing the function a dependent has with reference to its head (e.g., subject or attribute). Dependency grammar formalism is quite popular in computational linguistics because it represents a balanced trade-off between syntactic analysis precision and annotation feasibility. De-

43 Riemenschneider / Frank (2023).

44 https://github.com/PerseusDL/treebank_data (last access 11.07.2025).

45 https://github.com/vgorman1/Greek-Dependency-Trees (last access 11.07.2025); Gorman (2020).

46 https://github.com/perseids-publications/pedalion-trees (last access 11.07.2025).

47 https://git.informatik.uni-leipzig.de/celano, subdirectories `morphosyntactic_parser_for_OGA` and `lemmatizer_for_OGA` (last access 11.07.2025).

48 See Celano (2025) for more details.

49 Hajič et al. (2018).

scribing the many details of the syntactic annotation is outside the scope of the article: the reader is referred for them to Celano (2019) and the annotation guidelines.[50]

4.5. The IPA Transcription Layer

OGA contains an experimental IPA transcription layer (for tokens). IPA transcription allows querying the prosodic structure of AG, which could be enormously beneficial to studies on AG poetry or prose rhythm.

The annotation is the output of a ByT5 model trained on *Wiktionary* lemmata,[51] achieving an accuracy of 0.83 in producing correct IPA transcriptions. The training dataset contained both AG and Latin IPA transcriptions. *Wiktionary* provides IPA transcriptions corresponding to different historical periods: *OGA* currently provides the '5th century BCE Attic' pronunciation for AG and the 'Classical Latin' one for Latin. For example, the word ἁβρός corresponds to the IPA transcription /ha.brós/, while the word ἀασθείς to /a.aːs.tʰĕːs/.

It is to be noted that, while AG orthography easily allows conversion to IPA transcriptions of many graphemes (because of one-to-one correspondences), this does not hold true for a few more complex cases: for example, the length of the vowel α is not marked, and more complex rules are needed to treat the conversion of diphthongs such as ει. Moreover, identification of syllable division, which is marked by full stops in the IPA transcriptions above, is not a trivial task. For all these reasons, the task was approached using machine learning rather than a rule-based method.

4.6. The Composition Date Layer

Being able to identify when a text was composed is of crucial importance for query purposes. For this reason, texts in *OGA* were annotated for estimated composition dates. As is well known, the chronology of ancient works cannot always be precisely determined. Sometimes, a composition date is highly disputed or unknown, and this represents a modelling issue. To address this, composition dates were annotated by one student expert in AG literature following academic reference works or Wikipedia: since different modern authors can suggest different dates, the chosen dates have been documented with their sources,[52] so that querying of the corpus and future changes or corrections can be facilitated.[53]

4.7. The CTS Structure Layer

CTS structure generally refers to a hierarchical citation system used by the CTS protocol to retrieve passages from a literary work by means of URNs,[54] which include identifiers for an author, work, edition, and passage.

In reference to the annotation layer, however, the phrase 'CTS structure' is used with a narrower scope, indicating passage tags assigned to tokens. For example, since Herodotus' *Histories* are divided into books, chapters, and sections, a CTS structure tag provides each token of this work with the number of the book, chapter, and section it belongs to.

---

50 https://github.com/PerseusDL/treebank_data, subdirectories `v1/greek/docs` and `AGDT2/guidelines` (last access 11.07.2025).

51 https://git.informatik.uni-leipzig.de/celano, subdirectory `ipa_transcription_for_OGA` (last access 11.07.2025).

52 https://github.com/OperaGraecaAdnotata/OGA, subdirectory `work_chronology` (last access 11.07.2025).

53 The addition of composition dates represented a complex modelling problem also because ANNIS does not currently support data types such as numbers or dates; see more details at https://github.com/OperaGraecaAdnotata/OGA, subdirectory `query` (last access 11.07.2025).

