DIGITAL CLASSICS ONLINE

Beyond Screenshots: Machine-Actionable, Canonical, Semantic Citation of Graphed Data

Christopher William Blackwell

Abstract: In 2016 and 2017, a series of conferences for European philologists was organized around the question, "What digital services, collections or curricula need to be developed so that a field of study can flourish in a digital society?" This paper argues for the need to cite graphs of data with machine-actionable canonical citation, independently of the data organized by a graph. It describes ongoing work to implement a "Canonical Graph Service" into the CITE/CTS framework used by the *Homer Multitext* (HMT). It describes citation of graphs, parts of graphs, and sub-graphs by URN, with some examples of how such URN citations might usefully be resolved. Finally, I discuss the limits of this approach, problems that will not be solved by a Canonical Graph Service. This approach may facilitate the creation of generic tools for documenting syntax across languages, integrating data from diverse projects, and opening new areas of research to scholars outside of quantitative fields.

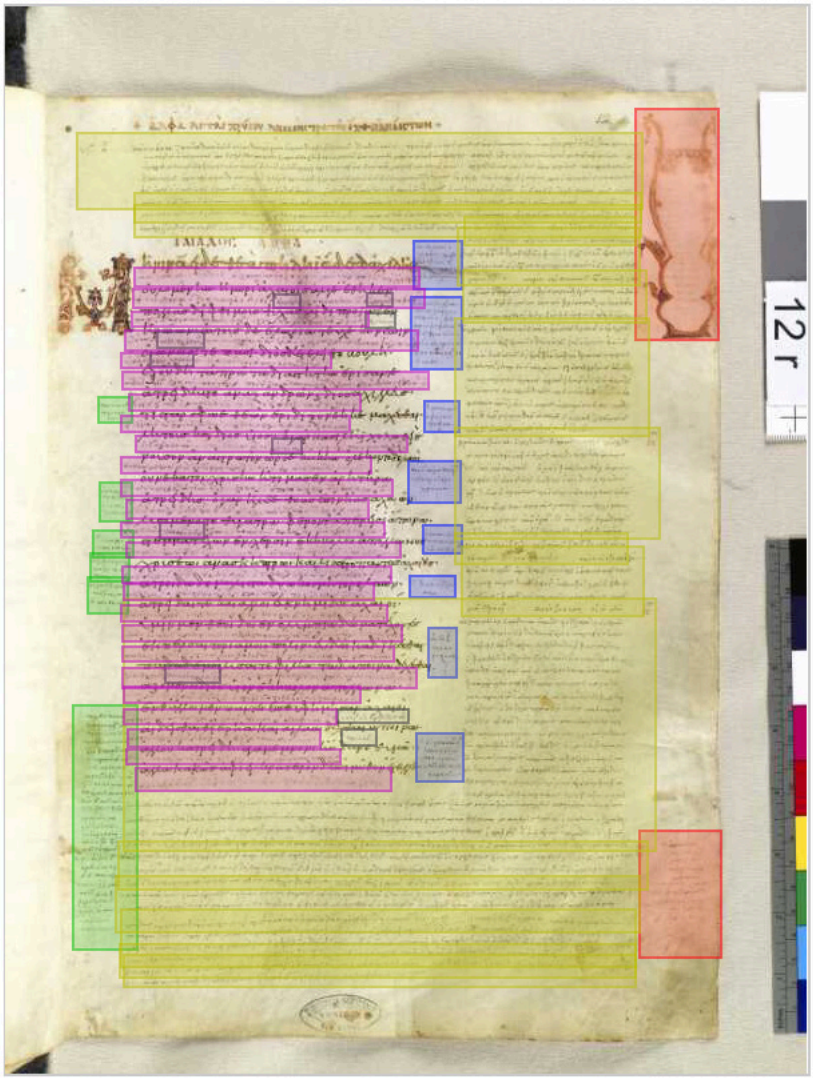

Figure 1: A visualization of texts aligned to a physical object, via the medium of a digital image. An implicit graph.

1. Background: Citation and Quotation of Data and Text

Research with any complex dataset requires many *procedural* approaches, from computational processes like find or diff, to entirely manual and intellectual tasks like reading ancient Greek or disambiguating names. But to publish the results of humanist research, we need an architecture that allows *declarative* scholarship. Once we have found things, or asserted them, we need to be able to name them. Scholars name things with *citation*.[1]

Classical scholarship has always relied on *canonical citation* for declarative scholarship. Citations, *e.g.* "John 3:16" or "*Iliad* 24.1", identify passages of text across editions and across technologies. For work in the digital realm, the Homer Multitext (hereafter "HMT") has developed a digital architecture for *machine-actionable canonical citation* that allows us to identify our objects of study with citations that are precise while retaining access to the larger context.[2]

This architecture is CITE, for "Collections, Indices, Texts, and Extensions". It is based on two standards for citations in URN format.[3] The URN citations defined by CITE allow us to cite scholarly data, and by virtue of being machine-actionable, we can *resolve* URNs to the data which they identify, and thus automate scholarly *quotation*.[4]

With the CITE Architecture, we define "text" as "an ordered hierarchy of citation objects"[5] can identify passages of text precisely with CTS URNs that capture the semantics of that definition:

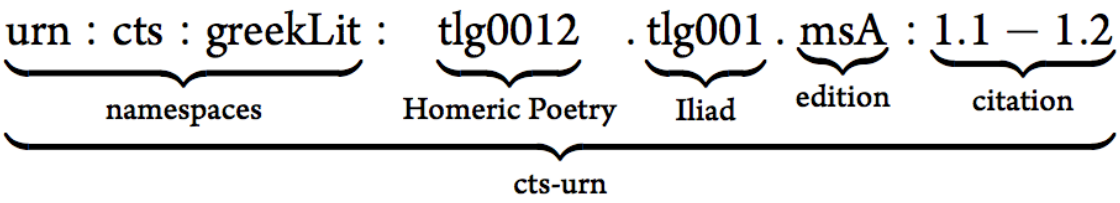

Figure 2: A CTS URN.

1 The information industry has exerted great intellectual effort to extract information and insight in the absence of any citation-practice, *e.g.* Guha & Gupta (2015). While this effort has led to many sophisticated and powerful heuristics and algorithms in computer science, the industry and Academe make radically different assumption. Guha and Gupta say, "Expecting a large number of different sites to use the same unique identifiers for these millions of entities is unrealistic." (2) A thousand years of philology, on the other hand, has depended on a large number of different scholars, over centuries, using the same unique identifiers for millions of entities.

2 http://www.homermultitext.org

3 http://cite-architecture.github.io

4 Robert Sokolowski calls quotation a 'curious conjunction of begin able to name and to contain'. Sokolowski (1984) 699. V.A. Howard is more succinct: quotation is 'replication-plus-reference'. Howard (1974) 310. For the HMT we are less interested in the metaphysical aspects of quotation than in the practical ones. Quotation, when accompanied by citation, allows us to bring the reader's attention to bear on a particular part of a larger whole efficiently and without losing the surrounding context. A work of Biblical exegesis, for example, can quote or merely cite 'Genesis 1:29' without having to reproduce the entire Hebrew Bible, or even the Book of Genesis; a reader can resolve that citation to a particular passage about the creation of plants, and can see that passage as a discrete node at the bottom of a narrowing hierarchy: Hebrew Bible, *Genesis*, Chapter 1, Verse 29. We take this for granted as philologists.

5 See Smith & Weaver (2009).

If an object of study is not an "ordered hierarchy of citation objects", CITE offers CITE URNs. These identify objects in Collections, objects that share a set of properties.

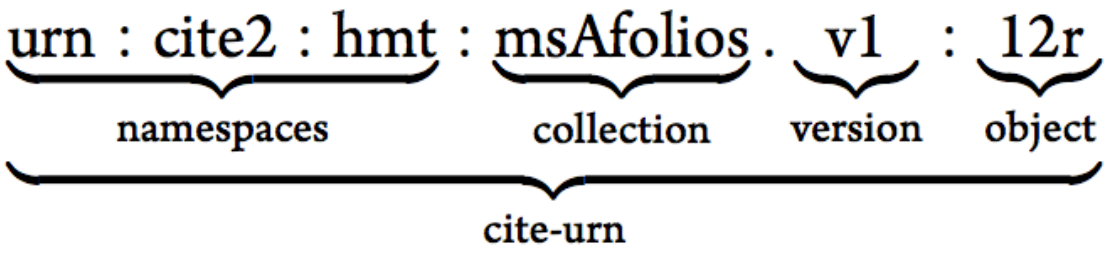

Figure 3: A CITE URN.

