Creating an Annotated Corpus for Extracting Canonical Citations from Classics-related Texts by using Active Annotation

Matteo Romanello

King's College London
Department of Digital Humanities
26-29 Drury Lane, London WC2B 5RL
matteo.romanello@kcl.ac.uk

Abstract. This paper describes the creation of an annotated corpus supporting the task of extracting information–particularly canonical citations, that are references to the ancient sources–from Classics-related texts. The corpus is multilingual and contains approximately 30,000 tokens of POS-tagged, cleanly transcribed text drawn from the *L'Année Philologique*. In the corpus the named entities that are needed to capture such citations were annotated by using an annotation scheme devised specifically for this task.
The contribution of the paper is two-fold: firstly, it describes how the corpus was created using Active Annotation, an approach which combines automatic and manual annotation to optimize the human resources required to create any corpus. Secondly, the performances of an NER classifier based on Conditional Random Fields, are evaluated using the created corpus as training and test set: the results obtained by using three different feature sets are compared and discussed.

1 Introduction

Modern publications in Classics–such as commentaries, monographs and journal papers–contain a wealth of information that is essential to scholars. The references to primary sources therein such as inscriptions, papyri, manuscripts, archaeological findings and ancient texts, that is scholars' primary sources, all constitute meaningful entry points to information. Despite the attention that other disciplines, such as Bioinformatics, have paid to investigating the automatic extraction of information from texts from a discipline-specific perspective very little research has been done to date in the field of Classics on this topic.

This paper describes the creation of a multilingual annotated corpus to support the task of Named Entity Recognition (NER) and information extraction from Classics-related secondary sources. The corpus covers the main European languages used in Classics, namely English, French, German, Italian and Spanish: this aspect is important if one considers that Classics, like (perhaps) other

Published version: A. Gelbukh (Ed.): CICLing 2013, Part I, LNCS 7816, pp. 60–76, 2013.
http://dx.doi.org/10.1007/978-3-642-37247-6_6

Humanities disciplines, showed the tendency to preserve the use of national languages within scholarly communications rather than adopting English as *lingua franca* [1]. The named entities that were annotated in the corpus are mentions of ancient authors, titles of ancient works and references to specific texts, or parts of them–which from now on will be referred to as *canonical references*. The corpus has been released under an open license, together with the software that was used to create it, with the hope that the increased availability of tools and data will foster new research on information extraction in the field of Classics[1].

Although the characteristics of canonical citations are thoroughly examined in section 4.2, let us consider for the sake of clarity just an example: 'Homer *Iliad* 1.1' is a reference to the very first line of Homer's *Iliad* and can be alternatively expressed as 'Hom. *Il.* I 1'–note the use of abbreviations and Roman numerals– or, in an even more concise fashion, as 'A 1'. This last variant form, which uses the uppercase letters of the Greek alphabet to identify the books of the *Iliad* and the lowercase ones for the *Odyssey*, is used especially in publications with a very strong focus on the study of epics, where the number of references to the homeric poems tends to be in the order of hundreds.

The main goal in creating such a corpus is to lay the foundations for the creation of an expert system that allows one to explore any corpus of Classics-related publications by using the citations of ancient texts as search key. The first step in doing so is to devise an NER system to extract citations and other named entities of interest from texts: to train such a system is, in fact, the main purpose of this corpus. Once such canonical citations and their meanings have been extracted from texts the system uses them to create a semantic index to the texts in the corpus as well as a citation graph that can be used for information retrieval or network analysis purposes. The possible applications of such a system to Classical scholarship–and its consequent impact–become evident as soon as one thinks, for example, about intertextuality. Intertextuality research is all about studying a text in relation to a system of texts and canonical references are the way in which such texts are cited in modern publications.

The paper is organized as follows: section 2 presents a brief overview of studies that have dealt with canonical citations from a computation perspective and work done in other disciplines in relation to the extraction of discipline-specific information from texts. In section 3 the statistical model of choice to build an NER classifier and the annotation approach that was used, are introduced. Section 4 provides information about the publication from which the corpus texts were drawn, presents the challenges and limits of extracting canonical citations from texts and describes the annotation scheme that was devised to annotate the corpus. In section 5 the details of the corpus annotation process are given and finally in section 6 the evaluation of the performances of an NER system, trained with this corpus using different feature-sets, are presented and discussed.

[1] Software and data that were used to produce the results described in this paper can be found on the conference website at `http://www.cicling.org/2013/data/216`.

2 Related Work

Canonical citations–that is references to ancient texts expressed in a concise fashion–are the standard citations used by scholars in Classics to refer to primary sources (i.e. ancient texts): as such, they have never really been an object of research in this field.

However, some attention to this practice of referencing started being paid as Humanities Computing–later called Digital Humanities–began emerging and being theorized [2]. Early publications in this field immediately identified such references as a suitable target for possible applications of hypertext [3,4,5]. Indeed, the hyperlink seemed the natural way to translate citations into machine actionable links between the citing and the cited texts, a practice that would later be explored in its hermeneutical implications by McCarty [2, pp. 101-103].

