Text Encoder and Annotator: an all-in-one editor for transcribing and annotating manuscripts with RDF

Fabio Valsecchi[1], Matteo Abrate[1], Clara Bacciu[1],
Silvia Piccini[2], and Andrea Marchetti[1]

[1] Institute of Informatics and Telematics (IIT) {name.surname}@iit.cnr.it,
[2] Institute for Computational Linguistics (ILC) {name.surname}@ilc.cnr.it,
National Research Council (CNR), Via G. Moruzzi 1, Pisa, Italy

Abstract. In the context of the digitization of manuscripts, transcription and annotation are often distinct, sequential steps. This could lead to difficulties in improving the transcribed text when annotations have already been defined. In order to avoid this, we devised an approach which merges the two steps into the same process. Text Encoder and Annotator (TEA) is a prototype application embracing this concept. TEA is based on a lightweight language syntax which annotates text using Semantic Web technologies. Our approach is currently being developed within the *Clavius on the Web* project, devoted to studying the manuscripts of Christophorus Clavius, an influential 16th century mathematician and astronomer.

Keywords: Manuscript Transcription, Annotation, RDF, Semantic Web

1 Introduction

Within the field of Digital Humanities, several projects are devoted to preserving, analyzing and studying the large amounts of manuscripts, books, newspapers, maps, photos and paintings stored in archives, museums and libraries around the world.

Transcription and annotation are widely used methods to make these sources accessible to a wider audience, by elucidating the context and the content of this significant cultural heritage.

In this area, we think that Semantic Web technologies, such as the Resource Description Framework (RDF), could make possible to approach text annotation in an innovative way. RDF-based annotations provide a method for enriching texts using structured data already described in details and maintained by the Semantic Web community (i.e., Linked Data sets). In addition, due to the interlinked structure of Linked Data, RDF-based annotations produce valuable annotated documents, characterized by a strong connection with external resources.

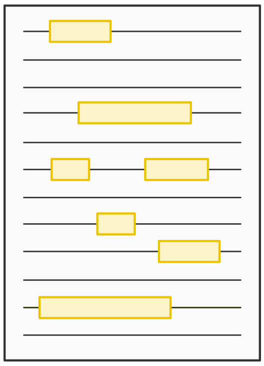
In this work we present the Text Encoder and Annotator (TEA), a web based tool for transcribing and annotating digital facsimiles of manuscripts or printed texts. It was designed as a fundamental tool for the *Clavius On the Web*[3] project [1], which aims to restore and enrich the manuscripts written by Christophorus Clavius (1538-1612), one of the most respected and influential mathematicians and astronomers of his time. Among the manuscripts analysed is his correspondence with other famous scientists of his era as well as his own scientific works. His texts *De Sphaera* or *Commentarius to Elements of Euclides* were a particular focus, since major historical figures such as René Descartes, Marin Mersenne and Johannes Kepler are known to have built their knowledge on them.

1.1 Background

In Hemminger and TerMat [4] annotations are defined as "markings (e.g., highlights) or comments made by a human agent that exist within or are attached to the text, whether in paper or digital format". In other words, annotation consists in attaching additional information such as comments, tags or links, to specific portions of a text. The process of annotating is a practice that dates back to ancient times: while studying and transcribing manuscripts, men of science tended to add explanatory notes which could take the form of interlinear or marginal glosses, brief scholia for personal purposes or postils. In the 5th century BC scholars were already adding glosses to Homer's works, and this was common practice for medieval scribes when copying biblical and legal manuscripts. The advent of the digital age has not changed this, on the contrary attention is nowadays directed to the development of tools to assist scholars in this task. Annotations can be mainly performed using two methods: *inline* and *standoff*, which reflects the ancient dichotomy between interlinear and marginal glosses, mutatis mutandis. The inline method includes annotations directly within the text, while standoff stores them separately from the text. Inline markup keeps the annotations and the annotated text close together, but it has the drawback of weighing the document down. Moreover, depending on the complexity of the markup language, the text could become hard to read. This aspect is crucial and must be taken into account when developing manual annotation tools, as users need to be able to read the annotated text with ease. Last but not least, a complex and heavyweight markup language could make the manual annotation process even more difficult for two reasons. Users have to firstly know all the syntax rules and secondly write a considerable amount of additional markup. In contrast, the standoff approach does not have markup overloading problems due to its total independence from the resource text. Annotations are in fact separately defined in a different location where the relative text offsets are stored and kept up to date. In addition, standoff markup has the advantage of allowing overlapping annotations. Nevertheless, this approach has some drawbacks related to the sequential process of transcribing and annotating. Typically, the available tools of this type separate the transcription and the annotation phases. However, if a transcription error is found during the subsequent annotation phase, it is