The URN above identifies one object (12r) in a collection (msAfolios). This is a collection of *physical objects*, the folio-sides of the Venetus A manuscript. This collection consists of records containing a shared set of properties, in this case metadata about the folios of this manuscript:

Property	Value
URN	urn:cite2:hmt:msAfolios.v1:12r
Sequence	25
Number	12
RV	recto

Table 1: A single object in a CITE Collection.

The fundamental unit of organization here is the *collection*. In this example, msAfolios is a notional collection; it is realized in a specific *version*: .v1. Any change to any of the members of this collection results in a new version.[6]

2. Background: Graphed Data

Humanists work with graphs, often more than they realize. Figure is a visualization of an implicit graph, whose nodes (or "vertices") are citations to passages of text, citations to regions-of-interest on a digital image, and citations to a physical object, folio 12 *recto* of the manuscript *Marcianus Graecus* Z454, and whose edges are scholarly assertions of relationships (in this case) defined by RDF statements. The HMT refers to graphs like this as "Diplomatic Scholarly Editions" graphs, DSE graphs.

6 In the Venetus A codex, some of the folios were added in the 15th century by Cardinal Basileus Bessarion to replace missing, original folios. If our collection added a property, replacement, with a boolean value, we might call that collection v2. urn:cite:hmt:msAfoios.v1:12r and urn:cite:hmt:msAfoios.v2:12r would identify the same object, but a citation to v1 would not resolve to any information about replacement folios.

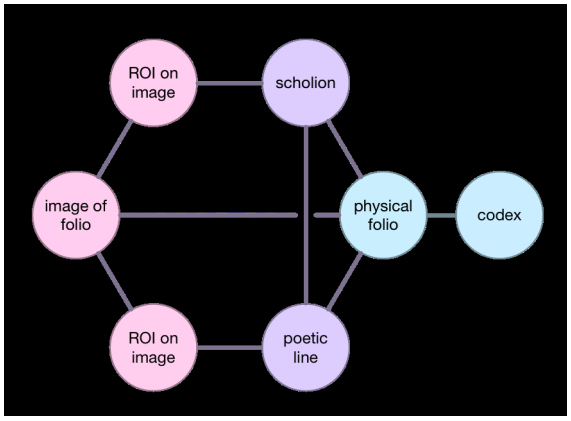

Figure 4: A graph of physical objects, images, and textual content.

In the case of this particular manuscript, which contains a text of the Homeric *Iliad* and commentary text, a specific DSE graph that might be the object of scholarly study relates a commentary text to the text it comments on, and relates both texts to their location on the physical folio, by means of visual evidence. Figure 4 is an abstract view of this scholarly assertion.

It is easy for humanists to fail to consider those commonplace associations—text, commentary, folio—as a *graph*, but other kinds of analysis are more obviously graphs. For philology in the 21st century, some of the most exciting opportunities are afforded by treebanks, explicit graphs capturing syntax or other semantic relationships, essentially documenting a reading of a text.

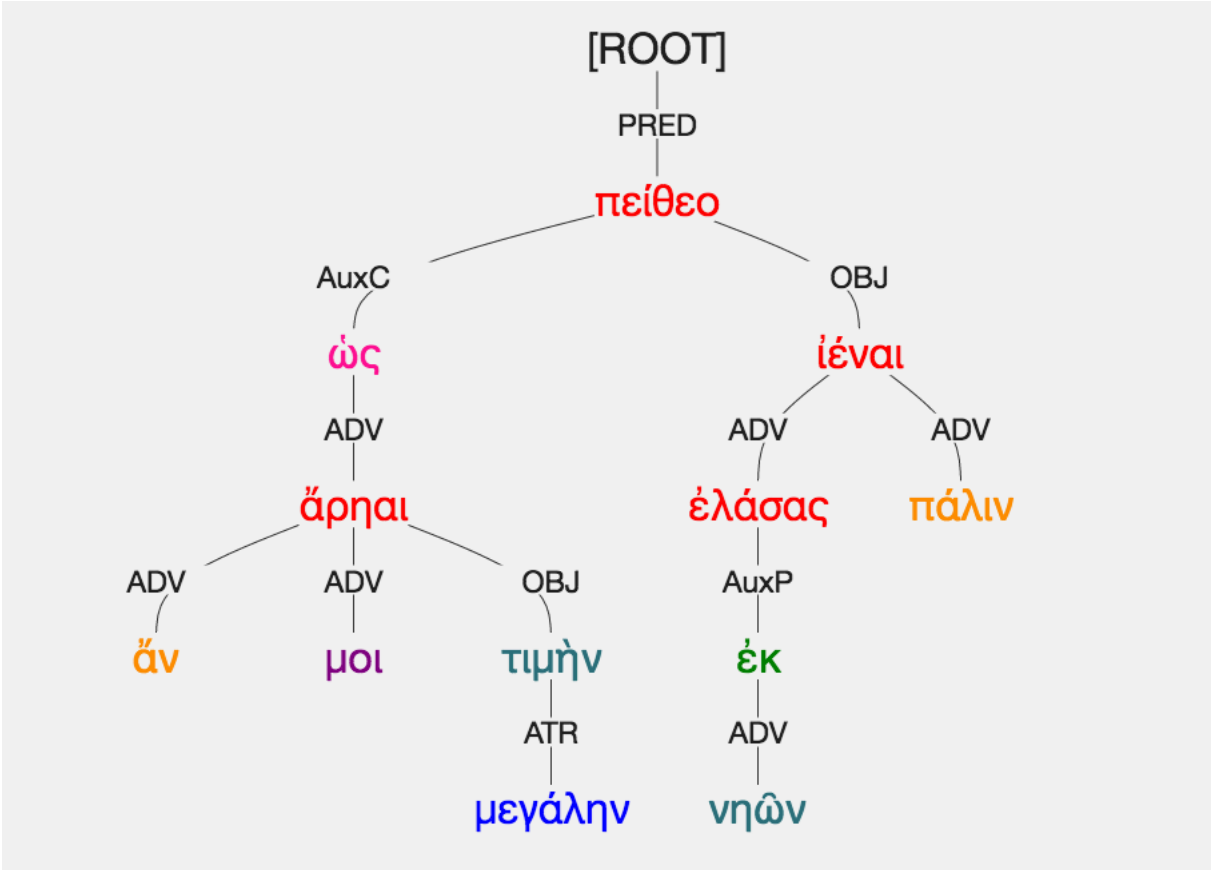

Figure 5: A graph of syntax.

In a syntactic treebank, like Figure 5, the nodes are words (and possibly punctuation), and the edges are defined syntactic relations. The treebank in Figure 5, created with the Arethusa tool[7], the word-tokens organized in the graph of syntax are also linked to morphological and lexical data.[8]

The pedagogical value of treebanking has been widely recognized, as have its potential as a tool for linguistic analysis.[9] But the potential of this kind of explicit graphing of semantic information remains to be fully exploited. For example, in the syntactic treebanks generated as part of the *Open Greek and Latin* project, or in the output of the analytical tools at *eAQUA*[10], individual textual objects are identifiable with generic, canonical citations, which can be resolved to their texts regardless of any particular technology.

The graphs themselves, explicitly drawn in the case of treebanks, or implicit in the case of word-concurrence or other analysis, are *reproducible* using those analytical tools but are not currently identified by concise citations offering similar capabilities to citations available for textual or image data. We cannot *cite* specific sub-graphs or individual nodes or edges as *members of the graph*; we cannot resolve abstract expressions of the graphs to representations of the graph in various formats.

