Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022), pages 189–192
Language Resources and Evaluation Conference (LREC 2022), Marseille, 25 June 2022

An ELECTRA Model for Latin Token Tagging Tasks

Wouter Mercelis, Alek Keersmaekers
KU Leuven / Brepols Publishers - CTLO, KU Leuven
KU Leuven: Blijde-Inkomststraat 21, B-3000 Leuven, Belgium
Brepols Publishers - CTLO: Begijnhof 39, B-2300 Turnhout, Belgium
{wouter.mercelis, alek.keersmaekers}@kuleuven.be

Abstract

This report describes the KU Leuven / Brepols-CTLO submission to EvaLatin 2022. We present the results of our current small Latin ELECTRA model, which will be expanded to a larger model in the future. For the lemmatization task, we combine a neural token-tagging approach with the in-house rule-based lemma lists from Brepols' ReFlex software. The results are decent, but suffer from inconsistencies between Brepols' and EvaLatin's definitions of a lemma. For POS-tagging, the results come up just short from the first place in this competition, mainly struggling with proper nouns. For morphological tagging, there is much more room for improvement. Here, the constraints added to our Multiclass Multilabel model were often not tight enough, causing missing morphological features. We will further investigate why the combination of the different morphological features, which perform fine on their own, leads to issues.

Keywords: ELECTRA, lemmatization, POS-tagging, morphological tagging, morphological features, token tagging

1. Introduction

This short report describes the systems developed by the KU Leuven / Brepols-CTLO team for the EvaLatin 2022 Evaluation Campaign. The first section will describe the language model that is used in all three tasks. Subsequently, the three tasks (lemmatization, POS-tagging and morphological tagging) are discussed, each divided in subsections concerning the followed methodology, the results and a discussion of these results.

2. Language Model

We pretrained a custom Latin ELECTRA-model[1] (Clark et al., 2020), using Brepols' Library of Latin Texts[2] as training data (160M tokens). ELECTRA models maintain the same basic computational architecture as BERT models (Devlin et al., 2018). While they are computationally less expensive, they nevertheless achieve better results, due to a more efficient training approach. This makes them particularly suited to training models with comparatively less amounts of data. In the future, we will train a larger Latin ELECTRA model with more training data, continuing the pioneering work of Bamman and Burns' Latin-BERT (Bamman and Burns, 2020).

3. Lemmatization

3.1. Methodology

For the lemmatization task, we combined a rule-based gazetteer approach (in which handcrafted rules provide lists of possible word forms for each lemma) with a neural token tagging task. Using a rule-based approach, Brepols provided a system (ReFlex) that generates all possible forms for each lemma in their database. As a first step in our lemmatization system, ReFlex returns for each token in the lemmatization task the corresponding lemmata. If there is only one possibility, no further action is needed. Otherwise, we predict the POS-tag of the token as described in the next section, and use this POS-tag to resolve the existing ambiguity, returning the lemma with the matching POS-tag. For the remaining ambiguous tokens, we had to make a pragmatic decision, as it is not feasible to train a separate classifier for each of the remaining tokens. Therefore, we trained one classifier on choosing the right lemma out of the list of possible lemmata that ReFlex returned, using the Huggingface Transformers implementation of ElectraForTokenClassification[3]. For example, if ReFlex returned 3 possible lemmata for a token, e.g. two nouns and a verb, we would assign them the labels n1, n2 and v1 respectively. The task of the classifier consists of predicting which label is needed in the current context, and thus returning the right lemma. This is not an optimal solution, as there is no linguistic reason why a certain lemma would be first or second in the ReFlex list. However, this approach is needed to make a decision between, for example, two or three nouns, as the disambiguation based on the POS-tag is impossible in this scenario. Based on the validation data, our approach was successful concerning nouns, but fails when faced with multiple verbs as possible lemmata. Lastly, a few manual rules were written based on a run on validation data, for example converting abbreviated praenomina to their spelled out counterparts.

[1] In the future, our pretrained Latin ELECTRA-model will be uploaded to Huggingface Transformers.

[2] See Brepols' Library of Latin Texts.