[3] http://claviusontheweb.it/

necessary to recompute the offsets in order to reflect the changes in the transcribed text. This implies an automatic recomputation of the offsets, a process that could be complex and costly. The logical conclusion is therefore to make transcription and annotation a joint process.



(a) Inline markup allows annotations to be included directly within the text keeping close the text fragments that have to be annotated with their annotations. A heavy markup language could make the text both heavy and hard to read. Instead a complex annotation language could make the annotation process difficult.

(b) Standoff annotations have no markup overloading problem since annotations are not stored within the text but in a different location. If transcription and annotation are treated as different steps, standoff could raise problems during the latter step if a transcription error is found.

1.2 Our Approach

In literature, several projects and initiatives have been already conducted in order to develop tools for annotating texts and also images. For instance, Pundit [3] and Refer.cx [14] provide RDF-based annotation features but limited only to web pages and without the possibility of transcribing text. Brat [13] allows text annotation supported by Natural Language Processing technology, however it does not provide a transcription feature. RDFaCE [8] is a text editor for annotating text of web pages using a graphical UI and displaying results with different views such as WYSIWYG and WYSIWYM (i.e., What You See Is What You Get/Mean). Amaya[4], developed in the W3C Annotea project [7], is a Web browser and editor for annotating text within web pages. Among the online resources provided by the collaborative initiative of Pelagios [12, 6], the Recogito tool [11] allows places to be annotated both on maps and texts. The tool provides features for marking text fragments and portions of images but

[4] https://dev.w3.org/Amaya/doc/WX/Annotations.html

the current version has been devised mainly for the annotation of places. Other examples of annotation tools can be found in the extensive survey by Uren [15].

To the best of our knowledge, none of the existing tools include an approach such the one we are proposing. From the analysis of the state of the art we conducted, it seems that there is no specific tool for managing documents (e.g., texts, images) which can assist scholars from the very beginning of the process. In particular, most of the tools lack a transcription feature and support the user only in annotation. The correct tool should support users in studying their documents by providing the digitized version of their manuscripts thus allowing for transcription and annotation.

Therefore, the core idea of this work is to combine transcription and annotation of text, thus streamlining the workflow process. Hence, we devised a lightweight language that enables this continuous and mixed process. Another key-point is that we propose to treat every textual phenomenon as an annotation independently of its specific type (e.g., semantic, syntactic, lexical). Every portion of text is treated in the same way, and RDF-based annotations are used to describe their content. RDF supports our purpose by allowing any possible annotation to be specified using the enormous amount of ontologies, vocabularies and Linked Data sets available on the Web.

2 Text Encoder and Annotator

In the light of the above, this article presents the Text Encoder and Annotator, a Web application, which provides an editor to transcribe texts as well as a lightweight language for annotating them with RDF (all within the same environment). It was envisaged for linguists, historians and more generally for scholars and students. We devised a layout composed of three main views, horizontally placed along the interface of the application (Figure 2):