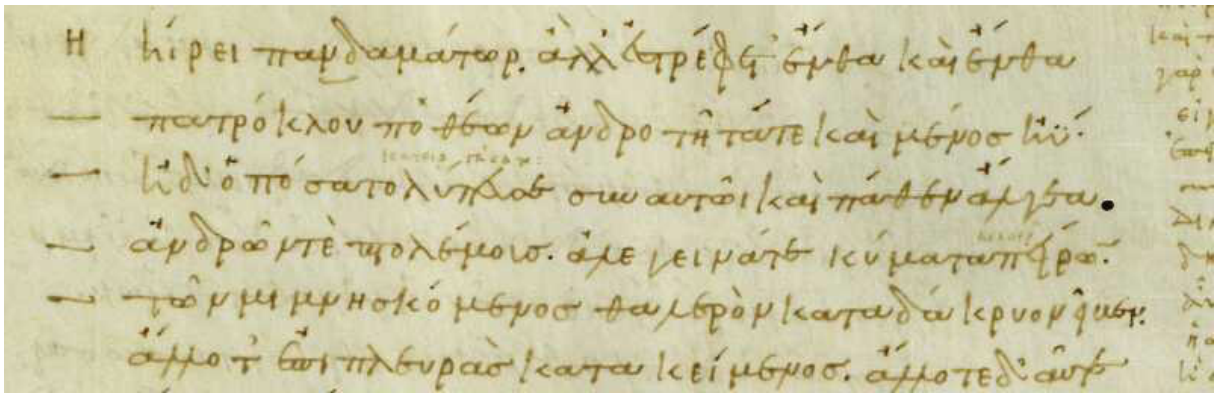

Figure 6: *Iliad* 24.5-24.10 on the Venetus A MS, showing *athetēsis* of four lines.

3. Use Case: Capturing Ancient Argument

The *scholia* on Byzantine manuscripts of the Homeric *Iliad*, the marginal comments, often discuss the editorial status of passages of the poetic text, noting when some ancient scholar of the *Iliad* expressed doubt as to the authenticity of lines of poetry. A single example will serve to illustrate how complex these ancient arguments can be. Folios 310 *verso* and 311 *recto* of *Marcianus Graecus* Z822, the "Venetus A", contains the opening lines of Book 24 of the *Iliad*. In these lines Achilles is in his tent, sleepless and grieving over his dead friend Patroclus:

7 Almas & Beaulieu (2013).

8 Preliminary work on CITE Citable Graphs has given attention to programmatic tokenization of texts specifically for syntactic annotation. While it is beyond the scope of this paper, the GitHub repository for our Citable Graph Extension includes utilities written in Scala for generated collections of syntactically significant tokens, including data on editorial status, and level of discourse (direct or indirect), from CTS texts. These utilities are written in Scala and represent the first steps in the next stage of development of the CITE architecture.

9 See, for example, Mambrini (2013); Mambrini & Passarotti (2016).

10 eAQUA; Schubert & Heyer (2013); Schubert (2013).

…αὐτὰρ Ἀχιλλεὺς
κλαῖε φίλου ἑτάρου μεμνημένος, οὐδέ μιν ὕπνος
ᾕρει πανδαμάτωρ, ἀλλ᾽ ἐστρέφετ᾽ ἔνθα καὶ ἔνθα
— Πατρόκλου ποθέων ἀνδροτῆτά τε καὶ μένος ἠΰ,
— ἠδ᾽ ὁπόσα τολύπευσε σὺν αὐτῷ καὶ πάθεν ἄλγεα
— ἀνδρῶν τε πτολέμους ἀλεγεινά τε κύματα πείρων:
— τῶν μιμνησκόμενος θαλερὸν κατὰ δάκρυον εἶβεν,
ἄλλοτ᾽ ἐπὶ πλευρὰς κατακείμενος, ἄλλοτε δ᾽ αὖτε
ὕπτιος, ἄλλοτε δὲ πρηνής:
— *Iliad* 24.3–24.11, Edition of the Venetus A.
…But Achilles
wept, remembering his beloved companion, nor did sleep,
the all-mastering, hold him, but he turned this way and that way
yearning for the manliness and noble strength of Patroclus
and all the things he had accomplished with him, and all the pains he suffered
passing through the wars of men and the pain-giving waves
remembering all these things, he let fall a great tear
at times lying on his side, at other times again
on his back, at times on his front.

On the Venetus A Manuscript, the scribe has included *obeloi* to the left of lines 6–9; these indicate *athetēsis*, an editor's decision that the lines are somehow inauthentic (see Figure 6).

A Scholion commenting on this passage explains the *athetēsis* (Scholion 24. A2 [HMT Edition, G. Hedden and M. Velthuisen, trans.]):

Yearning for Patroclus [From this phrase] until [the line beginning] "τῶν μιμνησκόμενος" the lines are athetized because they are cheap (εὐτελής). And with them lifted out, the grief of Achilles is made clear more emphatically:

- But he turned this way and that
- At times on his back…

And "ἀνδροτῆτα" and "μένος" indicate the same thing, for there is no difference, and [Homer] never uses "ἀνδροτῆτα" for "ἀνδρείαν", instead he uses "ἡνορέαν". And "remembering these things" is awkward, because he has said "remembering his companion" above. And Aristophanes athetized these lines earlier. If you don't want to athetize the lines, then either [ποθέων] should qualify everything (the main verbs ἐστρέφετ᾽ [24.5] and εἶβεν [24.9]) or there needs to be explicit punctuation after τὸ κύματα πείρων.>

The comment is in dense scholarly Greek. It notes, first, that the scholar Aristarchus athetized four lines (as represented by the *obeloi* in the main text on the manuscript); before stating Aristarchus' reasons, it shows that the Iliadic text still makes sense with those four lines removed. It then gives two alternate ways of reading the text with those four lines in place, a "default" way, and a preferred way, seemingly based on Nicanor's (lost) work *On Iliadic Punctuation*. These three analyses can be clearly expressed with three different syntactic graphs (Figure 7, and expanded at Figures 15, 16, and 17 in the Appendix.)

The three treebanks *organize the same syntactic tokens*, so while we can cite each Iliadic line and each word in each line, it would be ideal to cite a word as a syntactic-token organized by a particular graph. The word "ἐστρέφετ'" ("he turned") is shared across three analyses, as the same lexical entity with the same morphology, but its semantic identity is different in each of the three. We should be able to cite each of those identities.

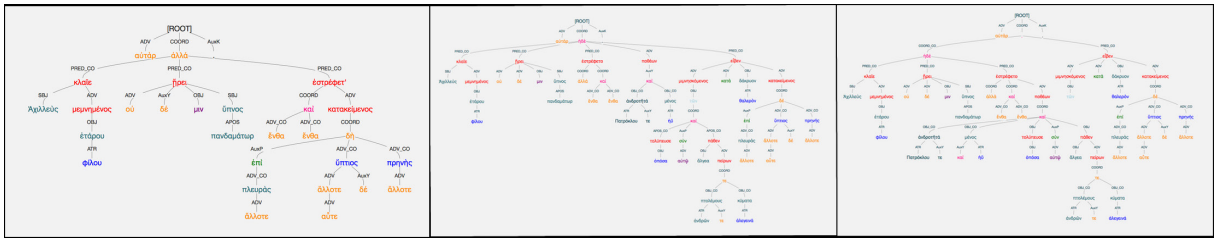

Figure 7: Three graphs of the same text, capturing three ancient readings of Iliad 24.3-24.11. Larger versions of these appear in the Appendix, Figures 15, 16, 17.

This kind of analysis permeates the Iliadic scholia. At *Iliad* 16.83–16.86, Achilles orders his friend Patroclus to drive the Trojans away from the Greek ships, but then to return and not pursue them onto the plain. The text (abridged for this example) looks like this:

πείθεο … ὡς ἄν μοι τιμὴν μεγάλην καὶ κῦδος ἄρηαι … ἐκ νηῶν ἐλάσας ἰέναι πάλιν

Heed me … so that you may raise up great honor and fame for me … having driven them from the ships, come back again.

A scholion on this passage presents ancient arguments for how to understand the syntax of the sentence: does the purpose clause ("so that… for me") act as an adverb on the verb "heed…" or as adverbial to the verb "come back"? The Greek of the scholion is dense and hard to follow, but we can express the two alternatives very clearly with two syntactic graphs (Figure 8).