More recent studies in the Digital Classics community have focussed on possible ways of representing and exploiting canonical references in a digital environment. [6] and [7] have tackled the issue of devising network protocols that allow us to share and retrieve passages of ancient texts, and related metadata, by leveraging this traditional way of referencing texts: their research lead to two different yet complementary standards, respectively the Canonical Text Services protocol[2] and a metadata format based on the OpenURL protocol[3]. Moreover, Romanello [8,9] explored new value added services for electronic publications that can be offered once the semantics of canonical citations have been properly encoded within web documents.

As large scale digitization initiatives started producing the first tangible results, the opportunity and necessity of extracting automatically named entities from texts appeared clearly. Services for the automatic extraction of named entities were deemed to be a crucial part of the emerging cyberinfrastructure for research in Classics [10]. Such named entities include not only names of people and geographical places but also some specialized entities such as canonical citations, thus confirming a tendency–observed in the field of Natural Language Processing and Information Extraction–to extend the hierarchy and number of named entities in order to capture discipline-specific information, such as proteins or genomes in bioinformatics [11, pp. 2-4]. If on this research topic there have been to date only a few studies and some preliminary results [12,13], more has been done on improving the accuracy of named entity classification and disambiguation from historical documents [14,15].

However, looking outside the domain of Classics, the automatic extraction of citations and bibliographic references from modern journal papers is a relatively well explored topic [16,17,18]. Although research on this area focussed mainly on references to modern publications the methodology they employ, such as machine-learning techniques and the feature sets used for training, can be largely applied to extraction of references to ancient texts.

[2] Homer Multitext project, `http://www.homermultitext.org/hmt-doc/`.
[3] Canonical Citation Metadata Format, `http://cwkb.org/matrix/20100922/`.

3 Methods

3.1 Conditional Random Fields

At first sight the extraction of canonical citations may seem a problem that can easily be solved by using a rule-based approach, such as that proposed by Galibert et al. [19] to extract from patents citations to other patents. The existence of standard abbreviations to refer to ancient authors and works, together with the fact that the body of classical literature is a finite set, seem to indicate that a set of hand-crafted rules can be used to extract such citations.

However, given the number of factors that can cause variations in the way canonical citations are expressed (see section 4.2 for a more detailed analysis), the compilation of a comprehensive set of rules to capture them can become a very time-consuming task. A machine-learning based system, instead, seems to offer a more scalable approach, particularly when combined with an annotation method, such as Active Annotation, which seeks to reduce the effort of producing new training data.

The supervised training method used here is a probabilistic undirected graphical model called Conditional Random Fields (CRFs). CRFs were theorized by Lafferty et al. [20] and, although they have been applied to a wider range of classification problems including computer vision and bioinformatics, they became the state-of-the-art method in sequence labeling tasks, such as Named Entity Recognition (see [21] for an introduction to CRFs and its possible applications).

The main benefit of using a model based on conditional probability, as opposed one based on joint probability, is that they can account for multiple and/or conditionally dependent features. In addition to CRFs, other supervised training methods that are well established in relation to NER tasks are Support Vector Machines (SVM) and Maximum Entropy. However, the comparison of the performances that can be achieved using different training methods, although it is undoubtedly of some interest from a technical point of view, goes beyond the scope of this paper.

Although here a C++ implementation of CRFs called CRF++[4] has been employed, implementations written in other languages are available such as CRFsuite[5] and Wapiti[6] in Python or a Java implementation that is distributed as part of the MAchine Learning for LanguagE Toolkit (Mallet)[7].

3.2 Active Annotation

Active Annotation indicates the use of the Active Learning paradigm in the specific context of creating a corpus for NLP tasks. The main idea underlying Active Learning is that the accuracy of a classifier is higher if the training examples are selected from the most informative. The more informative the training examples,

[4] CRF++, `http://crfpp.googlecode.com/`.
[5] CRFsuite, `http://www.chokkan.org/software/crfsuite/`.
[6] Wapiti, `http://wapiti.limsi.fr/`.
[7] Mallet, `http://mallet.cs.umass.edu/`.

the higher will be the performances of a model trained with them. The situations where it makes sense to use this paradigm are those where much unlabelled data can easily be obtained but labelled instances are time-consuming and thus expensive to produce. [22] contains an in-depth discussion of the differences and similarities between Active Learning and Active Annotation. To identify informative instances an uncertainty sampling method based on the least confident strategy is used [23], but other methods use entropy as uncertainty measure [24, pp. 12-26].

In order to optimize the effort of manually annotating the data the Active Annotation algorithm described in [23] was applied. This method proved to be more effective than random selection of candidates when performing an NER task on data drawn from the Humanities domain. The rationale behind it is that training data is more effective when it is drawn from those instances that are most difficult to classify for a given statistical model.