54 Blackwell / Smith (2020).

Indeed, *OGA* base texts are based only on those TEI XML texts containing a `<refsDecl n="CTS">` element (within the `<encodingDesc>` element), in which an XPath expression is provided for the identification of work divisions. For example, in the file `tlg1600.tlg001.perseus-grc2.xml` (corresponding to *Flavii Philostrati Opera*, Vol. 2), the following XPath expression is given:

```
/tei:TEI/tei:text/tei:body/tei:div/tei:div[@n="$1"]/tei:div[@n="$2"]
```

The variables `$1` and `$2` stand for the numbers of, respectively, the first and second kinds of division (the `<div>` elements), which, in this case, correspond to "book" and "chapter"—the names of these divisions are specified within a `<refsDecl>` element.

The importance of the CTS citation layer for philological, historical, and linguistic studies cannot be overstated, considering how heavily they rely on the network of references made possible through such a passage numbering system.

5. Conclusion

In this paper, the architecture of the beta version 0.2.0 of the *OGA* corpus was presented. The base texts and their nine annotation layers have been described: (i) tokenization; (ii) sentence segmentation; (iii) lemmatization; (iv) morphology; (v) dependency structure; (vi) dependency function; (vii) IPA transcription; (viii) composition date; and (ix) CTS structure. The layers are serialized as standoff PAULA XML and standoff LAULA XML and can be queried online through ANNIS.[55]

OGA is generated automatically through a series of scripts, which can be re-executed to reproduce it or update it, if its base texts, which derive from independently managed GitHub repositories, change or new texts are added.

Base texts in *OGA* are extracted from original TEI XML files, in which text is encoded together with paratext, and tokenized into morphosyntactic tokens with a rule-based system. The corpus presents a morphosyntactic annotation and lemmatization based on models trained on treebank data: a significant challenge was ensuring that the tokenization schemes of treebank texts and *OGA* base texts matched as closely as possible. The corpus is enriched with an experimental IPA transcription layer based on a ByT5 model trained on *Wiktionary* data and with a composition date layer that associates each work with an estimated composition date based on manual annotation. Finally, since all original TEI XML files specify the internal structure of a work, *OGA* comprises an annotation layer that allows retrieval of passages following a canonical citation scheme (CTS structure layer).

Building a multilayer corpus is challenging because a number of issues arise that do not have definitive answers. For example, identification of what counts as text and what counts as paratext is not always clear-cut (for example, should the title of a work be considered as text?)[56], and annotation schemes, including the tokenization one, are questionable on many points. This holds particularly true when annotation is applied to texts belonging to very different ages and genres.

It is also difficult to guarantee consistency at any level: the quality of the original texts, some of which still contain many errors, significantly impacts the quality of annotation layers. Similarly, trying to train models on data that are as similar to the *OGA* ones as possible requires intense normalization work and extensive manual annotation effort. Evaluating annotation quality on such a large corpus as *OGA* is also arduous, and more work is needed in the future.

55 https://annis.varro.informatik.uni-leipzig.de (last access 11.07.2025).

56 The choice has an impact on how the final text is represented and can be annotated.

Finally, it is important to note that annotation modeling choices are also dependent on the technologies and standards available: *OGA* adopts PAULA XML, a de facto standard standoff format, which provides a number of advantages, including the possibility of making the data easily queryable through ANNIS. As noted above, however, a few issues with PAULA XML arise in terms of efficient parsing, which are bound to be exacerbated as the size of the corpus increases.

For all of these reasons, a multilayer corpus should be considered work in progress, continuously evolving with the addition of new texts, more accurate annotations, and the adoption of new technologies and standards.

List of Abbreviations

AG	Ancient Greek
AGDT	Ancient Greek Dependency Treebank
CTS	Canonical Text Services
OGA	Opera Graeca Adnotata
PDL	Perseus Digital Library
SV	Scaife Viewer
TLG	Thesaurus Linguae Graecae

Sources

Online Sources

https://corpus-tools.org/annis (last access 11.07.2025).

https://corpus-tools.org/pepper (last access 11.07.2025).

https://github.com/alekkeersmaekers/glaux (last access 11.07.2025).

https://github.com/korpling/paula-xml (last access 11.07.2025).

https://github.com/OpenGreekAndLatin/First1KGreek (last access 11.07.2025).

https://github.com/OperaGraecaAdnotata/OGA (last access 11.07.2025).