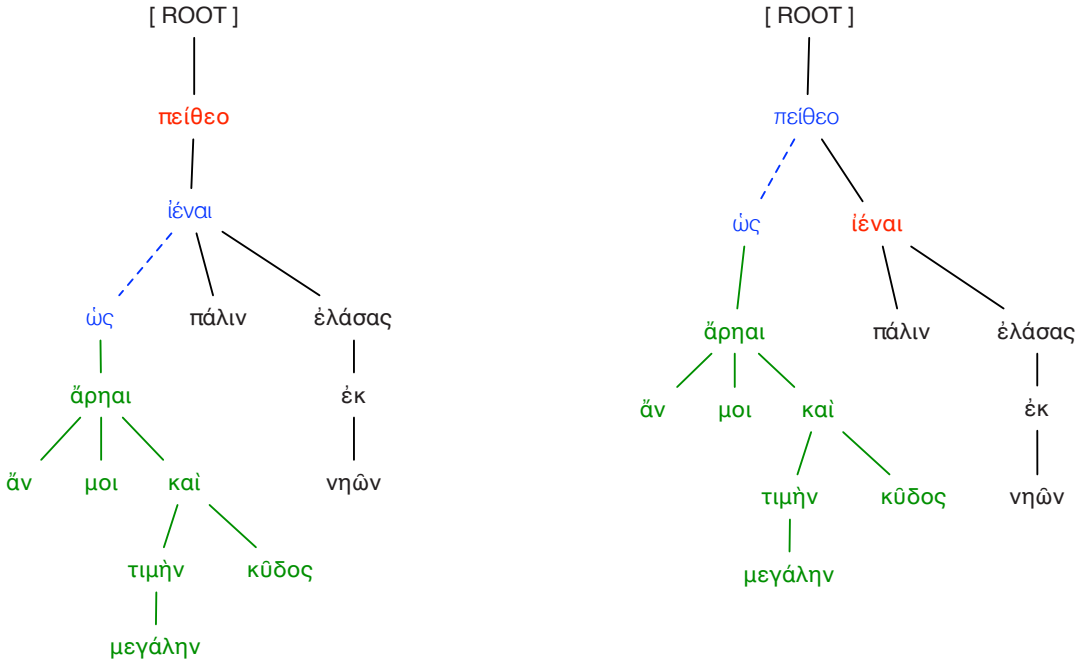

Figure 8: Two graphs of two readings, showing a relocation of a sub-graph. How can we cite these?

The ancient readers of Greek epic poetry present us with analyses, on virtually every folio of every Byzantine codex, that could best be visualized, taught from, and subjected to further automated or human analysis if the prose descriptions were made explicit as graphs. These graphs would organize the same Iliadic text in different ways. We can cite the Iliadic text with CTS URNs, and we can even cite very precisely each word-token of the text using CTS; we can likewise cite the text of the scholia. But how can we cite the graphs themselves as objects of scholarly study, concisely in a machine-actionable manner? In the example from *Iliad* 16, how can our citation practice identify the shared sub-graph (the purpose clause) whose dependency is the heart of the scholiast's comment?

4. CITE Objects and Extensions

In the CITE architecture, identifying a graph is relatively straightforward. We can create a collection of graph-objects, citing each with a URN, *e.g.* urn:cite2:hmt:dseGraph.v1:1000. This object, and all objects in this collection, might have only three or four properties: urn, label, author, and description.

But graphs are not simply collection-objects, in the sense of "objects sharing a set of properties", since each graph will have an arbitrary number of nodes and edges. A CITE URN alone cannot allow us to cite with any granularity, individual nodes or edges, paths, or sub-graphs.

CITE URNs are limited to expressing collection.version:object. By design this forces us to separate concerns, even at the cost of verbosity. CITE offers two approaches to non-textual data in hierarchies deeper than collection [+ version] + object. The most commonly used approach is to use URNs as values in a CITE object's properties. So, for example, a "folio" object may include among its properties a "codex" property, whose value is a CITE URN identifying the volume of which the folio is a part; that codex- URN provides access to the properties of the codex.

For some types of data, we cannot express objects sufficiently in the tabular character data of a CITE collection. With images, for example, a necessary expression requires a CITE collection recording URNs and other metadata for an image, and (separately) binary image files, the images themselves. At the same time, we want to make requests specific to this kind of data, images, beyond those of the generic CITE Collection Service: getBinaryImage, format transformations, scaling, cropping, &c. CITE Extensions (the E in CITE) exist for this purpose. We define an extension, cite:image, for which we define a type-specific data source and type-specific requests.

A CITE Image Collection is a CITE collection, and can be treated as such. But the cite:image extension specifies that the collection have at least three properties: urn, rights, and caption (it may have others in addition). The extension further specifies an additional data file that maps image URNs to binary image files at some specified location. An Image Service is responsible for resolving URNs to images with binary image data. The URN remains a technology-independent citation, and the concern of identity is separate from that of retrieval. In addition, this extension defines a sub-reference on an image- URN that identifies a rectangular region-of-interest:

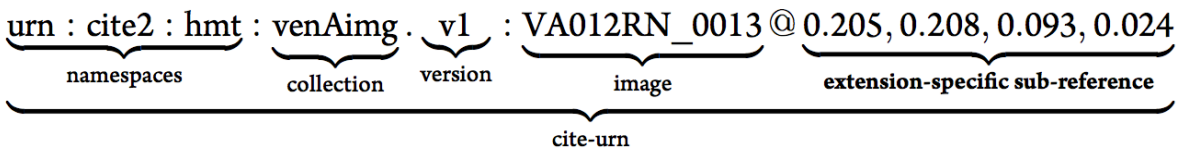

Figure 9: A CITE URN to an object in an extended collection, with a sub-reference identifying a region-of-interest on the image cited.

Following this model of CITE Extensions, to provide useful access to graphed data in CITE collections we are experimenting with a „CITE Graph Extension", a CITE URN with a defined sub-reference for identifying parts of a graph. And we are experimenting with adding a CANONICAL GRAPH SERVICE to the *Homer Multitext's* service architecture.

5. A Citable Graph Extension to CITE

A Graph Extension to CITE should allow us to cite graphs and parts of graphs, resolve those citations to various data formats, and do so *regardless of the kind of objects organized by the graph*, as long as the objects themselves are citable by URN.

So a prerequisite to any "citable graph" is citable data, either CITE collection-objects or CTS textual passages as nodes, and CITE objects defining relationships as the basis for edges.

What follows describes the generic implementation we are pursuing. A Graph Collection is an generic CITE collection, with at least six required propeties; the values of two of those properties are themselves URNs to other CITE collections. All necessary data is thus abstracted from any particular expression or technology.

A Graph Collection consists of a CITE collection, Graphs, with these properties:

Property	**Value**
URN	[CITE URN] The URN identifying a graph.
Label	[String] A short human-readable label.
Description	[String] A human-readable description.
Ordered	[Boolean] Whether the objects constituting the nodes of the graph are members of an ordered collection.
Nodes	[CITE URN] A version-level URN to a collection of Node Objects
Edges	[CITE URN] A version-level URN to a collection of Edge Objects

Table 2: Properties of a Graph object in a CITE Collection.

The Nodes collection has these properties:[11]

Property	Value
Node URN	[CITE or CTS URN] A URN identifying this Node.
Object URN	[CITE or CTS URN] A URN to a data-object organized by the graph.
Label	[String] A short label, for display.
ID	[String] A short ID, generated programmatically, identifying this node in the context of this graph. e.g. v1, v2…

Table 3: Properties of a Node object in a CITE Collection.

The Edges collections has these properties:[12]

Property	Value
Edge URN	[CITE URN] A URN identifying this Edge.
Relation URN	[CITE URN] A URN to a data-object that describes the edge's relationship
Label	[String] A short label, for display.
Source URN	[CITE URN] A URN to a object in the Node Collection. If there is no source-node (e.g. a root-dependency) this value is the Graph's URN
Target URN	[CITE URN] A URN to a object in the Node Collection.
ID	[String] A short ID, generated programmatically, identifying this Edge in the context of this graph. e.g. e1, e2…

Table 4: Properties of an Edge object in a CITE Collection.