Active Annotation is an iterative process which stops when there is no further improvement in the performances between two consequent iterations. During each iteration a set of candidates for annotation is selected from the development set and, after manual correction, is added to the training set. The improvement is assessed by comparing the system performances–typically the F1 measure– before and after adding this set of candidates to the training set. An instance is added to the candidate set when the Confidence Interval (CI) for one or more of its tokens is above a given threshold.

The CI value is calculated by computing the difference between the probability of the two best labels predicted by the statistical model for a given token. Let us consider an example of how the CI is calculated. For example, the CI for the token "Philon" is 0.161877 as the two best labels outputted by the classifier have a probability respectively of 0.578489 and 0.416612.

4 Annotating and Extracting Canonical Citations

4.1 APh as Corpus

The data used to create our corpus was drawn from the *L'Année Philologique (APh)*, a critical and analytical bibliographic index of publications in the field of Classics that has been published annually since 1924. Its thorough coverage, guaranteed also by a structure based on national offices, makes the APh an essential resource for everyone who studies classical texts. To give an idea of the scale of data, the work presented in this paper uses approximately 7.5-8% of a single volume (APh vol. 75) out of the 80 volumes already published at the time of writing.

Such a huge and constantly growing body of data calls for an automatic, and thus scalable, way of extracting information from it and therefore makes it suitable for our purposes. In addition to this, what makes an annotated corpus of APh records extremely valuable is its information density. The analytical reviews in the APh are rather concise (see fig. 1) and contain a variety of references

not only to canonical texts, papyri, manuscripts and inscriptions, but also to archaeological objects, such as coins or pottery. Although for the time being our work is limited to canonical references, additional annotations for the other entity types could be added to the corpus in the future.

Example of APh abstract

(a) In **Statius' « Achilleid » (2, 96-102)** Achilles describes his diet of wild animals in infancy, which rendered him fearless and may indicate another aspect of his character - a tendency toward aggression and anger.

(b) The portrayal of angry warriors in Roman epic is effected for the most part not by direct descriptions but indirectly, by similes of wild beasts (e.g. **Vergil, Aen. 12, 101-109 ; Lucan 1, 204-212 ; Statius, Th. 12, 736-740 ; Silius 5, 306-315**).

(c) These similes may be compared to two passages from **Statius (Th. 1, 395-433 and 8, 383-394)** that portray the onset of anger in direct narrative.

Fig. 1. APh vol. 75 n. 06697: analytical review of Susanna Braund and Giles Gilbert, *An ABC of epic ira: anger, beasts, and cannibalism.*

4.2 Challenges of Extracting and Resolving Citations

The extraction of citations is modelled as a typical NER problem consisting of two sub-tasks: NE classification and NE resolutions. After having defined the basic components of a citation (see 4.3), those elements are extracted from text: this is the classification task and consists of identifying the correct entity type for each token in the text.

Once named entities and relations between them have been extracted, the entity referred to needs to be determined. For example, once the citation "Lucan 1, 204-212" has been captured one needs to determine its content, that is a reference to the text span of Lucan's *Bellum Civile* going from line 204 to line 212 of the first book.

Although the use of Latin abbreviations to refer to authors and works makes canonical citations quite regular, the language in which a given text is written introduces another cause of variation in the way they can be expressed. In a text written in German, for example, author names and work titles that can be given either in their full or abbreviated form, are likely to be in German, especially when expressed in a discursive rather than concise form.

Before describing in detail the annotation scheme that was used, let us consider the main challenges of extracting and resolving such citations. The first challenge is the **ambiguity** of citations which is often caused by the concise abbreviations that are used to refer to a given author or work. A citation such as "Th." can stand for Thucydides, *Theogonia* or *Thebaid* if considered out of its context. If, instead, the preceding mention of Statius is taken into account, 'Th." can only refer to Statius' *Thebaid*.

A second challenge is posed by the fact that canonical citations often imply **implicit domain knowledge**. In the citations "Lucan 1, 204-212" or "Silius 5, 306-315", for example, the author name is given whilst the title of the work is left implicit. The reason for this–as any scholar or student of classical texts knows–is that the work indication can be implied when one refers to the *opus maximum* of an author, which is the case for Lucan's *Bellum Civile*, or when only one work is survived or ascribable with some certainty to that author, as for Silius Italicus' *Punica*.

Another challenge is related to citations expressed in a **discursive form** (e.g. "In Statius' « Achilleid » (2, 96-102)") as opposed to a more formalized and structured one (e.g. "Statius, Th. 12, 736-740") as they require a deeper parsing of the natural language. Finally, in some cases the information is not just hard to extract because of ambiguity or discursiveness but is not even recoverable: in such case we speak of **underspecification** or underspecified references. An example of this is the use of the abbreviation "ff." in the context of a reference to mean "and the following sections/lines/verses" (e.g. "Hom. *Il.* 1,1 ff.").

