Journal of Data Mining and Digital Humanities

Developing an annotator for Latin texts using Wikipedia

Raffaele Guarasci[1*]

1 University of Salerno, Italy

*Corresponding author: Raffaele Guarascirguarasci@unisa.it

Abstract
This work investigates the feasibility of using Wikipedia as a resource for annotations of Latin texts. Although Wikipedia is an excellent resource from which to extract many kinds of information (morphological, syntactic and semantic) to be used in NLP tasks on modern languages, it was rarely applied to perform NLP tasks for the Latin language. The work presents the first steps of the developement of a POS Tagger based on the Latin version of Wiktionary and a Wikipedia-based semantic annotator.

keywords
Part-of-speech tagger; Latin language; Python; Wikipedia annotator

INTRODUCTION

In recent years the huge amount of data made freely available from web resources – in particular Wikipedia –have been used for several Natural Language Processing (NLP) tasks, ranging from information extraction [Wu and Weld, 2010], ontology and taxonomy population [Ponzetto and Strube, 2007] to knowledge representation [Zesch et al., 2007; 2008] and semantic tasks [Witten and Milne, 2008]. Despite the proliferation of these works on modern language, currently Wikipedia has never been applied for Latin language tasks. For this reason, the idea underlying the work is to use the huge amount of information provided by Wikipedia as a source to develop an annotator that performs two kinds of analysis: a morphosyntactic analysis (part-of-speech tagging) and a semantic annotation. PoS-Tagging is a long-standing NLP task and a wide variety of approaches for the Latin language has been proposed in recent years.They are worthy to be noticed TreeTagger[1] [Schmid, 1994], based on decision trees, Tnt [Brants, 2000], based on HiddenMarkov model, Lapos [Tsuruoka et al., 2011], builtusingmaxiumentropyMarkov model, e OpenNLPTagger[2], thatcombinesdifferenttaggingmethod, and Stanford Taggerr [Toutanova et al., 2003].Semanticannotationinsteadis a largelyunexploredtasksregarding the Latin language. Our work is divided in two steps. In the first step we develop a part-of-speech tagger prototype by mining words and POS tags from the Latin version of Wiktionary[3] (the freely available multilingual dictionary of Wikipedia). Adding a small set of features to these extracted data we build the core of a weakly supervised PoS-tagger.The second step of the work consists in the development of a Wikipedia-based semantic annotator. The idea is to use the pages of Latin version of Wikipedia to perform a pattern matching and annotate texts linking them to pertinent Wikipedia pages. This kind of approach allows to obtain richly annotated Latin texts, taking the full advantage of the possibilities offered by Wikipedia

[1]http://www.cis.uni-muenchen.de/ ˜schmid/tools/TreeTagger/.
[2]https://opennlp.apache.org/
[3] https://en.wiktionary.org/wiki/Index:Latin

database. Notice that our goal is not to provide a tool comparable in efficiency with the most used PoS-taggers used in literature mentioned above, rather than to offer a supporttool for the morpho-syntactic and semanticannotationable to work in real time exploitingonlyavailableweb resources and repositories. In otherwordswefocused on buildinga simple and fast annotator thatisroughlyasgood.Followingtheseguidelineswedecided to operate a linguisticsimplification, notdistinguishingbetweendifferentvarieties of Latin (Classical Latin, Vulgar Latin, late Latin…), otherwiseitwouldnot be possible to use a source like Wikipedia in an effective way.

METHDOLOGY

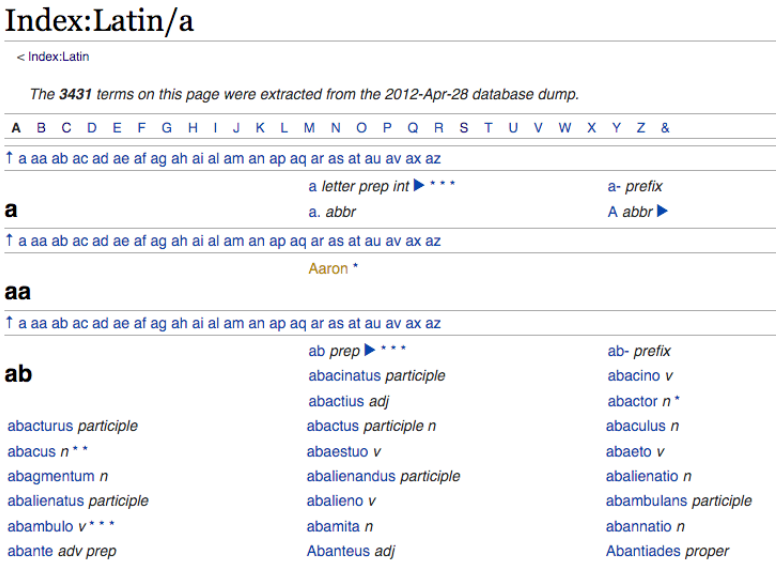
The work is composed bytwo steps: the development of a Part of Speech tagger prototype mining the Latin version of Wikipedia Wiktionary and the development of a semantic annotator based on latin Wikipedia pages. In a broad way the method consists of exploiting the huge amount of information that Wikipedia(and in particular Wiktionary) can provide. These information range from part-of-speech tags for words that appears in the dictionary to word frequencies and statistical information.

