Proceedings of 1st Workshop on Language Technologies for Historical and Ancient Languages, pages 119–123
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

A Gradient Boosting-Seq2Seq System for Latin POS Tagging and Lemmatization

Giuseppe G. A. Celano
Leipzig University
Augustusplatz 10, 04109 Leipzig
celano@informatik.uni-leipzig.de

Abstract

The paper presents the system used in the EvaLatin shared task to POS tag and lemmatize Latin. It consists of two components. A gradient boosting machine (LightGBM) is used for POS tagging, mainly fed with pre-computed word embeddings of a window of seven contiguous tokens—the token at hand plus the three preceding and following ones—per target feature value. Word embeddings are trained on the texts of the Perseus Digital Library, Patrologia Latina, and Biblioteca Digitale di Testi Tardo Antichi, which together comprise a high number of texts of different genres from the Classical Age to Late Antiquity. Word forms plus the outputted POS labels are used to feed a Seq2Seq algorithm implemented in Keras to predict lemmas. The final shared-task accuracies measured for Classical Latin texts are in line with state-of-the-art POS taggers (∼96%) and lemmatizers (∼95%).

Keywords: Latin, gradient boosting, Seq2Seq, POS tagging, lemmatization, treebank

1. Introduction

The EvaLatin shared task (Sprugnoli et al., 2020) consists of two NLP tasks, (coarse-grained) POS tagging and lemmatization, each of which can be addressed in two modalities, closed and open.

Closed modality does not allow use of annotated external resources, such as treebanks or lexica, while non-annotated resources, such as word embeddings, can be used. In open modality, use of any external resource is allowed.

Participation to the shared task in closed modality only is possible, the open-modality approach being optional. The Latin texts provided for training are 7,[1] and belong to different works (see Table 1).

author	work	tokens
Caesar	Bellum Civile	6,389
	Bellum Gallicum	44,818
Cicero	Philippica	52,563
Plinius Secundus	Epistulae	50,827
Seneca	De Beneficiis	45,456
	De Clementia	8,172
Tacitus	Historiae	51,420

Table 1: Training data

The Latin data differ in age (slightly) and genre, because the goal of the shared task is to evaluate how models perform not only on similar, but also different, kinds of text. Caesar's and Tacitus' works are historical accounts, Cicero's Philippica are speeches, Plinius' work consists in letters, while Seneca's are philosophical essays. Caesar (100 BC–44 BC) and Cicero (106 BC–43 BC) belong to the Golden Age, while Plinius (61 AD–c. 113 AD), Seneca (c. 4 BC–65 AD), and Tacitus (c. 56 AD–c. 120 AD) belong to the Silver Age.

The released data are provided in the conllu format, with sentence split and tokenization/word segmentation already performed. It is to note that the organizers decided to remove punctuation marks and to not tokenize enclitic *que* (i.e., "and"), although it usually is, in Latin treebanks, on syntactic grounds. As a consequence, tokenization/word segmentation could also be easily accomplished from raw text by splitting on whitespaces.[2]

Each token is aligned with only POS and lemma labels according to the Universal Dependencies (UD) scheme (Zeman et al., 2019).[3] As is known, the UD scheme provides general definitions for its morphosyntactic labels, in that they are supposed to be used for annotation of many typologically different languages.

There are currently three different UD Latin treebanks,[4] which use the same morphosyntactic labels slightly differently. For example, there is no consensus on whether a substantivized adjective should be morphologically annotated as an adjective or a noun (which will affect also lemma annotation), or how to treat, for example, "ubi" ("where") without an antecedent: is it a relative adverb or a subordinate conjunction? Unfortunately, there are many such problematic cases, still inadequately covered in guidelines. Notably, they cause not only divergencies between different treebanks, but also, often, inconsistencies within a treebank

[1] `https://circse.github.io/LT4HALA/EvaLatin.html`.