4.3 Annotation Scheme

The entity types that have been annotated in the corpus are: ancient author names, ancient work titles and canonical citations. The first two entity types are marked using respectively the tags `AAUTHOR` and `AWORK`. To annotate the canonical citations two different tags were used to distinguish between their different components: `REFAUWORK` denotes the part of the citation string which contains information about the cited author and/or work, whereas `REFSCOPE` was used to capture information about the specific text passage which is being cited.

This scheme differs from the one devised by Romanello et al. [12] which had one single tag, namely `REF`, to capture the entire citation string. The reason for using different tags for different parts of the citation string lies mainly in its characteristics. The part captured by `REFAUWORK` normally contains abbreviations of author names or work title, therefore consists of alphabetic characters and punctuation, whereas the one which is captured by the tag `REFSCOPE` contains references to specific sections of a work expressed mainly by a combination of numeric characters and punctuation.

Citations, author names and work titles, however, are only some of the named entities that one can identify in the APh data, and more generally in modern publications in Classics. For example, references to papyri (e.g. "P. Hamb. 312") proved to be, during the creation of the corpus, among the entities with which canonical citations are most likely to be confused because of their very similar surface appearance. Another class of references which was not annotated in the APh corpus–mainly because of resource and time constraints–were references to literary fragments (e.g. "frr. 331-358 Kassel-Austin").

5 Active Annotation Details

In the Active Annotation phase a CRF-based classifier is used to predict the label to be assigned to each token: the probability values of the two most likely labels are needed in order to compute the Confidence Interval (CI) which, in turn, determines whether an instance should be added to the list of effective candidates or not. The CRF model was trained using the full feature set described in section 6.1 with the only exception being the Part of Speech (POS) information: it was not possible to extract the POS tags during the corpus creation phase, as explained at the end of this section, because of the tokenization method that was initially used.

Let us see now in detail the decisions that were made in applying the Active Annotation method to creating the APh corpus, given that our situation differed in some respects from that of the experiments described by Ekbal et al. [23].

A first difference is that this corpus was created from scratch and therefore some seed instances had to be selected in order to create an initial dataset for training and testing. Given the large size of the development set (7k records, ∼480k tokens) and the fact that many of its instances are *negative*–that is they do not contain the NEs in the annotation scheme–the seeds were selected partly randomly and partly manually in order to keep a reasonable balance between positive and negative instances. Therefore, some 100 instances were selected, containing approximately 6.4k tokens, out of the ∼330k tokens in the training set. For the sake of clarity, an *instance* is each of the sentences an APh abstract is made of, whereas an APh abstract is considered as a record.

The CI threshold was set to 0.2 as this was the value that lead to the best results in the experiments described by [23]. In practice, this means that for an instance to be considered an effective candidate it needs to contain one or more tokens with CI value lower or equal to 0.2.

For each round of Active Annotation, all tokens with CI over this threshold were added to the candidate set, which was then pruned in order to avoid having duplicate records–each record in the list may contain several multiple tokens that are considered effective. At this point the 30 highest scoring records in the effective candidate are sent to the annotators for manual correction of the results obtained by automatic annotation and are then added to the training set. Due to constraints, the two domain-experts that annotated the corpus worked on separate datasets and therefore the inter-annotator agreement could not be reported.

Another issue we faced relates to the size of the training and test set, and specifically to the proportion between them. The test set, obtained during the seed selection phase, had an initial size of about 2k tokens. Keeping its size fixed throughout the Active Annotation process would have lead to a disproportion between training and test set, with a consequent impact on F-score, precision and recall. The main consequence of such disproportion is the risk of overfitting the statistical model, that is training a model that will perform well on a dataset similar to the training set but will not be general enough to perform as well on a dataset with different characteristics.

This problem was solved by increasing at regular intervals the size of the test set by adding a certain number of tokens, selected using the same method. The size of the test set was increased to 4146 tokens at the beginning of round 3, and then to 6594 tokens at the start of round 7, as reported in Table 1. The table gives some details for each round of Active Annotation: what was the initial size of the training set, how many tokens were added from effective candidates, what was the initial performance of the classifier (F-score) and what the improvement (F-score gain) after the manually corrected effective candidates were added to the training set.

Table 1. Details of the Active Annotation process.

#	r	p	F	F gain	train.	test	added
1	45.45	80.65	58.14	1.51	4233	2178	2032
2	51.82	79.17	62.64	4.50	6265	2178	1968
3	55.45	75.31	63.87	1.24	8233	2178	1762
4	71.18	77.16	74.05	1.53	8027	4146	1688
5	72.06	76.47	74.20	0.15	9715	4146	2100
6	73.17	77.28	75.17	0.97	11815	4146	1433
7	71.82	70.58	71.19	1.11	13248	6594	1813
8	71.66	72.00	71.83	0.64	15061	6594	1593
9	73.73	70.69	72.17	0.35	16654	6594	1856

Time and resourcing meant we were only able to complete 9 rounds, however, the corpus had by then reached a size which made it comparable to other datasets used for similar tasks.