1POS Tagger

The proposed method starts creating a corpus by mining the Latin version of Wiktionary to extract words and their corresponding part-of-speech tags, then use these data to create a lexicon of pos-tagged words.In this phase we have chosen to use Wikipedia only for the population of annotated dictionary. However, for the construction of the PoS-tagger we chose to use a machine learning algorithm based on averaged perceptron, due this kind of approachguarantees a solid and fast code, which allows to obtain good results using an almost completely unsupervised approach. Currently, being the work still in progress, we have not introduced morphosyntactic rules, nevertheless the algorithm already shows encouraging results, although on a test data sample.

1.1 Scraping Wiktionary

The Latin version of Wiktionary list a huge amount of Latin words manually annotated by Wiktionary contributors. The Wiktionary is an excellent starting point, since each provided lemma is associated to its part of speech tag.


Index:Latin/a

< Index:Latin

*The **3431** terms on this page were extracted from the 2012-Apr-28 database dump.*

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z &

↑ a aa ab ac ad ae af ag ah ai al am an ap aq ar as at au av ax az

a

a *letter prep int* ▶ * * *
a. *abbr*
a- *prefix*
A *abbr* ▶

↑ a aa ab ac ad ae af ag ah ai al am an ap aq ar as at au av ax az

aa

Aaron *

↑ a aa ab ac ad ae af ag ah ai al am an ap aq ar as at au av ax az

ab

ab *prep* ▶ * * *
abacinatus *participle*
abactius *adj*
ab- *prefix*
abacino *v*
abactor *n* *
abacturus *participle*
abactus *participle n*
abaculus *n*
abacus *n* * *
abaestuo *v*
abaeto *v*
abagmentum *n*
abalienandus *participle*
abalienatio *n*
abalienatus *participle*
abalieno *v*
abambulans *participle*
abambulo *v* * * *
abamita *n*
abannatio *n*
abante *adv prep*
Abanteus *adj*
Abantiades *proper*

Figure 1 : Snapshot of a Latin version of Wiktionary

Journal of Data Mining and Digital Humanities http://jdmdh.episciences.org
ISSN 2416-5999, an open-access journal

Since Wiktionary's data are free we started scraping source code of web pages containing entries of the dictionary using `urllib2`[4]and`Beautifulsoup`[5], two modules well known in literature written in Python language. Mining the Latin section of Wiktionary, we collected approximately 30000 words, each of them labelled with corresponding part-of-speech tag. Notice thath Wiktionary uses its own specific tagest, based on abbreviations. Despite this tagset differs from most used ones in the literature, we decided to keep them,in line with the purpose of the tool. Moreover, it is still possible mapping abbreviations to another tagset in a future work. There is another important aspect to note, especially from a linguistic point of view. Wiktionary, by its nature of "collaborative project to produce a free-content multi-lingual dictionary"[6] does not distinguish between terms belonging to different variants of Latin language (Classical Latin, Late Latin...), by contrast it consideres the Latin lexicon as an *unicum*, a big bag of Latin words.