1. *Image box*: it displays the digitized image of a specific manuscript to be transcribed and annotated. Zoom and pan mechanisms are available to correctly select the portion of manuscript a user is interested in.
2. *Editor box*: it is the main component of the tool. It is used to write text and enrich it with RDF-based annotations, which follow the specific language syntax described below. Text highlighting identifies the markup and improves its legibility. Moreover, a top bar contains shortcut buttons for inserting some basic annotations, characterized by common RDF predicates, such as *rdfs:seeAlso*, which can be used to link to external resources, *rdfs:comment* to specify text comments and *foaf:page* to include hyperlinks related to the topic of the annotation.
3. *Diagram box*: it is an optional view that can be activated on-demand if the user needs a summary of their annotations. The node-link diagram contains a white node representing the whole text of the document, blue nodes identifying the portion of text referred to in an annotation, orange nodes displaying the identifiers of the annotation and gray nodes describing the objects of the

RDF triples specified. The edges between orange and gray nodes represent the predicates of the triples defined in the annotations.

The tool has been implemented using different client-side libraries based on Javascript. Backbone.js[5] has been used for structuring the web application with different modules according to the Model-View-star (MV*) paradigm, the Jison library[6] for generating a parser able to analyze the lightweight language, CodeMirror[7] for including the text editor and D3.js[8] combined with Cola.js[9] for creating the summarizing visualization.

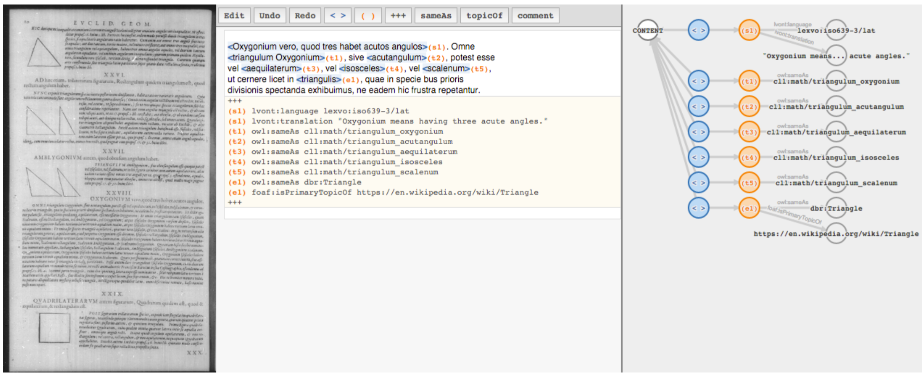


Fig. 2: The interface of the application is mainly composed of three views. From left to right there is a box containing the image of a document, an editor allowing the transcription and annotation and a diagram displaying a visual summary of the annotations. The prototype tool is available on github at http://github.com/nitaku/TEA.

2.1 Lightweight language

Considering the pros and cons of inline and standoff annotations discussed above, we think that a hybrid lightweight syntax is a suitable solution to fulfill the requirement for a simultaneous workflow of transcription and annotation.

We propose a Lightweight Markup Language[10] (LML) employed only within the interface of TEA, in order to provide an easy and quick way of transcribing and annotating text using RDF. The main reason behind the adoption of an LML is that common markup languages based on XML (e.g., TEI) are not easy

[5] http://backbonejs.org/
[6] http://zaa.ch/jison/
[7] http://codemirror.net
[8] http://d3js.org
[9] http://marvl.infotech.monash.edu/webcola/
[10] http://en.wikipedia.org/wiki/Lightweight_markup_language

to write and read in their raw form, due to their complex syntax. Moreover, the use of LMLs has already proven to be beneficial in other systems such as the Leiden-plus[11] language employed in the papyri.info editor. It is worth clarifying that we do not propose our approach as a format for the representation and interchange of texts, consequently it cannot be compared to standards such as the Text Encoding Initiative (TEI) [5]. Furthermore our language markup should be considered as distinct from semantic markup languages like Microformat [9], RDFa [2] and Microdata [10] since it has not been not devised for annotating HTML and XML documents.