[2] Identifying enclitic *que* is probably the main word segmentation problem for Latin, because of its high frequency and the fact that a high number of other words end in non-enclitic *que*, such as, for example, *quisque*, *quicumque*, or *aeque*. While almost all of these can be identified via rule-based algorithms, the series of tokens *quique*, *quaeque*, and *quodque* cannot: these word forms signify both pronouns ("everyone") and relative pronouns + enclitic *que*, and therefore can be disambiguated only by considering their syntactic contexts.

[3] See also, more specifically, `https://universaldependencies.org/guidelines.html`.

[4] See `https://universaldependencies.org/`. The UD Latin treebanks derive from conversion of similarly annotated treebanks (Celano, 2019).

itself, annotators getting easily confused.[5]

For this reason, I decided to participate only to the closed modality of the shared task, by proposing a two-step system (see Figure 1) which employs (i) a gradient boosting machine for POS tagging and (ii) a Seq2Seq algorithm leveraging POS labels for lemmatization.[6] I present the former in Section 3 and the latter in Section 4. In Section 2, I discuss the pre-computed word embeddings which feed the gradient boosting machine, while Section 5 contains some concluding remarks.

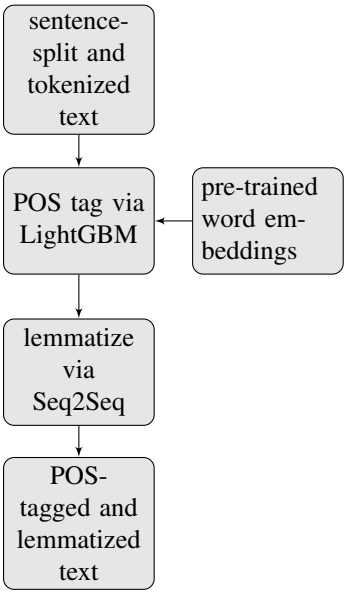

Figure 1: System pipeline

2. Data preparation and pre-computed word embeddings

Each text of the released data has been divided into three sets: training (80%), development (10%), and test (10%). By the union of all the training, development, and test sets, the final training, development, and test sets to use for machine learning have been created. This splitting strategy guarantees that each final set is, with respect to the data released, balanced and representative.

Token order within a sentence has been preserved because, as is shown in Section 3, preceding and following tokens of any given token has been used to predict the POS of such a given token. Order of sentences has also been kept because it is assumed to be irrelevant for the purposes of the machine learning task at hand.

Word embeddings are a common way to vectorize word forms. In recent years, FastText (Bojanowski et al., 2016) has emerged as a successful library for word representation. Differently from other word embedding algorithms, such as Word2vec (Goldberg and Levy, 2014), FastText represents words as the sum of character n-grams, thus allowing any prefixes, infixes, or suffixes to be weighted.

Some models for Latin, such as the one based on texts from Common Crawl and Wikipedia, have already been computed and are freely available.[7] However, since the data released for the shared task are literary texts without punctuation, a new model trained exclusively on punctuation-free literary texts from sources derived from high quality digitization and post-scan processing is probably expected to perform better than less specific—even if already available—models.

I therefore trained a model using the texts from the Perseus Digital Library (PDL),[8] Patrologia Latina (PL),[9] and Biblioteca Digitale di Testi Tardo Antichi (BDTTA).[10] As the shared task also aims to evaluate a model on texts of different (i) age and (ii) genre, using the above mentioned collections, which together comprise most of the existing pre-medieval Latin texts, guarantees that both variables are adequately represented.

Another most crucial reason to create a new model is that the released data adopts the convention of only using the grapheme "u" to represent both the Latin vocalic (/u/) and semivocalic (/w/) phonemes. As is known, editors of Latin texts adopt different conventions in this respect, and therefore non-normalized texts are very likely to generate underperforming models for the shared task at hand.