1.2 *Training the POS Tagger*

Since this is a work still in development we have chosen to leave the lexicon obtained by Wiktionary scraping in reasonably raw form, filtering only data without part-of-speech tag or multiword expressions (e.g. *arsbellica*, *silentiumestaurum…*). After that, following the approach proposed by [De Smedt et al., 2014]we used a simple Python script that navigate Wiktionary hyperlinks to retrieve inflected forms of words contained in the lexicon. On these data, we trained a machine learning algorhitm based on averaged perceptron to build our PoS-Tagger prototype. We started from an open-source implementation of an Averaged Perceptron tagger, focused on speed and developed for English Language[7]. We chose a Perceptron-based tagger due their robustness and ease of adaption and implementation in different contexts [Collins, 2002]. Furthermore, this approach allows us to use a very small set of language-dependent features, making our method an "extremely weakly supervised PoS-tagger".Since this work is still under developement, currently our prototype has been tested on a test sample of 10000 words extracted from ColLex.LA corpus[8]. In both cases, the accuracy is around 80%, it can be seen as an encouraging result because almost any grammatical or morphosyntactic feature has been taken into account yet. Our accuracy is significantly lower than other studies reported in the literature [Bamman and Crane, 2008; Passarotti, 2010; Lee et al., 2011; Muller and Schutze, 2015], but it should be considered that often these works are not comparable, because they differ both for different variants of Latin analysed, both for sizes and annotation standards used in corpora, as noted in[Eger et al., 2015], in addition all these approaches start from a hand-built resources, not from automatically web extacted data.

2 SEMANTIC ANNOTATOR

As mentionaed above, the second step of the work is the creation of a Wikipedia-based semantic annotator, in other words a system that perform a pattern matching linking PoS-tagged relevant terms to corresponding Wikipedia pages. Our idea is to enrich the PoS-tagged texts with a semantic layer. Systems albe to annotate texts on-the-fly exploiting several online

[4]https://pymotw.com/2/urllib2/
[5]http://www.crummy.com/software/BeautifulSoup/
[6]http://www.wiktionary.org/
[7]https://github.com/sloria/textblob-aptagger
[8] http://collex.hucompute.org/

resources are already well known in literature [Ferragina and Scaiella 2010; Meij et al., 2012, Leskovec et al., 2009]. In our case, we settled for a smaller goal. At this early stage of the work we focused on recognize and link proper nouns in Latin language, sort of Named Entity Recognition task. First we need to recognize which PoS-tagged entities are proper nouns, so we chosed a rough approach.A proper noun can be: an entity tagged with *proper* label in the corpus built from Wiktionary or each word starts with capital letter.Even if it seems a coars simplification, it can give an idea of the criterion underlying the annotator. We wrote a quite simple crawler in Python language that performs a query on Google[9] and thennavigateVicipædia[10](the Latin version of Wikipedia) to match nouns with corresponding webpages.

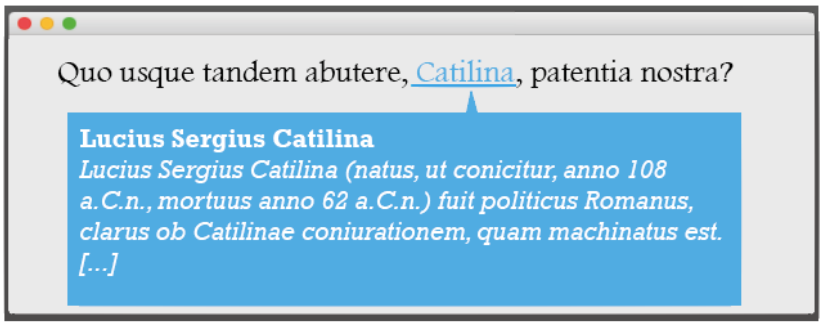


Figure 2 : An example taken from the demo version that shows a snippet resulting from Google and Vicipædia query for the relevant term of the text

To improve the accuracy of the system we used an edit distance algorhitm, that allows us to perform partial matches. This function is particulary useful for Latin language, as it is rich in graphical variants (*Cæsar/Caesar*, *Rabanus/Hrabanus*…).

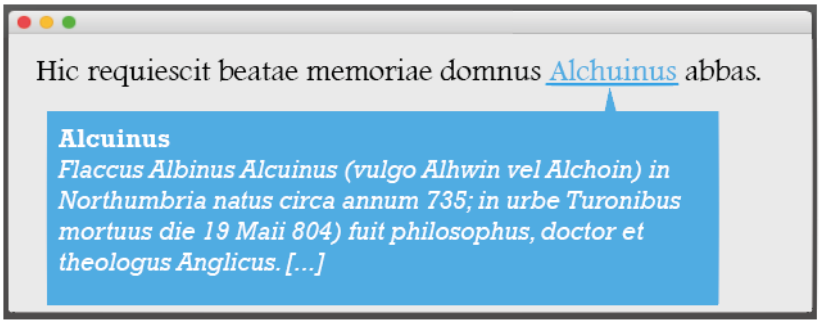


Figure 3 : An example of a graphical variation matched using Edit Distance

3 OPEN ISSUES AND FUTURE WORK

Here we presented the first steps of the development of a real-time morphosyntactic and semantic annotator for Latin language. The core feature of the proposed approach is that it is completely based on freely avalaible web resources, such as the Latin version of Wikipedia and Wiktionary. Although the work is still at an early stage, first tests have highlighted some interestgin results.The tool is still to be considered in an alpha stage, since at this stage only