After round 9, the size of the corpus was ∼23k tokens, which became slightly less than ∼26k after re-tokenizing the text. The re-tokenization was needed in order to be able POS-tag the text–and then include this information in the feature set–as the whitespace-based tokenization that was initially applied proved not to be suitable for this purpose. The reason for doing this at two separate stages was the poor performances of the tokenizer when an additional list of abbreviations was not provided: this is due to the high number of abbreviations that are present in our texts and lead to a very high number of wrongly tokenized words (e.g. “Hom.” being split into “Hom” and “.”). This problem was solved by providing the tokenizer with a list of abbreviations that had been extracted from the corpus.

6 Evaluation of the NER System

6.1 Named Entity Features

Linguistics Features Since the system was designed to be language-independent, the number of linguistic features was kept to a minimum. The neighbouring

words of each token w_i in the range $w_{i-3} \dots w_{i+3}$ were considered as features, whereas experiments with using word suffixes and prefixes of length up to 4 characters showed a degradation of the performance. This may be due to the fact that this feature was extracted also for tokens containing digits and/or numbers.

The POS information of the current token is extracted automatically for all languages using TreeTagger[8] and included in the feature set without any manual correction.

Orthographic Features

- **punctuation**: this feature takes value `no_punctuation` when the token does not contain any punctuation sign at all. Otherwise it takes one of the following values: `continuing_punctuation`, `stopping_punctuation`, `final_dot`, `quotation_mark` or `has_hyphen` which is particularly important in range-like notations.
- **brackets**: when a token contains both an open and a closed parenthesis, e.g. "(10)" or "[Xen.]", the feature value is set either to `paired_round_brackets` or `paired_squared_brackets` depending on the kind of parenthesis. Similarly, when it contains either an open or a closed one, possible values are `unpaired_round_brackets` or `unpaired_square_brackets`.
- **case**: this feature is set to `all_lower` or `all_caps` when the token contains all lowercase or all uppercase characters. Other possibilities are that the token contains a mix of lower and uppercase characters (`mixed_caps`) or that only the first letter is uppercase (`init_caps`).
- **number**: three possible values of this feature are determined by the presence (`number` or `mixed_alphanum`) or absence of numeric characters (`no_digits`). The values `dot_separated_number` and `dot_separated_plus_range`, are used to identify known sequences of numbers and punctuation signs, such as "1.1.1" or "1.1.1-1.1.3", that are often found in canonical citations and particularly in the part of a citation indicating the scope of the reference.
- **pattern**: the surface similarity between tokens is captured by means of two features: a compressed pattern and an extended pattern. The former is computed by replacing lowercase characters with "`a`", uppercase ones with "`A`", numbers with "`0`" and punctuation signs with "`-`", whereas in the latter sequences of similar characters are replaced by one single pattern character.

Semantic Features Since the corpus of classical texts is a finite one, the use of semantic features should improve quite significantly the performances of our system, at least as far as the entities `AAUTHOR`, `AWORK` and `REFAUWORK` are concerned. Such features are extracted by matching each token against dictionaries of author names, work titles and their abbreviations. For the sake of performance, the dictionaries that are used for look-up are converted into a suffix array, a highly efficient data structure for indexing strings. This was implemented by using

[8] TreeTagger, `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.

Pysuffix[9], a Python implementation of Karkkainen's algorithm for constructing suffix arrays [25].

Data from the Classical Works Knowledge Base (CWKB) project [10], together with a list of canonical abbreviations that is distributed as part of the Perseus digital library[11], was used in order to create such dictionaries. CWKB in particular proved to be an essential source as it contains the canonical abbreviations and the name variants in the main European languages of 1,559 authors and 5,209 works.

Four separate features are extracted from each token to capture the fact that it is successfully matched against the author dictionary or the work dictionary: the feature takes value `match_authors_dict` or `match_works_dict` if the matching is total, whereas if the matching is partial–meaning that the token is contained in a dictionary entry, but the token length is smaller than the length of the matching entry–it is set either to `contained_authors_dict` or `contained_works_dict`.

6.2 Error Analysis

Examining the tokens that were added at each round to the list of effective candidates turned out to be extremely instructive as one can see which tokens are most problematic for the classifier, that is those with lowest CI score, for instance distinguishing canonical references from papyri references given the high surface similarity between the two.

Let us now look more closely at the performances of the classifier obtained by taking into consideration the training and test set as in the last round (n=9) of Active Annotation. The results obtained in this section were obtained by using the full feature set now including also the POS tags, as opposed to the feature set used during the candidate selection phase.

Table 2. Evaluation results aggregated by class for the last round (9) of active learning. Precision, recall and F-score are based on the number of absolute correct tags: values for the same measures, but limited to entirely correct entities are given in round brackets.