https://github.com/PatristicTextArchive (last access 11.07.2025).

https://github.com/vgorman1/Greek-Dependency-Trees (last access 11.07.2025).

https://github.com/perseids-publications/pedalion-trees (last access 11.07.2025).

https://github.com/PerseusDL/canonical-greekLit (last access 11.07.2025).

https://github.com/PerseusDL/treebank_data (last access 11.07.2025).

https://git.informatik.uni-leipzig.de/celano (last access 11.07.2025).

https://trankit.readthedocs.io/en/latest/performance.html (last access 11.07.2025).

https://www.dfhg-project.org (last access 11.07.2025).

https://www.iso.org/standard/37326.html (last access 11.07.2025).

Digital Corpora

ANNIS = https://annis.varro.informatik.uni-leipzig.de (last access 11.07.2025).

Diogenes = https://d.iogen.es (last access 11.07.2025).

Diorisis = https://doi.org/10.6084/m9.figshare.6187256.v1 (last access 11.07.2025).

Opera Graeca Adnotata v0.1.0 = https://doi.org/10.5281/zenodo.8158675 (last access 11.07.2025).

Opera Graeca Adnotata v0.2.0 = https://doi.org/10.5281/zenodo.14206061 (last access 11.07.2025).

GLAUx = https://glaux.be/ (last access 11.07.2025).

Perseus Digital Library = https://www.perseus.tufts.edu/hopper (last access 11.07.2025).

Philologic4 = https://perseus.uchicago.edu (last access 11.07.2025).

Scaife Viewer = https://scaife.perseus.org (last access 11.07.2025).

Thesaurus Linguae Graecae = https://stephanus.tlg.uci.edu (last access 11.07.2025).

References

Blackwell / Smith (2020). C. W. Blackwell / N. Smith, The CITE Architecture (CTS/CITE) for Analysis and Alignment. it-information Technology 62/2 (2020), 91–98, https://doi.org/10.1515/itit-2019-0044 (last access 11.07.2025).

Berti (2021): M. Berti, Digital Editions of Historical Fragmentary Texts, Heidelberg 2021.

Boschetti et al. (2009): F. Boschetti / M. Romanello / A. Babeu / D. Bamman / G. Crane, Improving OCR Accuracy for Classical Critical Editions, in: Proceedings of the 13th European Conference on Research and Advanced Technology for Digital Libraries (2009), 156–167, https://dl.acm.org/doi/10.5555/1812799.1812822 (last access 11.07.2025).

Celano (2019): G. G. A. Celano, The Dependency Treebanks for Ancient Greek and Latin, in: Monica Berti (ed.), Digital Classical Philology: Ancient Greek and Latin in the Digital Revolution, Berlin / Boston (2019), 279–298, https://doi.org/10.1515/9783110599572-016 (last access 11.07.2025).

Celano (2024a): G. G. A. Celano, Opera Graeca Adnotata: Building a 34M+ Token Multilayer Corpus for Ancient Greek, ArXiv (2024), https://arxiv.org/abs/2404.00739 (last access 11.07.2025).

Celano (2024b): G. G. A. Celano, Opera Graeca Adnotata 0.2.0, Zenodo (2024), https://doi.org/10.5281/zenodo.14206061 (last access 11.07.2025).

Celano (2025): G. G. A. Celano, A State-of-the-Art Morphosyntactic Parser and Lemmatizer for Ancient Greek, in: Proceedings of the First Workshop on Natural Language Processing and Language Models for Digital Humanities (2025), 48–65, https://aclanthology.org/2025.lm4dh-1.5/ (last access 30.03.2026).

Gorman (2020): V. B. Gorman, Dependency Treebanks of Ancient Greek Prose, Journal of Open Humanities Data 6/1 (2020), https://openhumanitiesdata.metajnl.com/articles/10.5334/johd.13 (last access 11.07.2025).