[9] https://pypi.python.org/pypi/google
[10] https://la.wikipedia.org

the core of the system was built, there are many open issues. In future implementationslanguage-based features, a set of morphosyntactics rulesand the possibility to manage multiword expressions will be added, in order to have a more comprehensive system and to increase accuracy.

References

Bamman, David, and Gregory Crane. "Building a dynamiclexicon from a digitallibrary." *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2008.

Brants, Thorsten. "TnT {a statistical part-ofspeechtagger. In Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)." *Seattle, WA* (2000).

Collins, Michael. "Discriminative training methods for hiddenmarkovmodels: Theory and experiments with perceptronalgorithms." *Proceedings of the ACL-02 conference on Empiricalmethods in naturallanguage processing-Volume 10*. Association for ComputationalLinguistics, 2002.

De Smedt, Tom, et al. "Using Wiktionary to Build an Italian Part-of-Speech Tagger." *Natural Language Processing and Information Systems*. Springer International Publishing, 2014. 1-8.

Eger, Steffen, Tim vorderBrück, and Alexander Mehler. "Lexicon-assistedtagging and lemmatization in Latin: A comparison of sixtaggers and twolemmatizationmethods." *LaTeCH 2015* (2015): 105.

Ferragina, Paolo, and Ugo Scaiella. "Tagme: on-the-flyannotation of short text fragments (by wikipediaentities)." *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010.

Lee, John, Jason Naradowsky, and David A. Smith. "A discriminative model for joint morphologicaldisambiguation and dependencyparsing." *Proceedings of the 49th Annual Meeting of the Association for ComputationalLinguistics: Human Language Technologies-Volume 1*. Association for ComputationalLinguistics, 2011.

Leskovec, Jure, Lars Backstrom, and JonKleinberg. "MemeTracker: tracking news phrase over the web." (2009).

Meij, Edgar, Wouter Weerkamp, and Maarten de Rijke. "Addingsemantics to microblogposts." *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012.

Müller, Thomas, and HinrichSchütze. "Robustmorphologicaltagging with word representations." *Proceedings of NAACL*. 2015.

Passarotti, Marco. "Leavingbehind the less-resourced status. the case of latin through the experience of the indexthomisticustreebank." *7th SaLTMiL Workshop on Creation and use of basiclexicalresources for less-resourcedlanguages LREC 2010, Valetta, Malta, 23 May 2010 Workshop programme*. 2010.

Ponzetto, Simone Paolo; Strube, Michael. Deriving a large scale taxonomy from Wikipedia. In: *AAAI*. 2007. p. 1440-1445.

Schmid, Helmut. "Probabilistic part-of-speechtaggingusingdecisiontrees."*Proceedings of the international conference on new methods in language processing*. Vol. 12. 1994.

Toutanova, Kristina, et al. "Feature-rich part-of-speechtagging with a cyclicdependency network." *Proceedings of the 2003 Conference of the North American Chapter of the Association for ComputationalLinguistics on Human Language Technology-Volume 1*. Association for ComputationalLinguistics, 2003.

Tsuruoka, Yoshimasa, YusukeMiyao, and Jun'ichiKazama. "Learning with lookahead: can history-basedmodelsrivalgloballyoptimizedmodels?."*Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for ComputationalLinguistics, 2011.

Witten, Ian; Milne, David. An effective, low-costmeasure of semanticrelatednessobtained from Wikipedia links. In: *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an EvolvingSynergy, AAAI Press, Chicago, USA*. 2008. p. 25-30.

Wu, Fei; Weld, Daniel S. Open information extractionusing Wikipedia. In:*Proceedings of the 48th Annual Meeting of the Association for ComputationalLinguistics*. Association for ComputationalLinguistics, 2010. p. 118-127.

Zesch, Torsten; Gurevych, Iryna. Analysis of the Wikipedia categorygraph for NLP applications. In: *Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007)*. 2007. p. 1-8.

Zesch, Torsten; Muller, Christof; Gurevych, Iryna. ExtractingLexicalSemantic Knowledge from Wikipedia and Wiktionary. In: *LREC*. 2008. p. 1646-1652.