Class	**p**	**r**	**F**
AAUTHOR	57.89 (62.75)	38.60 (40)	46.32 (48.85)
AWORK	68.11 (62.20)	78.85 (72.86)	73.09 (67.11)
REFAUWORK	71.58 (71.43)	78.16 (75)	74.73 (73.17)
REFSCOPE	72.37 (66.34)	86.14 (67.68)	78.66 (67)
Overall	69.64 (65.22)	79.73 (62.28)	74.34 (63.72)

[9] Pysuffix `http://code.google.com/p/pysuffix/`.
[10] Classical Works Knowledge Base `http://cwkb.org/`.
[11] Perseus Digital Library `http://www.perseus.tufts.edu/hopper/`.

The entities on which the classifier records the worst performances are the names of ancient authors (AAUTHOR), where precision, recall and F-score are respectively 57.89%, 38.60% and 46.32%. There are two facts that emerge when looking more closely at the errors. On the one hand, the recall appears to be very low when compared to the results achieved on all other entities. On the other hand, approximately 40% of all the AAUTHOR entities retrieved by the classifier are in fact named entities, just not ancient authors. Among the errors one can find names of historical figures who in some case were also authors, such as Caesar, with the resulting issue of distinguishing contexts where someone is mentioned particularly in relation to his role of author from other contexts. Interestingly, errors in classifying such entities are less frequent when the mention of an ancient author immediately follows a canonical reference, as in the example of fig. 1, and more common when such mentions appear within a discursive context. This issue may be related to how the corpus was annotated and specifically to the fact that it is lacking a generic named entity tag for people (e.g. PERSON) who are not specifically ancient authors and also to fact that no features related to the global context are extracted.

With the exception of the performances on AAUTHOR entities, which are particularly poor as we have just seen, those on the remaining entities are pretty much in line, with F-score values respectively of 73.09%, 74.73% and 78.66%. The identification and classification of titles of ancient works mentioned in the text (AWORK) presented at least two issues. The first one concerns the number of false positives and explains the relatively low precision: in the APh data, title of works are normally given between French quotation marks (i.e. « and »), also known as Guillemets, but they are also used for honorific titles, concepts, and the like, all cases where in other styles scare quotes are used instead. This is due to the style which is adopted throughout the publications and leads to some ambiguity and relative problems of classifications. A second issue is the number of errors in identifying the boundaries of such entities, which is evident when comparing the F-score on absolute correct tags with the F-score calculated on correct entities: this may be related to feature set, and specifically to the fact that no specific features were used to capture multiword names.

6.3 Evaluation

As has been stated above, there were two main motivations for creating such an annotated corpus: the lack of both datasets and software for this specific kind of NER and the intuition that the bigger the corpus the more representative the results of evaluation achieved when using such a corpus as training and test set. Therefore, the evaluation was carried out by using different feature sets, so to have at least a baseline to be used for comparison. Moreover, the 10-fold cross-validation performed on chunks of the corpus of varying size largely confirmed by empirical evidence the initial intuition as explained below.

Firstly, we performed the cross-validation on the whole corpus (size = 28,893 tokens) by using three different feature sets: the results are given in table 4. With the first set, considered as the baseline and consisting solely of POS tags as

Table 3. Break-down of the evaluation results by class. The results are relative to the last round (n=9) of active learning.

Class	TP	FP	FN	Tot. retr.	Total	p	r	F
B-AAUTHOR	29	18	50	47	79	0.62	0.37	0.46
I-AAUTHOR	15	14	20	29	35	0.52	0.43	0.47
B-AWORK	49	31	20	80	69	0.61	0.71	0.66
I-AWORK	171	72	39	243	210	0.70	0.81	0.75
B-REFAUWORK	29	12	11	41	40	0.71	0.73	0.72
I-REFAUWORK	39	15	8	54	47	0.72	0.83	0.77
B-REFSCOPE	78	23	21	101	99	0.77	0.79	0.78
I-REFSCOPE	239	98	30	337	1269	0.71	0.89	0.79
O	6389	165	249	6554	6638	0.97	0.96	0.97

features, precision, recall and F-score of respectively 66.34%, 42.22% and 51.12% were achieved. The second feature set used includes POS tag information as well as the wide range of orthographic features: with this feature set an improvement of respectively +11.75%, +21.61% and +18.37% on precision, recall and F-score was registered. Yet the highest scores were obtained when using the full feature set, which consists of the previous two plus the semantic features. The precision, recall and F-score obtained when using this third feature set were respectively 79.85%, 69.07% and 73.44%. It is noteworthy that the use of semantic features– typically indicating whether a given token matches successfully against one or more dictionaries or lexica–does not always lead to an improvement of the overall performances, as observed by [26].

Table 4. Results of the 10-fold cross-validation using the whole corpus (25104 tokens).