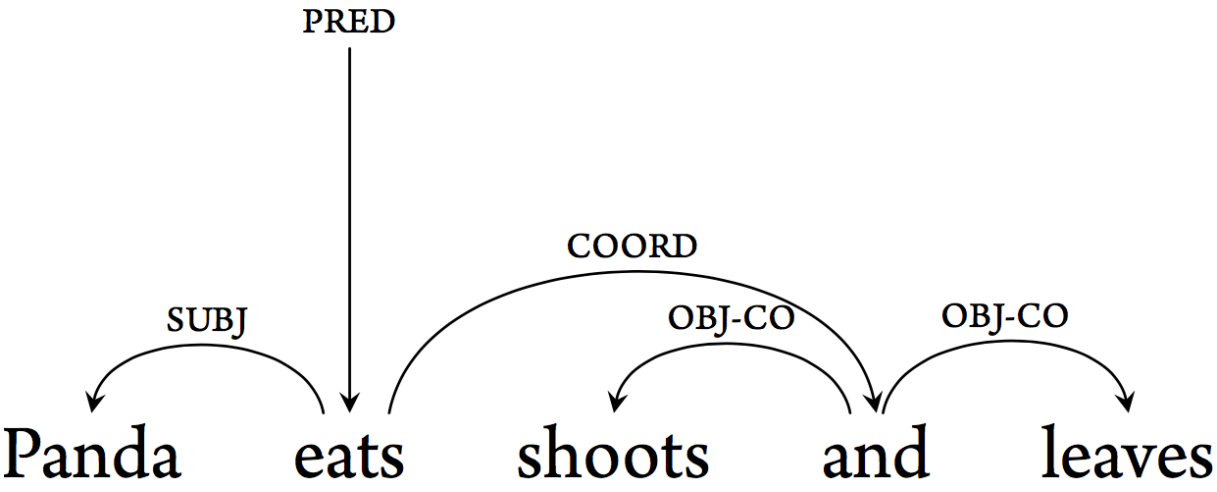

Figure 10: The Panda's Diet: Syntactic Analysis

11 If the Nodes of a graph are members of an ordered collection, their sequence in the enumeration is significant.

12 The HMT operates on the belief that all scholarly graphs are directed graphs. Specifically, all scholarly graphs are "quivers" or "directed multidigraphs (edges with own identity)". That is, a scholarly graph consists of a set of Nodes and a set of Edges; each Edge has an assigned source and target, and a scholarly asserted identity (the nature of the relationship between source and target); Nodes may be joined by more than one Edge.

6. Example Data

Figure 10 is a syntactic graph of a simple sentence that is famously subject to two interpretations. We can capture this graph in a CITE Graph Collection with the following data:

6.1 The Graph Object

URN:	urn:cite:demo:syntaxGraphs.v1:1
Label:	„The Panda‘s Diet: Syntactic Analysis“
Description:	„A syntax graph of a sentence about a panda.“
Ordered:	true
Nodes:	urn:cite:demo:sn1.v1:
Edges:	urn:cite:demo:se1.v1:

Table 5: A single Graph object in a CITE Collection.

6.2 The Nodes Collection

Urn	ObjectUrn	Label	ID
urn:cite:demo:sn1.v1:1	urn:cts:fu:demo.panda:1.1	“Panda”	v1
urn:cite:demo:sn1.v1:2	urn:cts:fu:demo.panda:1.2	“eats”	v2
urn:cite:demo:sn1.v1:3	urn:cts:fu:demo.panda:1.3	“shoots”	v3
urn:cite:demo:sn1.v1:4	urn:cts:fu:demo.panda:1.4	“and”	v4
urn:cite:demo:sn1.v1:5	urn:cts:fu:demo.panda:1.5	“leaves”	v5

Table 6: Five Node objects in a CITE Collection.

6.3 The Edge Collection

Urn:	urn:cite:demo:se1.v1:1
Relation Urn:	urn:cite:demo:syntaxRelations.v1:PRED
Label:	„PRED“
SourceURN:	urn:cite2:demo:syntaxGraphs.1.v1
TargetURN:	urn:cts:fu:demo.panda:1.2
Index:	e1

Urn:	urn:cite:demo:se1.v1:2
Relation Urn:	urn:cite2:demo:syntaxRelations.v1:SUBJ
Label:	„SUBJ“
SourceURN:	urn:cts:fu:demo.panda:1.2
TargetURN:	urn:cts:fu:demo.panda:1.1
Index:	e2

Urn:	urn:cite:demo:se1.v1:3
Relation Urn:	urn:cite2:demo:syntaxRelations.v1:COORD
Label:	„COORD“
SourceURN:	urn:cts:fu:demo.panda:1.2
TargetURN:	urn:cts:fu:demo.panda:1.4
Index:	e3

Urn:	urn:cite:demo:se1.v1:4
Relation Urn:	urn:cite2:demo:syntaxRelations.v1:OBJ_CO
Label:	„OBJ_CO“
SourceURN:	urn:cts:fu:demo.panda:1.4
TargetURN:	urn:cts:fu:demo.panda:1.3
Index:	e4

Urn:	urn:cite:demo:se1.v1:5
Relation Urn:	urn:cite2:demo:syntaxRelations.v1:OBJ_CO
Label:	„OBJ_CO“
SourceURN:	urn:cts:fu:demo.panda:1.4
TargetURN:	urn:cts:fu:demo.panda:1.5
Index:	e5

Table 7: Five Edge objects in a CITE Collection.

6.4 Notes on this Data

The Relation URN values in the Edge collection point to objects in a CITE collection and can resolve to whatever properties are recorded for those objects. In the example above, the same “syntax-relation-object” (OBJ_CO) is attached to *two* edge-objects in the graph. The URN urn:cite2:demo:syntaxRelations.v1:OBJ_CO might resolve to a collection-object with properties containing a short description, and URNs to further documentation.[13]

13 Such as the excellent, cross-referenced documentation under development by Giuseppe Celano at the University of Leipzig: https://github.com/PerseusDL/treebank_data.

Likewise, while in this case each node-object's data is textual, and identified by a CTS URN, there is no requirement that it be. Syntactic ellipsis (the omission of words) might be indicated by a CITE URN pointing to an "ellipsis" object in a collection of syntactic elements.[14]

For display to human readers, we can use the label value for nodes and edges; for automated processing, we can use the Relation URN values. For citation of parts of the graph, we can use their Index values as a sub-reference to the graph's URN.

The graph is citable as itself. Its individual nodes and edges are uniquely identified both as the data being organized (words, syntactic relations) and as member of this graph.

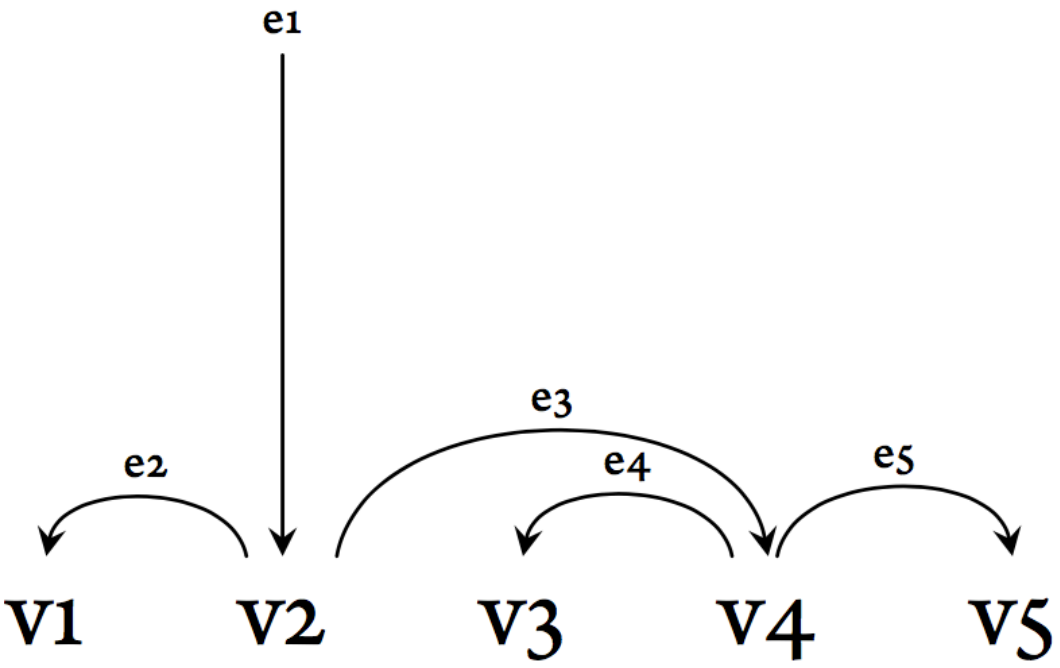

Figure 11: The Panda's Diet: a generic graph with concerns separated.

7. Citing a Graph

The data above is how a graph like this would be recorded. Our CITE Manager utility would process that into (in the HMT's implementation) data-objects defined by classes in the Scala language.[15]

A URN parameter of urn:cite2:demo:syntaxGraphs.v1:1 returns an expression of the data, as above, in any of several formats: JSON, XML, &c.

14 This is another benefit to separating the concerns of objects of study, from graphs organizing those objects of study. A text has a sequence, but a syntactic analysis might have its own sequence, with extra-textual data inserted into the sequence of text-tokens.