[3] For this specific implementation, see ElectraForTokenClassification on Huggingface Transformers

In the same vein, ReFlex returned the original adjective when processing an adjectival adverb, while the EvaLatin dataset expects the adjectival adverb itself as the predicted lemma. We adopted the following rule to circumvent this problem: if the POS-tag is ADV, ReFlex does not return its normal lemma, but the associated adverb.

3.2. Results

The results of the lemmatization task are described in Table 1.

KU Leuven / Brepols-CTLO closed	LEMMATIZATION
Ab Urbe Condita (classical)	85.44
Metamorphoseon (cross-genre)	87.22
Naturalis Historia (cross-genre)	85.75
De Latinae Linguae Reparatione (cross-time)	84.60

Table 1: Results of the lemmatization task

3.3. Discussion

While it is clear that our system performs worse than our competitors (Sprugnoli et al., 2022), this can be at least partly attributed to differences in defining a lemma. As mentioned in the previous section, we had to implement manual rules to make sure that the ReFlex lemmata were consistent with EvaLatin lemmata. This was done based on frequent mistakes while tagging a validation dataset (20% of the provided training dataset). However, due to time constraints, it was not feasible to remove all these inconsistencies. It comes apparent, for example, that EvaLatin prefers the plural form as a lemma for demonyms such as *Allobroges*, *Samnites*, *Romani*, while ReFlex resorts to the singular *Allobrox*, *Samnis* and *Romanus*. A second problem are the so-called deponent verbs, where EvaLatin prefers the passive form as a lemma, while ReFlex returns the active form, even if this form is only attested once (otherwise, ReFlex also gives the passive form). Likewise, EvaLatin takes *fio* ("I become") as a separate lemma, while ReFlex considers it the passive form of *facio* ("I make"). Thirdly, ReFlex will always return the original verb when faced with adjectival participles such as *iratus* ("angered"), *tutus* ("guarded") and *excellens* ("towering"), while EvaLatin chooses the adjective in these cases. Finally, the relative pronoun *quis* was consistently tagged as *qui*, while the ablative *quo* (with lemma *qui* in EvaLatin) was tagged as *quo* by ReFlex as if it were an adverb ("where"). These relative pronoun errors make up 6,3 % of the lemmatization errors, which is a significant amount. In the future, we will take the frequency of a lemma into account, to avoid situations in which a very common word such as *cum* ("with", "when") is lemmatized as an infrequent lemma *Cous* ("of Cos", "Coan").

4. POS-tagging

4.1. Methodology

Our POS-tagging system is very straightforward: we trained a Huggingface Transformers ElectraForTokenClassification model on the provided datasets. Based on our own previous experiments with inflectional languages, we decided to make one modification. As most modern language models do, ELECTRA models make use of a subword tokenizer, which processes frequent forms as one token and splits less common forms into smaller subwords, e.g. *amat* ("he/she loves") is tokenized as *amat*, while *amabamini* ("you were loved") becomes *ama #bam #ini*. Thus, an important step consists of determining on which subword of the complete word the actual token tagging will take place. Usually, a tagger uses the embedding of the first subword, or the average of all the subwords. Our system uses the last subword of a token, as crucial morphological information is stored in the last part of the word, because Latin is an inflectional language (Ács et al., 2021). In the future, we will further experiment with other, more advanced subword pooling techniques, as discussed in Ács et al. Ács et al. (2021).

4.2. Results

The results of the POS-tagging task are described in Table 2.

KU Leuven / Brepols-CTLO closed	POS-TAGGING
Ab Urbe Condita (classical)	96.33
Metamorphoseon (cross-genre)	94.66
Naturalis Historia (cross-genre)	89.96
De Latinae Linguae Reparatione (cross-time)	92.11

Table 2: Results of the POS-tagging task

4.3. Discussion

The results show that our system performs well, coming just short of the results of our competitors in the EvaLatin campaign. In 52,7 % of the mistakes on the test set, PROPN is either the gold label that gets a different tag, or PROPN is wrongly predicted instead of the correct tag. Many of the latter are geographical adjectives such as *Romanus* that can also be used as nouns. Furthermore, less frequent words with non-Latin roots such as *psitthachoras* (a certain kind of tree)

are often tagged as PROPN as well, probably because of the similarity with Greek personal names. This type of words is especially frequent in Pliny, describing various plants etc. Secondly, the aforementioned problem concerning the distinction between adjectives and participles (and thus, verbs) explains some mistakes in this task as well.