Feature Set	r	p	F
POS	42.22	66.34	51.12
POS+ortho	63.83	78.09	69.49
POS+ortho+sem	69.07	79.85	73.44

Secondly, an analogous 10-fold cross-validation was performed but on chunks of the corpus of varying size: the purpose of this evaluation experiment was to verify the correlation between the results and the size of training and test set. In total 10 iterations were performed: in the first one only 10% of the corpus was considered, then for each new iteration another 10% was added until in the 10^{th} and last iteration the entire corpus was used. For training the full feature set was used, that is POS tag information, orthographic and semantic features.

The results of this experiment are plotted in fig. 2. The first pattern it is possible to observe is the gradual convergence of precision and recall as the size of train and test sets increase. A similar pattern can also be found in the accuracy

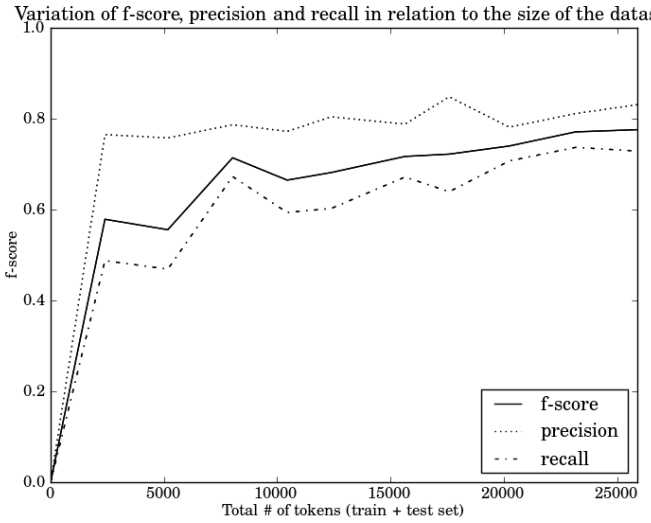

Fig. 2. This diagram shows how accuracy measures vary in relation to the size of the dataset. F-score, precision and recall were calculated using a 10-fold cross-validation on data chunks of regularly increasing size.

scores that were registered during the corpus creation as reported in Table 1. Another, and more interesting, phenomenon that was observed is the spike in performances when using less than 50% of the whole corpus for both training and testing. A similar, although not identical, pattern can also be found in round n=6 of Active Annotation, where the highest F-score value was measured. What this shows is that the reason for relatively high measure of precision, recall and F-score sometimes is to be found in the size of the dataset used.

7 Conclusions and Further Work

The significance of the work presented in this paper lies mainly in that it starts to fill the lack of datasets and software for discipline-specific NER on modern texts related to Classics. The evaluation results themselves–although encouraging, particularly when considering that were obtained on a multilingual corpus–were not entirely satisfactory. However, our main hope in releasing both the dataset and the software under an open license is to fuel new research on this topic, so that better results can eventually be achieved.

In addition to the gap that this resource fills, it makes possible to perform with greater accuracy basic yet essential steps of text processing such as tokenization, POS tagging and sentence segmentation. The main reason for this is that texts in this field contain a high number of abbreviations that are not common in other fields. Canonical references, in fact, are essentially an abbreviation

of the title of the cited work followed by the indication of the citation scope. As a result, a list of abbreviations can easily be extracted from the corpus and then supplied to the tokenizer or POS tagger. Similarly, sentence breaks were manually identified in the corpus texts, thus making possible the use of this corpus to train more accurate sentence tokenizers.

Furthermore, the corpus annotation could be improved and extended in several ways. A first basic enhancement would be to manually correct the POS tags since they were assigned in an unsupervised fashion. Second, the layer of named entity annotations could be extended in order to improve the overall performances by introducing a generic class for those entities that do not fall into any specific category, as suggested by the evaluation results presented in section 6.3. Other named entities that could be annotated in the corpus include references to papyri, manuscripts and fragmentary texts. Third, the layers of syntactic annotation and anaphoric annotation could be added to the existing ones: this would allow for a deeper language parsing and therefore it would make possible to capture those citations that are expressed in a more discursive fashion.

Finally, further work is currently being carried out in the following main directions: 1) the use of other supervised learning methods, in addition to CRF, in order to compare the results obtained; 2) the disambiguation and co-reference resolution of the automatically extracted named entities; 3) the comparison and evaluation of the results that are obtained by applying a model trained with cleanly transcribed texts on a corpus of potentially noisy OCRed documents.

Acknowledgments

The author wants to gratefully thank Prof. Eric Rebillard for giving him access to the APh data and for his precious help in annotating the corpus, as well as the anonymous reviewers of the paper for their very helpful comments.

