Developing an annotator for Latin texts using Wikipedia

Raffaele Guarasci

▶ **To cite this version:**

Raffaele Guarasci. Developing an annotator for Latin texts using Wikipedia. Journal of Data Mining and Digital Humanities, In press. hal-01279853v2

HAL Id: hal-01279853
https://hal.science/hal-01279853v2

Submitted on 30 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

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 Guarasci rguarasci@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 development 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 Hidden Markov model, Lapos [Tsuruoka et al., 2011], built using maxium entropy Markov model, e OpenNLPTagger[2], that combines different tagging method, and Stanford Tagger [Toutanova et al., 2003]. Semantic annotation instead is a largely unexplored tasks regarding 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 semi-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 database.

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

METHDOLOGY

The work is composed by two 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.

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 support tool for the morpho-syntactic and semantic annotation able to work in real time exploiting only available web resources and repositories. Our goal is to build a framebork easy to use, implement, extend and share; a baseline ready to be tailored to different tasks fully based on freely available data. Every part of the tool (algorhitms, dataset, training and test set, etc.) will be released on a public repository under a MIT Licence. Following these guidelines, we decided to operate some simplification, in particular concerning linguistic aspects. At this early stage, the tool does not operate distinction between different varieties of Latin (Classical Latin, Vulgar Latin, late Latin…), otherwise it would not be possible to use a source like Wikipedia in an effective way.

1 POS 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 approach guarantees 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.

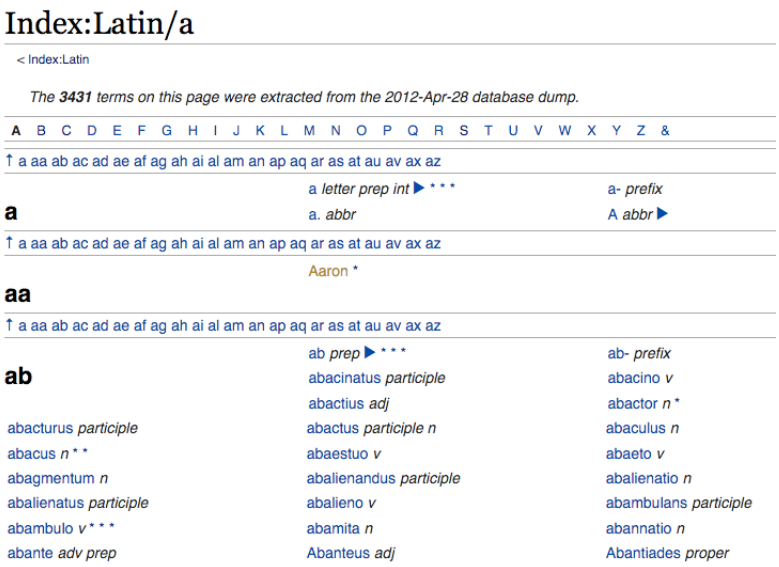

Figure 1 : Snapshot of a Latin version of Wiktionary

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 that 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 considers the Latin lexicon as an *unicum*, a big bag of Latin words. Although this can be seen as a limit to this type of approach, it looks like an acceptable compromise for the development of a tool aiming to be released freely under open source licence since it is completely based on frelly availebe web resources.

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. *ars bellica*, *silentium est aurum*...). 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 algorithm 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

[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

[Collins, 2002]. Collins' perceptron algorithm has several advantages: it is simple to implement, it has short training times, and it apparently improves its performance with increasing amount of data. According to [Passarotti, Dell'Orletta, 2010] the development of resources and tools for less-resourced languages like the historical ones can benefit from the exploitation of language-independent tools and methods developed over the years by many projects for modern languages. In particular, using a machine-learning approach like averaged perceptron allows us to use a very small set of language-dependent features, making our method a lightweight semi-supervised PoS-tagger. This kind of perceptron-based approach requires a large amount of supervised (manual-annotated) data to achieve good performances. This aspect could be a weak point, in order to overcome this limitation, we decided to exploit the great amount of pre-annotated data available on Wikipedia. Since this work is still under development, 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 extracted data.

2 SEMANTIC ANNOTATOR

As mentioned 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 allow to annotate texts on-the-fly exploiting several online 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 chose 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 coarse 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 then navigate Vicipædia[10] (the Latin version of Wikipedia) to match nouns with corresponding webpages.