15 The Scala Programming Language. Scala has many benefits for work such as this, which we expect to generate a very large body of data to be processed. There is a well-supported library for working with graphed data in Scala, scalax.collection.graph. Our CGS is a body of utilities fo processing data, and an API mediating between the CITE architecture and *ScalaGraph*.

A sub-reference on the URN identifies individual edges or nodes. urn:cite2:demo:syntaxGraphs.v1:1@v1 identifies the Node whose ID value is v1 in the Graph's definition. Multiple nodes or edges can be identified by comma separated indices: e.g. urn:cite2:demo:syntaxGraphs.v1:1@e1,v3.

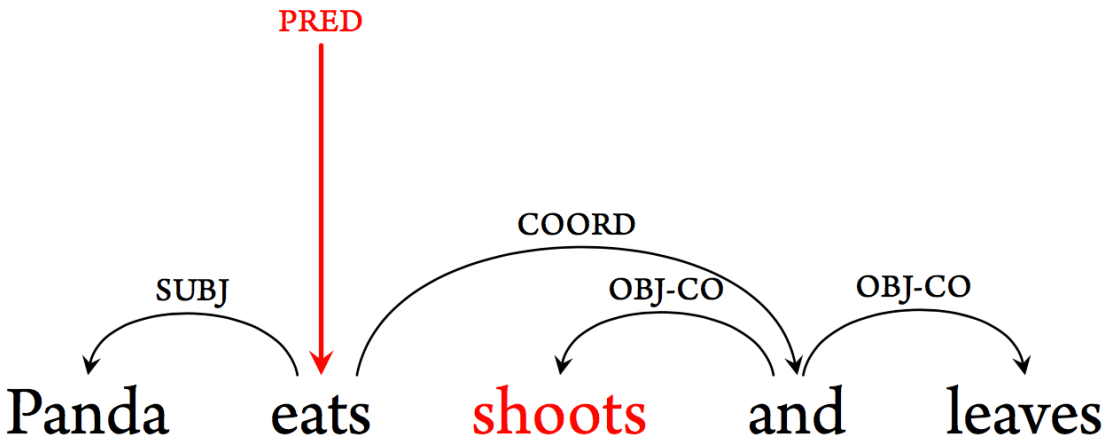

Figure 12: Citing a two objects in a graph: urn:cite2:demo:syntaxGraphs.v1:1@e1,v3

A range-notation in the sub-reference identifies a path in the graph; if the path identified in the URN is not valid for the graph, then the citation is a bad citation, like asking for Book 300 of the *Iliad*.

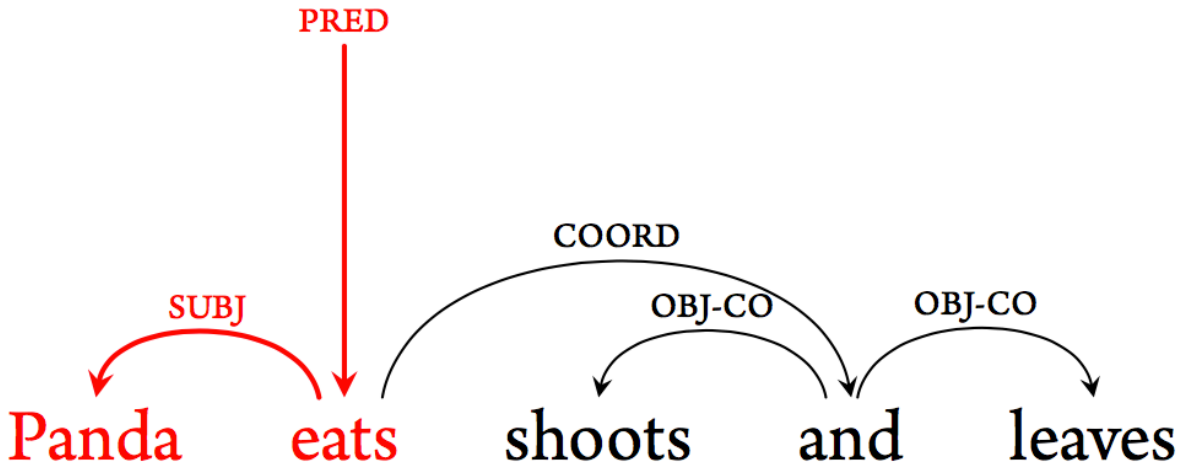

Figure 13: Citing a path between objects in a graph: urn:cite2:demo:syntaxGraphs.v1:1@e1-v1

8. Resolving Graph URNs

All work on graphed data in the HMT remains very experimental. Plans for 2017 are to implement a Graph Service that can resolve graph URNs in several ways: as JSON or XML data-structures, with the option to further transform those into d3 visualizations, .dot files, or LateX fragments.

9. Requests of a Graph Service

We plan initially to implement the following requests in a CITE Graph Service:

- GetGraph [&urn=…]. Given a URN parameter, return the graph; optional &format= parameters with possible values of "xml" or "json". Nodes and Edges identified in any sub-reference on the URN would be identified as selected in the response.
- FindInGraphs [&urn=…]. Given a CITE URN or CTS URN parameter, return the URNs of all graphs for which the parameter URN is a data-value on a Node or an Edge.
- isCyclic [&urn=…]. Returns a boolean value; useful for deciding what sort of visualizations might be most appropriate.
- ResolvePath [&urn=URN+SUBREF]. Given a URN to a graph-object with an @ delimited sub-reference to a path—*e.g.* urn:cite2:demo:syntaxGraphs.v1:1@e1-v1 from Figure —returns a URN with the range-reference resolved to a comma-separated list of nodes and edges representing the shortest path.

The question of resolving paths in a graph highlights the particular challenges of humanist computing. For network analysis or GIS applications, a "path," defined by a starting object and ending object, may be assumed to be defining the *shortest* sequence of nodes and edges between those points. Humanists are more likely to want to see *all* valid paths, and might want to take advantage of the OHCO2 text model in defining starting and ending points of a path. For example, assume a graph of (a) lines of the *Iliad*, (b) comments on those lines, and (c) Iliadic lines cited in comments. A scholar might reasonably ask for "all paths from lines in *Iliad* Book 2 (the catalogue of ships) to any line in *Iliad* Book 15 (when the Trojans are burning the ships)."

How properly to resolve the URN urn:cite2:demo:syntaxGraphs.v1:1@e1-v1? A principle of the CITE architecture has always been "you get what you ask for." A graph URN with a range subreference identifies **a** range. A Graph Service can resolve that range to an explicit list of nodes and edges—*e.g.* urn:cite2:demo:syntaxGraphs.v1:1@e1,v2,e2,v1—using well-established algorithms for finding the shortest path in a graph.[16] To identify all possible paths, a scholar could define a series of URNs explicitly identifying nodes and edges in a subreference. How that scholar might identify all possible paths between two objects in a graph is a separate concern.[17] We are concerned with identification and retrieval of scholarly objects of study, whether they are identified, or created, computationally or through human insight and intuition.

Our experience with CTS and CITE suggest that as we work with data in a Graph Service, other requests will suggest themselves. In the case of CTS, for example, requests like "GetFirstRef" proved useful by pushing back onto the server methods that are possible, but inconvenient or inefficient, for client-side applications.

10. Further capabilities

In the examples from the Iliadic commentary on the Venetus A manuscript, described above, the ancient commentators offered competing interpretations of syntax. Those analyses, expressed as graphs, different in more or less subtle ways. A Graph Service, having access to

16 See, for example, Fuhao & Jiping (2009); Noto & Satox (2000).

17 It may be impossible to isolate all possible paths between two objects in a graph algorithmically, since this problem is "NP-Hard". See Knuth (1974).

the objects organized as nodes in two graphs identified by CITE URNs, could recognize them as conflicting analyses of a single set of tokens. A CompareGraphs request, with two URNs as parameters, could return generic JSON or XML reply attaching two sets of edges to a single set of "unified nodes", a set of pairs of graph-node URNs that share the same Relation Urn as described above.

For example, the demonstration sentence above is subject to two interpretations: either the Panda eats two things (shoots and leaves), or the Panda does three things: eats, shoots, and leaves. If each of those syntactic analyses were citable by a Graph URN, the request request=compareGraphs with the parameters urn1=urn:cite:demo:syntaxGraphs.v1:1 and urn2=urn:cite:demo:syntaxGraphs.v1:2could return a generic data structure that could be visualized as in Figure 14.