FastText requires raw text as an input. Its extraction from the annotated XML files of especially the PDL is a non-trivial task, which would require a separate study. The texts of the PDL, as well as those of the PL and BDTTA, follow the Epidoc Schema, which is a subset of the TEI schema. An original text is interspersed with a lot of "markup text" introduced by XML elements such as "del"—to signal that a certain word should be deleted—or "note"—to add a comment on a specific point in the text.

The PDL texts also represent a particular challenge because some of them cannot be parsed by XML parsers:[11] indeed, a number of externally defined character references, such as "ē", raise exceptions, and therefore require preprocessing.

After extracting the text from the above mentioned collections and converting all "v" into "u",[12] I trained a model through FastText with the following hyperparameters: skip-gram mode, minimum length of char n-gram 2, maximum length of char n-gram 5, dimensions 300, and learning rate

[5] A solution for this are more precise guidelines and word lists to account for specific phenomena, such as `https://git.informatik.uni-leipzig.de/celano/latinnlp/-/tree/master/guidelines` and `https://git.informatik.uni-leipzig.de/celano/latinnlp/blob/master/tokenize/to-tokenize.xml`.

[6] The models are made available at `https://github.com/gcelano/evalatin2020`.

[7] `https://fasttext.cc/docs/en/crawl-vectors.html`.

[8] `https://github.com/PerseusDL/canonical-latinLit/tree/master/data`.

[9] `https://github.com/OpenGreekAndLatin/patrologia_latina-dev/tree/master/corrected`.

[10] `http://digiliblt.lett.unipmn.it/g_bulk_opere.php`.

[11] I used the Java SAXParser, available in BaseX 9.3.1.

[12] I did not lowercase the texts, because I did not verify that this improves accuracy.

0.03.[13]
The model so created outperformed the Latin model provided by FastText in a number of preliminary tests. I also experimented with a lot of different hyperparameters and even texts: it is worth mentioning that models relying on the PHI Latin texts[14] turned out to perform worse than the one based on the above mentioned collections, probably because the PHI Latin texts comprise a considerable number of fragmentary works, whose texts mainly consist of broken words.

3. LightGBM: a powerful gradient boosting machine

LightGBM (Ke et al., 2017) is an efficient gradient boosting machine which combines high accuracies, fast training speed, and easy of use. It is developed by Microsoft, and has so far been successfully employed for a high number of different machine learning challenges.
Two kinds of features are employed to predict the POS labels:[15] (i) word embeddings and (ii) 2-character token endings. Word embeddings are calculated for any given token and its three preceding and three following tokens. Position is always calculated within a given sentence: if no token precedes or follows, a vector of 0 is used. Similarly, 2-character endings of any of the above mentioned tokens are extracted and made independent variables—if no token precedes or follows an underscore is used. Vectorization for the endings is automatically performed by LightGBM.
After some experimenting, the following hyperparameter values turned out to be optimal: boosting_type = 'gbdt', num_leaves = 50, max_depth = -1, learning_rate = 0.03, n_estimators = 47946, subsample_for_bin = 100000, objective = 'multiclass', class_weight = None, min_split_gain = 0.0, min_child_weight = 0.001, min_child_samples = 1, subsample = 1.0, subsample_freq = 0, colsample_bytree = 1.0, reg_alpha = 0, reg_lambda = 0.001, random_state = 1, importance_type = 'split', max_bin = 500.

tagger	test accuracy	time
LightGBM	96.2	>3h
Marmot	95.18	31.9s
Lapos	95.22	18.78s

Table 2: Taggers compared

As Table 2 shows, the test accuracy of LightGBM[16] is higher than those of two popular taggers, Lapos (Tsuruoka et al., 2011) and Marmot (Mueller et al., 2013), which have been used with default hyperparameters. Striking is, however, training time, in that both Lapos and Marmot are extremely fast and do not require any pre-computed word embeddings. On the other hand, LightGBM required a very high number of estimators (47,946) in order to get about 1% more accuracy than the other taggers. This therefore discouraged me, after finding the hyperparameters, from re-training the model with the train set + development set. With more training data (which could even include the test set), a winning accuracy for the shared task might have been achieved.
The LightGBM development accuracy calculated is 96.39%, while the test accuracy is 96.2%. These values are very similar to the final one calculated for Classical Latin on the shared task test set (95.52%). These accuracies are in line with state-of-the-art POS taggers for Classic Latin (Gleim et al., 2019).[17] As expected, the shared task cross-genre and cross-time accuracies calculated are lower (see Table 3).