[8] http://collex.hucompute.org/
[9] https://pypi.python.org/pypi/google
[10] https://la.wikipedia.org

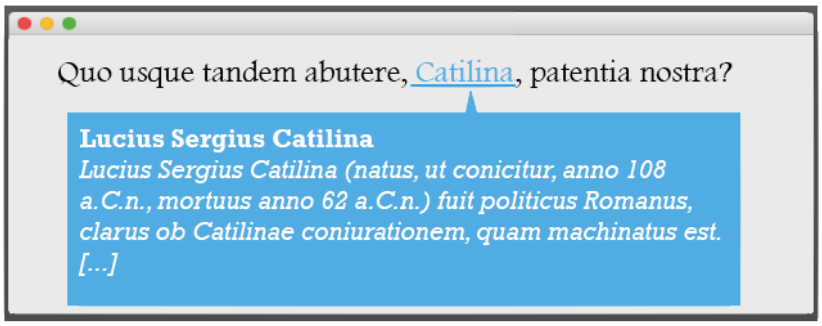

Figure 2 : An example taken from the demo version that shows a snippet resulting from Google and Vicipædia query for the relevant term of the text

To improve the accuracy of the system we used an alignment algorithm, following an approach already used in the literature for the handling of ancient texts [Boschetti, 2007]. These kinds of algorithms evaluate the similarity of any string with another string or part of it, so they work well concerning term variant extraction tasks [Jacquemin, Tzoukermann, 1999; Weller et al., 2011]. They are based on an implementation of Levenshtein distance [Yujian, Bo, 2007], that compare costs to perform additions, subtractions, substitutions and transpositions of blocks in order to transform the first string in the second one or in a part of it. Following this principle, any chunk of text is aligned with the portion of text where the distance is lowest (*i.e.* the similarity is highest) [Boschetti, 2007]. This function is particularly useful for Latin language, as it is rich in graphical variants (*Cæsar/Caesar*, *Rabanus/Hrabanus*…). Indeed, Latin has no standard orthography, (in most cases the spelling has not been normalized by the editor, but remains as it is in the text) it can mean that the same word may appear spelled differently throughout texts.

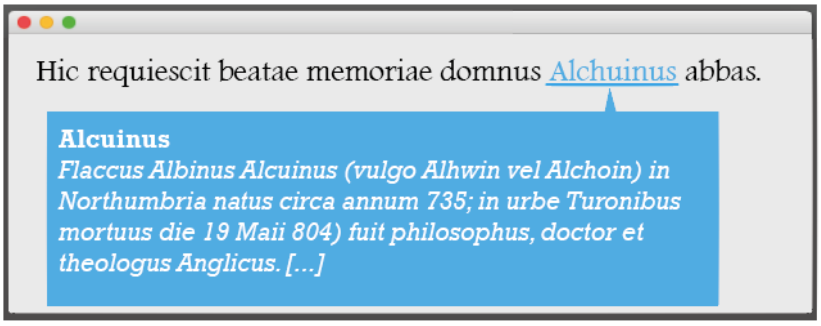

Figure 3 : An example of a graphical variation matched using Edit Distance

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 the core of the system was built, there are many open issues. In future implementations language-based features, a set of morphosyntactics rules and 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 dynamic lexicon from a digital library." *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2008.

Boschetti, Federico. "Methods to extend Greek and Latin corpora with variants and conjectures: Mapping critical apparatuses onto reference text." *Proceedings of the Corpus Linguistics Conference (CL2007)*. 2007.

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

Collins, Michael. "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, 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 vor der Brück, and Alexander Mehler. "Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization methods." *LaTeCH 2015* (2015): 105.

Ferragina, Paolo, and Ugo Scaiella. "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)." *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 morphological disambiguation and dependency parsing." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.

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

Yujian, Li, and Liu Bo. "A normalized Levenshtein distance metric." *IEEE transactions on pattern analysis and machine intelligence* 29.6 (2007): 1091-1095.

Jacquemin, Christian, and Evelyne Tzoukermann. "NLP for term variant extraction: synergy between morphology, lexicon, and syntax." *Natural language information retrieval*. Springer Netherlands, 1999. 25-74.

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

Müller, Thomas, and Hinrich Schütze. "Robust morphological tagging with word representations." *Proceedings of NAACL*. 2015.

Passarotti, Marco. "Leaving behind the less-resourced status. the case of latin through the experience of the index thomisticus treebank." *7th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages LREC 2010, Valetta, Malta, 23 May 2010 Workshop programme*. 2010.

Passarotti, Marco, and Felice Dell'Orletta. "Improvements in parsing the Index Thomisticus treebank. revision, combination and a feature model for medieval Latin." *Training* 2 (2010): 61-024.

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

Schmid, Helmut. "Probabilistic part-of-speech tagging using decision trees."*Proceedings of the international conference on new methods in language processing*. Vol. 12. 1994.

Toutanova, Kristina, et al. "Feature-rich part-of-speech tagging with a cyclic dependency network." *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, 2003.

Tsuruoka, Yoshimasa, Yusuke Miyao, and Jun'ichi Kazama. "Learning with lookahead: can history-based models rival globally optimized models?."*Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2011.

Weller, Marion, et al. "Simple methods for dealing with term variation and term alignment." *9th International Conference on Terminology and Artificial Intelligence (TIA 2011)*. 2011.

Witten, Ian; Milne, David. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In: *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA*. 2008. p. 25-30.

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

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

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