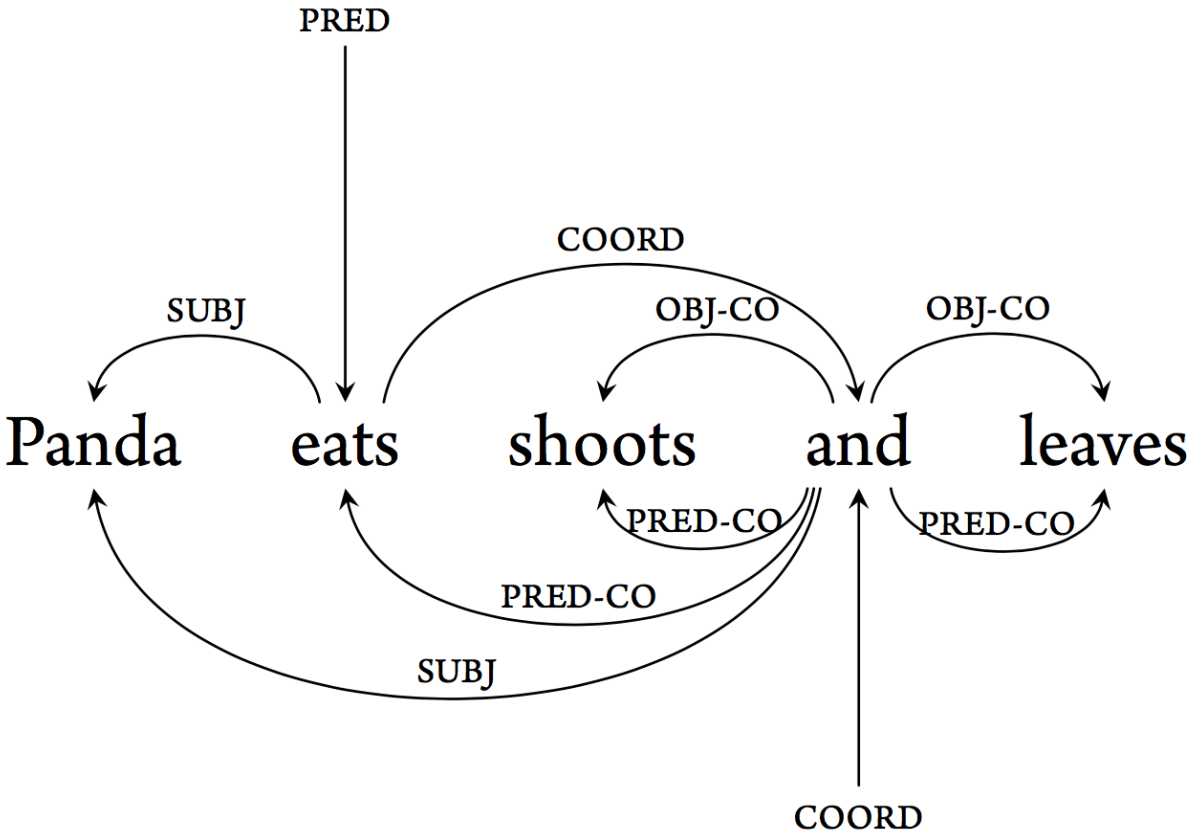

Figure 14: The Panda's diet [top], or a panda crime-spree [bottom]? An example of overlaying two graphs of the same tokens.

11. Final Notes

The CITE Graph Service will not solve, or even address, any problems of Graph Theory. Things that are computationally expensive or impossible with graphs will remain so: minimum spanning tree, longest path, subgraph isomorphism, maximum clique, &c. (Unfortunately, many of the most desirable operations, for linguists, on collections of graphs fall into this category of "NP-Complete" problems.) But we hope that this extension to the CITE Architecture will let us work with graphed data as scholars have worked with textual data for millennia, using canonical citation for citation and reproduction of objects of scholarly interest, maintaining context, independent of any particular technology. Just as canonical citation of texts allows

integration of textual evidence regardless of language, translation language, or technology, canonical citation of graphed data might serve to help integrate analytical projects. And while scholars in quantitative fields often develop skills in creating and visualizing graphed data with technologies like TikZ for or the d3 library for web-based visualization, those technologies have very steep learning curves; a generic Graph Service, by separating concerns, might make creation, publication, and analysis of graphed data more accessible to a wider research community.

Immediate uses for a Graph Service would to capture syntax and DSE relationships, particularly among scholia on different manuscripts that reproduce the same ancient sources or seem to cross-reference ancient sources. Other kinds of semantic graphs, such as "tectogrammatic" graphs[18], would be valuable additions, especially as they might analyze alternate readings ("multiforms") of the poetic text preserved in the scholia.

In parallel to work on a service architecture for graphs, some attention will have to be paid to user-friendly interfaces for capturing URN-citable data, related by URN-citable relationships, in formats friendly to this service. The work of *Perseids*, particularly the modular *Arethusa* web-application, will be a valuable starting point.[19]

12. Abbreviations

CITE	Collections, Indices, Texts, Extensions. The digital library architecture developed for the HMT.
CTS	Canonical Text Services. A part of CITE.
HMT	Homer Multitext.
JSON	Javascript Object Notation.
OHCO2	Ordered Hierarchy of Citation Objects. An abstract model of "text".
URN	Universal Resource Name.
XML	Extensible Markups Language.

18 F. Mambrini, "Thucydides 1.89-1.118: A Multi-layer Treebank.," *CHS Research Bulletin*, vol. 1, no. 2, 2013.

19 http://www.perseids.org.

13. Bibliography

Almas, B. and Beaulieu M.-C., "Developing a New Integrated Editing Platform for Source Documents in Classics," Literary and linguistic computing, vol. 28, no. 4, pp. 493–503, 2013.

Fuhao, Z., and Jiping L., "An Algorithm of Shortest Path Based on Dijkstra for Huge Data." In 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 4:244–47, 2009. doi:10.1109/FSKD.2009.848.

Fuhao, Z., and Jiping L., "An Algorithm of Shortest Path Based on Dijkstra for Huge Data." In 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 4:244–47, 2009. doi:10.1109/FSKD.2009.848.

Guha, R. V., and Vineet Gupta. "Communicating Semantics: Reference by Description," 2015. https://research.google.com/pubs/pub44679.html.

Howard, V.A., "On Musical Quotation", Monist 58 (1974) 310.

Knuth, D. E., "Postscript About NP-Hard Problems." SIGACT News 6, no. 2 (April 1974): 15–16. doi:10.1145/1008304.1008305.

Mambrini F., "Thucydides 1.89-1.118: A Multi-layer Treebank.," CHS Research Bulletin, vol. 1, no. 2, 2013.

Mambrini, F. and Passarotti M., "Subject-Verb Agreement with Coordinated Subjects in Ancient Greek, A Treebank-Based Study." Journal of Greek Linguistics. 16:87–116, 2016.

Schubert, C. and Heyer G., "Neue Methoden der geisteswissenschaftlichen Forschung – Eine Einführung in das Portal eAQUA," Working Papers Contested Order/ eAQUA Working Papers, vol. 1, no. 0, pp. 4–9, Dec. 2010. https://doi.org/10.11588/ea.2010.0

Sokolowski, R., "Quotation." The Review of Metaphysics 37.4 (1984) : 699–723.

Smith, N., and Weaver, G., "Applying domain knowledge from structured citation formats to text and data mining: Examples using the CITE architecture," Text Mining Services (2009).

Schubert, C., "Zitationsprofile, Suchstrategien und Forschungsrichtungen," eAQUA Working Papers, vol. 1, no. 0, pp. 42–55, Dec. 2013. https://doi.org/10.11588/ea.2012.2.11409; URN (PDF): http://nbn-resolving.de/urn:nbn:de:bsz:16-ea-114092

Noto, M., and Sato, H., "A Method for the Shortest Path Search by Extended Dijkstra Algorithm." In 2000 IEEE International Conference on Systems, Man, and Cybernetics, 3:2316–20 vol.3, 2000. doi:10.1109/ICSMC.2000.886462.