5. Morphological tagging

5.1. Methodology

Rather than predicting all the features at once, which causes issues of data sparsity on the one hand, and a large amount of labels on the other hand, we trained a separate classifier for each of the morphological features defined in the dataset. Next, we calculated the probability of the full morphological tag as the product of the probabilities of the individual features: e.g. P(Case=Gen—InflClass=IndEurO—Number=Sing) is defined as P(Case=Gen) * P(InflClass=IndEurO) * P(Number=Sing). This is similar to the approach used by RFTagger (Schmid and Laws, 2008) and is defined by Tkachenko and Sirts Tkachenko and Sirts (2018) as the Multiclass Multilabel model. For this, we used the same architecture as discussed before in the POS-tagging section. Afterwards, we combine the predicted labels into one tag. Rather than taking a naive approach (taking the highest-scoring prediction for each feature and combining them, without constraints), which can lead to impossible combinations (such as adjectives receiving a mood feature), we predefine a set of possible combinations of tags, which act as constraints on the output of our system. These tag combinations are mostly based on POS-tags (e.g. interjections do not have any morphological features), but are sometimes more fine-grained, particularly for verbs as there are different rules needed to distinguish, for example, finite verbs and participles. Combining this approach with a lexicon of tags that occur in the training data ensures that no impossible predictions are formed.

5.2. Results

The results of the morphological tagging task are described in Table 3.

KU Leuven / Brepols-CTLO closed	MORPHOLOGICAL TAGGING
Ab Urbe Condita (classical)	69.91
Metamorphoseon (cross-genre)	63.06
Naturalis Historia (cross-genre)	58.04
De Latinae Linguae Reparatione (cross-time)	60.09

Table 3: Results of the morphological tagging task

5.3. Discussion

The results of this task are rather disappointing. A big part in this is played by exceptions, which we will illustrate with an example. In the test data, we find instances of the word *opus*, with only InflClass=IndEurInd as a morphological feature. This is an exception to the usual morphological features of a noun, which involve an InflClass, a Case and a Number. However, to accommodate our ruleset in such a way that the exceptions are handled as well, we have to allow nouns to only have a IndEurInd feature. As such, our constraint-based system is weakened by these few exceptions, leading to mistakes where the Number feature for example, is mistakenly omitted. Furthermore, morphologically identical features, such as the nominative and accusative for neuter words, have considerably more errors than features that are morphologically different. This is already apparent while training the data: on the validation data we see that there are 317 nominatives falsely tagged as accusatives, compared to only 17 falsely tagged datives and 34 genitives (8786 nominatives received the right tag). Currently, we are looking into better ways of combining the different tags, as our separate morphological feature classifiers are performing considerably better than the sum of their parts.

6. Conclusion

In this report, we described the first steps in using an ELECTRA model for Latin token tagging tasks. In the future, we will train a larger model on the one hand, and refine our system on the other hand, especially with regards to the morphological tagging task.

7. Acknowledgements

Our work has been funded by grant no. HBC.2021.0210 of Flanders Innovation and Entrepreneurship.

8. Bibliographical References

Ács, J., Kádár, Á., and Kornai, A. (2021). Subword pooling makes a difference. *CoRR*, abs/2102.10864.

Schmid, H. and Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK, August. Coling 2008 Organizing Committee.

Sprugnoli, R., Passarotti, M., Cecchini, F. M., Fantoli, M., and Moretti, G. (2022). Overview of the evalatin 2022 evaluation campaign. In Rachele Sprugnoli et al., editors, *Proceedings of the LT4HALA 2022 Workshop - 2nd Workshop on Language Technologies for Historical and Ancient Languages, satellite event to the Twelfth International Conference on Language Resources and Evaluation (LREC 2022)*, Paris, France, June. European Language Resources Association (ELRA).

Tkachenko, A. and Sirts, K. (2018). Modeling composite labels for neural morphological tagging. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 368–379, Brussels, Belgium, October. Association for Computational Linguistics.

9. Language Resource References

Bamman, D. and Burns, P. J. (2020). Latin BERT: A contextual language model for classical philology. *CoRR*, abs/2009.10053.

Kevin Clark and Minh-Thang Luong and Quoc V. Le and Christopher D. Manning. (2020). *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.