classical	cross-genre	cross-time
95.52	88.54	83.96

Table 3: Final accuracy scores for POS tagging

4. A Seq2Seq algorithm for lemmatization

Lemmatization is the NLP task aiming to associate a group of morphologically associated word forms to one of these word forms which is conventionally taken as representative of the entire group.
Lemmas usually concide with dictionary entries. However, since dictionaries adopt slightly different conventions and sometimes are even inconsistent in themselves, there are a number of open issues, such as, for example, whether an adverb should be lemmatized with its related adjective.
To solve the lemmatization task, I adopt the Seq2Seq algorithm implemented in Keras.[18] It is a popular algorithm often employed for machine translation. It can be easily applied to the lemmatization task, in that lemmatization can be interpreted as a case of translation from a word form to another.
The algorithm allows translation on a character level. It consists of a LSTM layer functioning as an encoder, whose internal states are exploited by another LSTM layer, a decoder, to predict the target sequence.
In order to facilitate prediction, a target lemma is associated with a word form plus its POS label generated by LightGBM. POS labels are expected to disambiguate between morphologically ambiguous word forms.
The following hyperparameters were used: batch size 64, epochs 10, latent dimensions 2500. The development set accuracy calculated is 99.82%, while the test set accuracy is 97.63%. The accuracy calculated on the shared task test set is 94.6%. The drops in accuracy are arguably due to both some overfitting and the fact that the POS labels used for the test sets were not the gold ones, but those predicted by LightGBM, which therefore contained errors (see Table 4 for all final shared task accuracy scores).
One issue which was met when decoding some input tokens of the test data released for the shared task is that some Greek words in it contained a few Greek characters not

[13] Refer to the documentation for more details on hyperparameters: `https://fasttext.cc/docs/en/options.html`
[14] `https://latin.packhum.org/`.
[15] Morphological features are not required in EvaLatin.
[16] I checked that the POS tag assigned to a Greek word or "lacuna" is always "X", as required by the shared task guidelines.
[17] See also, for example, "la_proiel" at `https://universaldependencies.org/conll18/results-upos.html`.
[18] `https://keras.io/examples/lstm_seq2seq/`.

present in the training data. I had to substitute them with some Greek characters belonging to the set of those used in the training phase. This was not an issue at all, however, in that the lemma for any Greek word is always the placeholder "uox_graeca". Moreover, any "lacuna" in the text (i.e., any token including more than one period), which is always associated with "uox_lacunosa", has been automatically assigned the right lemma via a rule-based script.

An unsolved problem is caused by Arabic numbers: they are not present in the training data provided, and therefore it is not clear what lemma labels should be predicted.

classical	cross-genre	cross-time
94.6	81.69	83.92

Table 4: Final accuracy scores for Lemmatization

5. Conclusion

The paper has shown a two-component system to POS tag and lemmatize Latin. The first consists in a LightGBM algorithm predicting POS labels from word embeddings and 2-character endings of a given token plus its three preceding and following tokens. The algorithm returns accuracies ($\sim$96%) in line with those of state-of-the-art POS taggers for Classical Latin. The POS labels outputted plus word forms are then used to feed a Keras Seq2Seq algorithm, whose final result calculated on the shared task test set for Classical Latin (94.6%) can also be considered highly comparable to state-of-the-art lemmatizers (for example, the 1st ranked lemmatizer scored 95.9%, i.e., -1.3%).

6. Acknowledgements

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation; project number 408121292).