Our language is used for marking portions of text and assigning them identifiers that will be used in a different, reserved and *"standoff-like"* section of the text where the details of the annotations are specified (Figure 3). More precisely, a portion of text can be annotated by enclosing it in a *span* using angle brackets, while round brackets specify a string identifying the annotation (i.e., ⟨*annotated portion of text*⟩*(identifier)*). This inline syntax uses a very limited amount of characters and does not weigh the text down too much, keeping it easy-to-read. Identifiers are then used in a distinct part of the text, called the *directive section*, where the annotation body is specified. Three plus signs (i.e., *+++*) are used both to open and close this section that can be repeated within the text more than once. Inside this block, annotations can be specified as RDF triples with the identifier of a certain span of text as subject. The choice of predicates and objects is totally free, although some default predicates are suggested in order to perform the most common and basic annotations (e.g., rdfs:seeAlso, rdfs:comment, foaf:page). Currently, the syntax used in the directive section defines triples as three text values separated by a space.

2.2 Example

We here provide an example of annotation performed on a portion of text extracted from Clavius' *Euclidis Elementorum Libri XV. Accessit XVI de solidorum regularium comparatione. Omnes perspicuis demonstrationubus, accuartisque scholiis illustrati.* [1] (paragraph 30). It is an annotated translation from Greek into Latin of Euclid's Elements. This text was considered one of the most comprehensive and authoritative of the XVI century. A free English translation of the Latin text fragment, shown in Figure 3, follows: *"It is called Oxigonium as it has three acute angles. Every Oxigonium triangle, or Acutangle triangle, could be either Equilateral, or Isosceles or Scalene as you can see from the classification provided above and not reported here"*. Figure 3 shows the code resulting from the text encoding and annotation process. Portions of text are marked using spans while the body of the annotations is specified within the directive section. The annotation identifiers (e.g., s1, t1, t2) are used as the subjects of the triples while predicates and objects are freely chosen by annotators. The Latin language (i.e., lexvo:iso639-3/lat) and the translation have been specified in the first annotation $s1$ using the Lexvo ontology[12]. Lexical entries of the

[11] http://papyri.info/editor/documentation?docotype=text
[12] http://lexvo.org/ontology

mathematical lexicon of Clavius[13] (e.g., cll:math/triangulum_oxygonium) and the DBpedia Triangle resource (i.e., dbr:Triangle) have been linked through the *seeAlso* predicate of the RDF Schema[14]. The triangle entry of Wikipedia has been specified as an interesting web page (i.e., foaf:page) for the annotation $e1$ using the FOAF vocabulary[15].

```
<Oxygonium vero, quod tres habet acutos angulos>(s1). Omne
<triangulum Oxygonium>(t1), sive <acutangulum>(t2), potest esse
vel <aequilaterum>(t3), vel <isosceles>(t4), vel <scalenum>(t5),
ut cernere licet in <triangulis>(e1), quae in speciebus prioris
divisionis spectanda exhibuimus, ne eadem hic frustra repetantur.
+++
(s1) lvont:language lexvo:iso639-3/lat
(s1) lvont:translation "..."
(t1) rdfs:seeAlso cll:math/triangulum_oxygonium
(t2) rdfs:seeAlso cll:math/triangulum_acutangulum
(t3) rdfs:seeAlso cll:math/triangulum_aequilaterum
(t4) rdfs:seeAlso cll:math/triangulum_isosceles
(t5) rdfs:seeAlso cll:math/triangulum_scalenum
(e1) rdfs:seeAlso dbr:Triangle
(e1) foaf:page https://en.wikipedia.org/wiki/Triangle
+++
```

Fig. 3: An annotated fragment of the *Euclidis Elementorum Libri XV. Accessit XVI*. Text is marked with spans highlighted in blue. Identifiers in orange are used in the directive section where they correspond to the subjects of RDF triples.