14. Appendix: Ancient Homeric Analyses

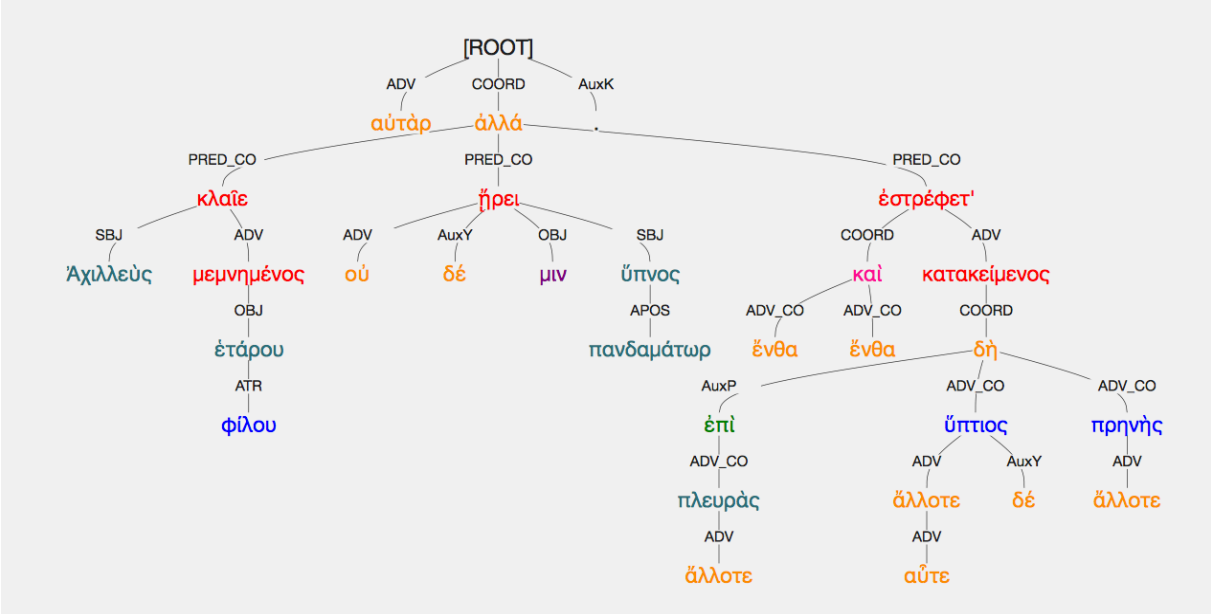

Figure 15: A treebank of *Iliad* 24.3–24.11, reading the text while omitting the lines Aristarchus athetized.

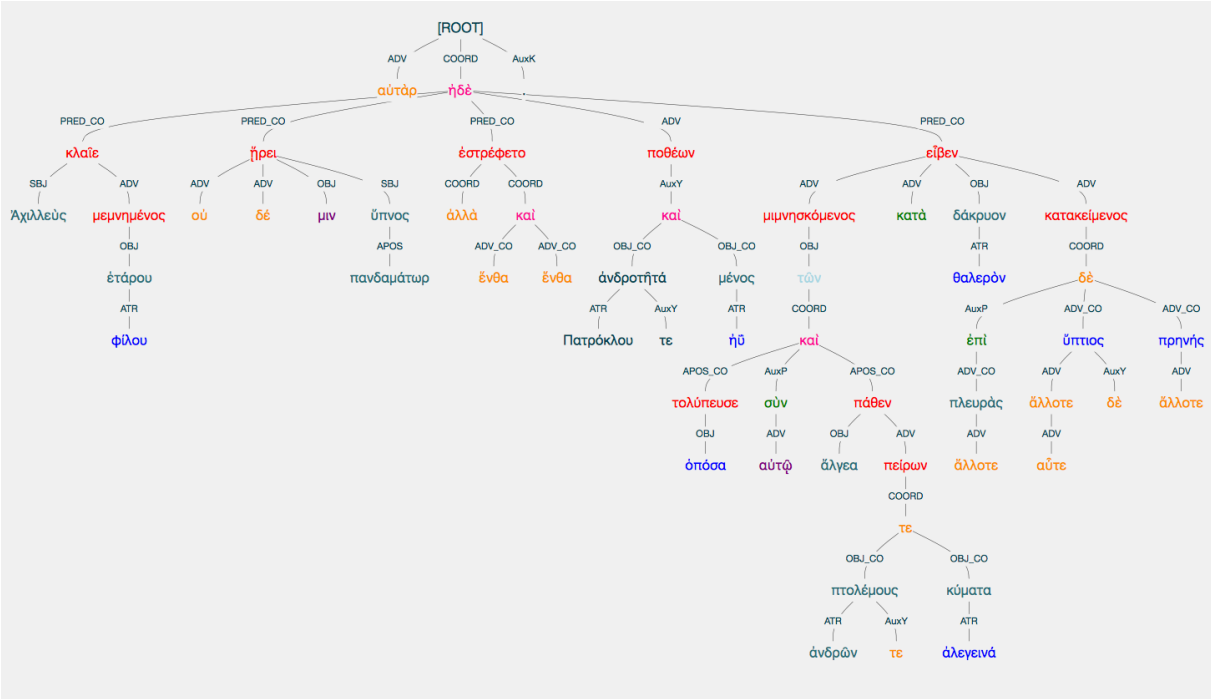

Figure 16: A treebank of *Iliad* 24.3–24.11, reading the text while including the lines Aristarchus athetized, but not following Nicanor's punctuation.

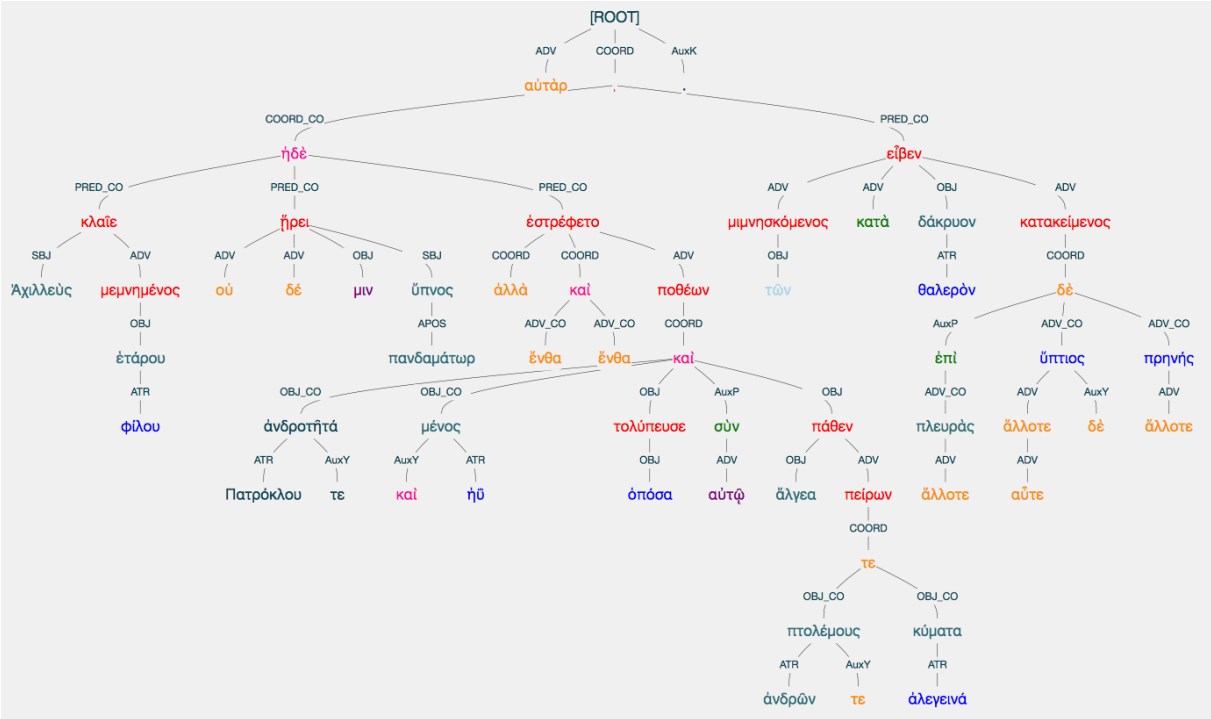

Figure 17: A treebank of *Iliad* 24.3–24.11, reading the text while including the lines Aristarchus athetized, and following Nicanor's punctuation. This is the reading the scholiast prefers, if we do not accept the athetization.

15. Autorenkontakt[20]

Christopher W. Blackwell

The Louis G. Forgione University Professor
Department of Classics
Furman University
Greenville, South Carolina, USA 29613

Email: christopher.blackwell@furman.edu

20 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 International.