References

1. Mimno, D.: Computational Historiography : Data Mining in a Century of Classics Journals. ACM Transactions on Computational Logic (2005) 1–19
2. McCarty, W.: Humanities Computing. Palgrave Macmillan (2005)
3. Crane, G.: From the old to the new: intergrating hypertext into traditional scholarship. In: Proceedings of the ACM conference on Hypertext, Chapel Hill, North Carolina, United States, ACM (1987) 51–55
4. Bolter, J.D.: The Computer, Hypertext, and Classical Studies. The American Journal of Philology **112** (1991) 541–545
5. Bolter, J.D.: Hypertext and the Classical Commentary. In: Accessing antiquity : the computerization of classical studies. University of Arizona Press, Tucson (1993) 157–171
6. Ruddy, D., Rebillard, E.: Text Linking in the Humanities: Citing Canonical Works Using OpenURL (2009)
7. Smith, N.: Digital Infrastructure and the Homer Multitext Project. In Bodard, G., Mahony, S., eds.: Digital Research in the Study of Classical Antiquity. Ashgate Publishing, Burlington, VT (2010) 121–137

8. Romanello, M.: New Value-Added Services for Electronic Journals in Classics. JLIS.it **2** (2011)
9. Romanello, M.: A semantic linking framework to provide critical value-added services for E-journals on classics. In Mornati, S., Chan, L., eds.: ELPUB2008. Open Scholarship: Authority, Community, and Sustainability in the Age of Web 2.0 - Proceedings of the 12th International Conference on Electronic Publishing held in Toronto, Canada 25-27 June 2008. (2008) 401–414
10. Crane, G., Seales, B., Terras, M.: Cyberinfrastructure for Classical Philology. Digital Humanities Quarterly **3** (2009)
11. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes **30** (2007) 3–26
12. Romanello, M., Boschetti, F., Crane, G.: Citations in the digital library of classics: extracting canonical references by using conditional random fields. In: Proceedings of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries. NLPIR4DL '09, Morristown, NJ, USA, Association for Computational Linguistics (2009) 80–87
13. Romanello, M., Thomas, A.: The World of Thucydides: From Texts to Artefacts and Back. In Zhou, M., Romanowska, I., Zhongke, W., Pengfei, X., Verhagen, P., eds.: Revive the Past. Proceeding of the 39th Conference on Computer Applications and Quantitative Methods in Archaeology. Beijing, 12-16 April 2011, Amsterdam University Press (2012) 276–284
14. Smith, D.A., Crane, G.: Disambiguating Geographic Names in a Historical Digital Library. Lecture Notes in Computer Science (2001) 127–136
15. Babeu, A., Bamman, D., Crane, G., Kummer, R., Weaver, G.: Named Entity Identification and Cyberinfrastructure. In Kovács, L., Fuhr, N., Meghini, C., eds.: Research and Advanced Technology for Digital Libraries. Springer (2007) 259–270
16. Kramer, M., Kaprykowsky, H., Keysers, D., Breuel, T.: Bibliographic Meta-Data Extraction Using Probabilistic Finite State Transducers (2007)
17. Councill, I.G., Giles, C.L., Kan, M.y.: ParsCit: An open-source CRF Reference String Parsing Package. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odjik, J., Piperidis, S., Tapias, D., eds.: Proceedings of LREC. Number 3, Citeseer, European Language Resources Association (ELRA) (2008) 661–667
18. Kim, Y.M., Bellot, P., Faath, E., Dacos, M.: Automatic annotation of incomplete and scattered bibliographical references in Digital Humanities papers. In Beigbeder, M., Eglin, V., Ragot, N., Géry, M., eds.: CORIA. (2012) 329–340
19. Galibert, O., Rosset, S., Tannier, X., Grandry, F.: Hybrid Citation Extraction from Patents. In Chair, N.C.C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D., eds.: Proceedings of the Seventh conference on International Language Resources and Evaluation LREC10, European Language Resources Association (ELRA) (2010)
20. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Brodley, C.E., Danyluk, A.P., eds.: Machine Learning International Workshop then Conference. Number Icml in ICML '01, Citeseer (2001) 282–289
21. Sutton, C., McCallum, A.: An Introduction to Conditional Random Fields for Relational Learning. In Getoor, L., Taskar, B., eds.: Introduction to Statistical Relational Learning. Number x. MIT Press (2006)
22. Vlachos, A.: Active annotation. Proceedings of the Workshop on Adaptive Text Extraction and Mining (ATEM 2006) (2006) 64–71

23. Ekbal, A., Bonin, F., Saha, S., Stemle, E., Barbu, E., Cavulli, F., Girardi, C., Poesio, M.: Rapid Adaptation of NE Resolvers for Humanities Domains using Active Annotation. Journal for Language Technology and Computational Linguistics **26** (2011) 39–51
24. Settles, B.: Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)
25. Kärkkäinen, J., Sanders, P., Burkhardt, S.: Linear work suffix array construction. Journal of the ACM **53** (2006) 918–936
26. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: JNLPBA '04: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Morristown, NJ, USA, Association for Computational Linguistics (2004) 104–107