3 Conclusion & Future Works

This article describes an approach for merging the distinct steps of transcription and annotation as a single process. We implemented a tool based on a lightweight syntax language that allows RDF annotations to be performed. We conducted some preliminary tests, which involved 50 students, who were asked to use the prototype, and provide feedback. Future works will consist in developing an improved language with a syntax capable of handling nested as well as overlapping (i.e., not hierarchically nested) annotations. A preliminary analysis showed us that the syntax presented above must be slightly changed in order to treat nested and overlapping cases. These particular phenomena will be handled by writing the identifier of an annotation on both sides of the span marking the text. In

[13] http://claviusontheweb.it/lexicon/math/
[14] http://www.w3.org/TR/rdf-schema/
[15] http://xmlns.com/foaf/spec/

this way, it will not be possible to misinterpret the start and closure of a certain annotation span. New syntax operators will be introduced: *milestone* elements, for annotating a single location in the text (e.g., a gap), *partition* elements, for the identification of phenomena such as line, page or sentence breaks. Additional syntax will be introduced to provide shortcuts ("syntactic sugar") to the most common annotations. The Turtle syntax[16] will also be taken into account for the RDF triples specification. Finally, different formats (e.g., turtle, json, csv, xml) will be chosen for exporting annotations according to various data models (e.g., Open Annotation, NLP Interchange Format (NIF)).

References

1. Matteo Abrate, Angelo Mario Del Grosso, Emiliano Giovannetti, Angelica Lo Duca, Damiana Luzzi, Lorenzo Mancini, Andrea Marchetti, Irene Pedretti, and Silvia Piccini. Sharing cultural heritage: the clavius on the web project. In *Language Resources and Evaluation Conference*, 2014.
2. Ben Adida, Mark Birbeck, Shane McCarron, and Steven Pemberton. Rdfa in xhtml: Syntax and processing. *Recommendation, W3C*, 2008.
3. Marco Grassi, Christian Morbidoni, Michele Nucci, Simone Fonda, and Francesco Piazza. Pundit: augmenting web contents with semantics. *Literary and linguistic computing*, 2013.
4. Bradley M Hemminger and Julia TerMaat. Annotating for the world: Attitudes toward sharing scholarly annotations. *Journal of the Association for Information Science and Technology*, 65(11):2278–2292, 2014.
5. Nancy Ide and Jean Véronis. *Text encoding initiative: Background and contexts*. Springer Science & Business Media, 1995.
6. Leif Isaksen, Rainer Simon, Elton TE Barker, and Pau de Soto Cañamares. Pelagios and the emerging graph of ancient world data. In *Proceedings of the 2014 ACM conference on Web science*, pages 197–201. ACM, 2014.
7. José Kahan, M-R Koivunen, Eric Prud'Hommeaux, and Ralph R Swick. Annotea: an open rdf infrastructure for shared web annotations. *Computer Networks*, 39(5):589–608, 2002.
8. Ali Khalili, Sören Auer, and Daniel Hladky. The rdfa content editor-from wysiwyg to wysiwym. In *Computer Software and Applications Conference*. IEEE, 2012.
9. Rohit Khare and Tantek Çelik. Microformats: a pragmatic path to the semantic web. In *Proceedings of the 15th international conference on World Wide Web*, 2006.
10. Steven Ruggles, Matthew Sobek, Catherine A Fitch, Patricia Kelly Hall, and Chad Ronnander. *Integrated public use microdata series: Version 2.0*. Historical Census Projects, Department of History, University of Minnesota, 1997.
11. Rainer Simon, Elton Barker, Leif Isaksen, and Pau de Soto Cañamares. Linking early geospatial documents, one place at a time: Annotation of geographic documents with recogito. *e-Perimetron*, 10(2):49–59, 2015.
12. Rainer Simon, Leif Isaksen, Elton Barker, and Pau de Soto Cañamares. The pleiades gazetteer and the pelagios project, 2015.
13. Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2012.

[16] https://www.w3.org/TR/turtle/

14. Tabea Tietz, Jörg Waitelonis, Joscha Jäger, and Harald Sack. Smart media navigator: Visualizing recommendations based on linked data. In *13th International Semantic Web Conference, Industry Track*, 2014.
15. Victoria Uren, Philipp Cimiano, José Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: science, services and agents on the World Wide Web*, 4(1):14–28, 2006.