7. Bibliographical References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

Celano, G. G. A. (2019). The Dependency Treebanks for Ancient Greek and Latin. *Digital Classical Philology*, pages 279–98.

Gleim, R., Eger, S., Mehler, A., Uslu, T., Hemati, W., Lücking, A., Henlein, A., Kahlsdorf, S., and Hoenen, A. (2019). Practitioner's View: A Comparison and a Survey of Lemmatization and Morphological Tagging in German and Latin. *Journal of Language Modelling*, 7(1):1–52.

Goldberg, Y. and Levy, O. (2014). Word2vec Explained: Deriving Mikolov et al.'s Negative-sampling Word-Embedding Method. *arXiv preprint arXiv:1402.3722*.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LighGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

Mueller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging". In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA, October. Association for Computational Linguistics.

Sprugnoli, R., Passarotti, M., Cecchini, F. M., and Pellegrini, M. (2020). Overview of the EvaLatin 2020 Evaluation Campaign. In Rachele Sprugnoli et al., editors, *Proceedings of the LT4HALA 2020 Workshop - 1st Workshop on Language Technologies for Historical and Ancient Languages, satellite event to the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Paris, France, May. European Language Resources Association (ELRA).

Tsuruoka, Y., Miyao, Y., and Kazama, J. (2011). Learning with Lookahead: Can History-Based Models Rival Globally Optimized Models? In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 238–246. Association for Computational Linguistics.

8. Language Resource References

Zeman, D., Nivre, J., Abrams, M., Aepli, N., Agić, Ž., Ahrenberg, L., Aleksandravičiūtė, G., Antonsen, L., Aplonova, K., Aranzabe, M. J., Arutie, G., Asahara, M., Ateyah, L., Attia, M., Atutxa, A., Augustinus, L., Badmaeva, E., Ballesteros, M., Banerjee, E., Bank, S., Barbu Mititelu, V., Basmov, V., Batchelor, C., Bauer, J., Bellato, S., Bengoetxea, K., Berzak, Y., Bhat, I. A., Bhat, R. A., Biagetti, E., Bick, E., Bielinskienė, A., Blokland, R., Bobicev, V., Boizou, L., Borges Völker, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Boyd, A., Brokaitė, K., Burchardt, A., Candito, M., Caron, B., Caron, G., Cavalcanti, T., Cebiroğlu Eryiğit, G., Cecchini, F. M., Celano, G. G. A., Čéplö, S., Cetin, S., Chalub, F., Choi, J., Cho, Y., Chun, J., Cignarella, A. T., Cinková, S., Collomb, A., Çöltekin, Ç., Connor, M., Courtin, M., Davidson, E., de Marneffe, M.-C., de Paiva, V., de Souza, E., Diaz de Ilarraza, A., Dickerson, C., Dione, B., Dirix, P., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eckhoff, H., Eli, M., Elkahky, A., Ephrem, B., Erina, O., Erjavec, T., Etienne, A., Evelyn, W., Farkas, R., Fernandez Alcalde, H., Foster, J., Freitas, C., Fujita, K., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Gerdes, K., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., González Saavedra, B., Griciūtė, B., Grioni, M., Grūzītis, N., Guillaume, B., Guillot-Barbance, C., Habash, N., Hajič, J., Hajič jr., J., Hämäläinen, M., Hà Mỹ, L., Han, N.-R., Harris, K., Haug, D., Heinecke, J., Hennig, F., Hladká, B., Hlaváčová, J., Hociung, F., Hohle, P., Hwang, J., Ikeda, T., Ion, R., Irimia, E., Ishola, Ọ., Jelínek, T., Johannsen, A., Jørgensen, F., Juutinen, M., Kaşıkara, H., Kaasen, A., Kabaeva, N., Kahane, S., Kanayama, H., Kanerva, J., Katz, B., Kayadelen, T., Kenney, J., Kettnerová, V., Kirchner, J., Klementieva, E., Köhn, A., Kopacewicz, K., Kotsyba, N., Kovalevskaitė, J., Krek, S., Kwak, S., Laippala, V., Lambertino, L., Lam, L., Lando, T.,