Hajič et al. (2018): J. Hajič / E. Bejček / A. Bémová / E. Buráňová / E. Hajičová / J. Havelka / P. Homola / J. Kárník / V. Kettnerová / N. Klyueva / V. Kolářová / L. Kučová / M. Lopatková / M. Mikulová / J. Mírovský / A. Nedoluzhko / P. Pajas / J. Panevová / L. Poláková / M. Rysová / P. Sgall / J. Spoustová / P. Straňák / P. Synková / M. Ševčíková / J. Štěpánek / Z. Urešová / B. Vidová Hladká / D. Zeman / Š. Zikánová / Z. Žabokrtský, Prague Dependency Treebank 3/5 (2018), http://hdl.handle.net/11234/1-2621 (last access 11.07.2025).

Keersmaekers (2021): A. Keersmaekers, The GLAUx Corpus: Methodological Issues in Designing a Long-Term, Diverse, Multi-Layered Corpus of Ancient Greek, in: Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change (2021), 39–50, https://aclanthology.org/2021.lchange-1.6/ (last access 11.07.2025).

Keersmaekers / Van Hal (2024): A. Keersmaekers / T. Van Hal, Creating a Large-Scale Diachronic Corpus Resource: Automated Parsing in the Greek Papyri (and Beyond), Natural Language Engineering 30/5 (2024), 1035–1064, https://doi.org/10.1017/S1351324923000384 (last access 11.07.2025).

Krause / Zeldes (2014): T. Krause / A. Zeldes, ANNIS3: A New Architecture for Generic Corpus Query and Visualization, Digital Scholarship in the Humanities 31/1 (2014), 118–139, https://doi.org/10.1093/llc/fqu057 (last access 11.07.2025).

Nguyen et al. (2021): M. V. Nguyen / V. D. Lai / P. B. Veyseh / T. H. Nguyen, Trankit: A Light-Weight Transformer-Based Toolkit for Multilingual Natural Language Processing, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (2021), 80–90, https://aclanthology.org/2021.eacl-demos.10/ (last access 11.07.2025).

Odebrecht et al. (2017): C. Odebrecht / M. Belz / A. Zeldes / A. Lüdeling / T. Krause, RIDGES Herbology: Designing a Diachronic Multi-Layer Corpus, Language Resources and Evaluation 51 (2017), 695–725, https://link.springer.com/article/10.1007/s10579-016-9374-3 (last access 11.07.2025).

Przepiórkowski et al. (2011): A. Przepiórkowski / M. Bańko / R. L. Górski / B. Lewandowska-Tomaszczyk / M. Łaziński / P. Pęzik, National Corpus of Polish, in: Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań 2011, 259–263.

Riemenschneider / Frank (2023): F. Riemenschneider / A. Frank, Exploring Large Language Models for Classical Philology, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (2023), 15181–15199, https://aclanthology.org/2023.acl-long.846/ (last access 11.07.2025).

Schroeder / Zeldes (2016): C. T. Schroeder / A. Zeldes, Raiders of the Lost Corpus, Digital Humanities Quarterly 10/2 (2016), https://www.digitalhumanities.org/dhq/vol/10/2/000247/000247.html (last access 11.07.2025).

Vatri / McGillivray (2018): A. Vatri / B. McGillivray, The Diorisis Ancient Greek Corpus: Linguistics and Literature, Research Data Journal for the Humanities and Social Sciences 3/1 (2018), 55–65, https://doi.org/10.1163/24523666-01000013 (last access 11.07.2025).

Zeldes (2017): A. Zeldes, The GUM Corpus: Creating Multilayer Resources in the Classroom, Language Resources and Evaluation 51 (2017), 581–612, http://dx.doi.org/10.1007/s10579-016-9343-x (last access 11.07.2025).

Zeldes (2018): A. Zeldes, Multilayer Corpus Studies, New York 2018.

Figure References

Fig. 1: Standoff annotation layers in PAULA XML.

Fig. 2: Pipeline for the creation of *OGA*.

Author Contact Information[57]

Dr. Giuseppe G. A. Celano
Universität Leipzig
Institut für Informatik
Augustusplatz 10
04109 Leipzig
E-mail: celano@informatik.uni-leipzig.de

57 The rights pertaining to content, text, graphics, and images, unless otherwise noted, are reserved by the author. This contribution is licensed under CC BY-SA 4.0.