Larasati, S. D., Lavrentiev, A., Lee, J., Lê H`ông, P., Lenci, A., Lertpradit, S., Leung, H., Li, C. Y., Li, J., Li, K., Lim, K., Liovina, M., Li, Y., Ljubešić, N., Loginova, O., Lyashevskaya, O., Lynn, T., Macketanz, V., Makazhanov, A., Mandl, M., Manning, C., Manurung, R., Mărănduc, C., Mareček, D., Marheinecke, K., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., McGuinness, S., Mendonça, G., Miekka, N., Misirpashayeva, M., Missilä, A., Mititelu, C., Mitrofan, M., Miyao, Y., Montemagni, S., More, A., Moreno Romero, L., Mori, K. S., Morioka, T., Mori, S., Moro, S., Mortensen, B., Moskalevskyi, B., Muischnek, K., Munro, R., Murawaki, Y., Müürisep, K., Nainwani, P., Navarro Horñiacek, J. I., Nedoluzhko, A., Nešpore-Bērzkalne, G., Nguy˜ên Thị, L., Nguy˜ên Thị Minh, H., Nikaido, Y., Nikolaev, V., Nitisaroj, R., Nurmi, H., Ojala, S., Ojha, A. K., Olúòkun, A., Omura, M., Osenova, P., Östling, R., Øvrelid, L., Partanen, N., Pascual, E., Passarotti, M., Patejuk, A., Paulino-Passos, G., Peljak-Łapińska, A., Peng, S., Perez, C.-A., Perrier, G., Petrova, D., Petrov, S., Phelan, J., Piitulainen, J., Pirinen, T. A., Pitler, E., Plank, B., Poibeau, T., Ponomareva, L., Popel, M., Pretkalniņa, L., Prévost, S., Prokopidis, P., Przepiórkowski, A., Puolakainen, T., Pyysalo, S., Qi, P., Rääbis, A., Rademaker, A., Ramasamy, L., Rama, T., Ramisch, C., Ravishankar, V., Real, L., Reddy, S., Rehm, G., Riabov, I., Rießler, M., Rimkutė, E., Rinaldi, L., Rituma, L., Rocha, L., Romanenko, M., Rosa, R., Rovati, D., Roșca, V., Rudina, O., Rueter, J., Sadde, S., Sagot, B., Saleh, S., Salomoni, A., Samardžić, T., Samson, S., Sanguinetti, M., Särg, D., Saulīte, B., Sawanakunanon, Y., Schneider, N., Schuster, S., Seddah, D., Seeker, W., Seraji, M., Shen, M., Shimada, A., Shirasu, H., Shohibussirri, M., Sichinava, D., Silveira, A., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Soares-Bastos, I., Spadine, C., Stella, A., Straka, M., Strnadová, J., Suhr, A., Sulubacak, U., Suzuki, S., Szántó, Z., Taji, D., Takahashi, Y., Tamburini, F., Tanaka, T., Tellier, I., Thomas, G., Torga, L., Trosterud, T., Trukhina, A., Tsarfaty, R., Tyers, F., Uematsu, S., Urešová, Z., Uria, L., Uszkoreit, H., Utka, A., Vajjala, S., van Niekerk, D., van Noord, G., Varga, V., Villemonte de la Clergerie, E., Vincze, V., Wallin, L., Walsh, A., Wang, J. X., Washington, J. N., Wendt, M., Williams, S., Wirén, M., Wittern, C., Woldemariam, T., Wong, T.-s., Wróblewska, A., Yako, M., Yamazaki, N., Yan, C., Yasuoka, K., Yavrumyan, M. M., Yu, Z., Žabokrtský, Z., Zeldes, A., Zhang, M., and Zhu, H. (2019). Universal Dependencies 2.5. LINDAT/CLARIAH-CZ Digital Library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.