Adapting transformer models to morphological tagging of two highly inflectional languages: a case study on Ancient Greek and Latin

Alek Keersmaekers
KU Leuven
Blijde-Inkomststraat 21, 3000 Leuven (BE)

alek.keersmaekers@kuleuven.be

Wouter Mercelis
KU Leuven*, Brepols Publishers†
*Blijde-Inkomststraat 21, 3000 Leuven (BE)
†Begijnhof 39, 2300 Turnhout (BE)
wouter.mercelis@kuleuven.be

Abstract

Natural language processing for Greek and Latin, inflectional languages with small corpora, requires special techniques. For morphological tagging, transformer models show promising potential, but the best approach to use these models is unclear. For both languages, this paper examines the impact of using morphological lexica, training different model types (a single model with a combined feature tag, multiple models for separate features, and a multi-task model for all features), and adding linguistic constraints. We find that, although simply fine-tuning transformers to predict a monolithic tag may already yield decent results, each of these adaptations can further improve tagging accuracy.

1 Introduction

Morphological information is an essential enrichment for corpora of highly inflectional languages such as Ancient Greek and Latin. Yet given that the field of natural language processing has traditionally been heavily oriented to Modern English, a relatively analytic language, the automated processing of morphologically rich languages has been a challenge for some time already (see e.g. Tsarfaty et al., 2010).

For Ancient Greek (henceforth simply 'Greek') and Latin, Sommerschield et al. (2023) have noted that, as for many other languages, the transformer-based approach has recently become popular for morphological tagging, showing promising results. However, it is still an open question what the most appropriate way is to employ transformer models for this task, i.e. whether specific adaptations are necessary for inflectional languages.

The aim of this paper is therefore to systematically compare a number of adaptations that were previously found to be beneficial for morphological tagging of Greek and Latin using older methods and assess the importance of these adaptations in a transformer context. We will first discuss previous work related to this topic (Section 2). Next, we will present the experimental set-up of this project (3), including the data and models we used, and assess which parameter combinations contribute to optimal performance for the two languages (4.1). We will also give a general evaluation of the errors of the best-performing models (4.2). Finally, we will summarize the main results of this study and discuss ways for further improvement (5), and address its limitations (6).

2 Previous work

Given the vast body of literature on morphological tagging, this section will focus on related work to the central topic of this paper, viz. transformer-based approaches to Greek and Latin morphological tagging, as well as earlier approaches that have explicitly aimed to adapt tagging techniques to the typological characteristics of these languages. We will therefore not discuss studies that focus on comparing a number of readily available tagging tools (e.g. Celano et al., 2016; Poudat and Longrée, 2009), since these tools typically differ on various parameters, so that it is difficult to tell why exactly certain tools are better to handle Greek and Latin than others.

The morphological richness of Greek and Latin has various consequences: data sparsity arises due to a high number of tokens compared to types,[1] the tag set (i.e. the number of possible combinations of

[1] For example, for each type (unique word form) in the GUM English Universal Dependencies Treebank (see https://universaldependencies.org/) there are 10.7 tokens. For the Latin PROIEL treebank there are only 6.5, and for the Greek Perseus treebank even less, viz. 4.8 (note that they are all roughly similar in size: 212K, 205K and 202K tokens respectively).

Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024), pages 165–176

morphological features) is very large and morphology and syntax are often interrelated (e.g. with case marking). As for data sparsity, Hajic (2000) advocates for the use of morphological dictionaries for inflectional languages in general, viz. knowledge bases containing lists of morphologically inflected forms and their analysis. In this way the correct analysis for unattested or lowly attested forms can be retrieved from this dictionary instead of solely relying on the training data of the tagger (additionally, even if multiple analyses are present in the lexicon for a given form, the number of possible tags will be heavily constrained by it). Various researchers have observed a positive effect of employing such lexica for Greek (e.g. Dik and Whaling, 2008; Keersmaekers, 2020) and Latin (e.g. Eger et al., 2015).

As for the size of the tag set, it is important to remark that it is only large if we treat the combination of part-of-speech and all the morphological features as one singular label (as is customary for English), i.e. the tag would be 'noun, singular, feminine, dative'. Some researchers on inflectional languages have recommended 'splitting' the tags, i.e. making separate predictions for all the individual morphological features, instead (e.g. Schmid and Laws, 2008; Tkachenko and Sirts, 2018). Such an approach has been advocated by e.g. Keersmaekers (2020), Riemenschneider and Frank (2023) for Greek and Eger et al. (2015) for Latin, but so far it has not been compared to a 'singular label'-approach yet.

Finally, as for the interrelatedness of morphology and syntax, some scholars (e.g. Lee et al., 2011) have shown that performing morphological tagging and syntactic parsing jointly can help both tasks, but since this requires a high performing syntactic parsing model as well, such an approach falls outside the scope of this paper.

As noted in the introduction of this paper, recently (encoder-only) transformer models have become popular for Greek and Latin morphological tagging. They have been employed in various ways, including directly finetuning a pretrained large language model (LLM) for this task (Mercelis and Keersmaekers, 2022a; Wróbel and Nowak, 2022; Riemenschneider and Frank, 2023), by extracting the embeddings of a pretrained LLM and processing them combined with other information through a simpler architecture (Straka and Straková, 2020; Singh et al., 2021; Swaelens et al., 2023), or, occasionally, utilizing prompts on generative transformer architectures (Stüssi and Ströbel, 2024).

The effect of the various parameters described above, including the use of a morphological lexicon and the 'splitting' of morphological tags, has so far not been systematically investigated in a transformer context. In fact, there are reasons to suspect that their effect may be diminished, given that transformer architectures have specific adaptations to handle data sparsity and morphological richness. Firstly, transformer models are typically pre-trained on millions (or billions in the case of modern languages) of unannotated tokens, allowing them to recognize forms beyond the specific training set for morphological tagging. Nevertheless, the problem remains that morphological richness inherently implies a proportionally larger number of word form types, and due to the closed nature of historical language corpora these pre-trained models are also typically trained on lower amounts of data as compared to modern languages. [2] Secondly, in most modern transformer architectures subword tokenization is typically employed (see e.g. Kudo and Richardson, 2018), which splits morphologically complex words in several parts, based on statistical pattern recognition. For example, the tokenizer of the transformer model we will employ for Latin (see 3.1) splits the morphologically complex verb *honorificentur* into *honorific+entur*, so that even if the full form *honorificentur* might be scarcely attested, the individual parts *honorific-* and *-entur* would be more frequent. In this paper we will therefore systematically investigate whether modern transformer architectures have completely superseded the need for any special adaptations for inflectional languages, or if morphological lexica and splitting tags may still offer improvements.

3 Methodology

3.1 Data and models

In this paper, we compare morphological tagging for Greek and Latin. While these languages are typologically rather similar (both highly

[2] Although sometimes a modern language model is finetuned for ancient languages, as e.g. in Singh et al., 2021.

inflectional Indo-European languages), the external resources we used for each of them respectively results in two very different experimental conditions.

For Greek we have a relative large and diverse body of manually tagged data (1.46M tokens, of which we reserved 1.24M as training data and 219K as test data), which is a result of a data homogenization effort of various treebanks by the GLAUx project (Keersmaekers, 2021). This dataset consists of various text genres (29 in total according to the GLAUx classification) from all three major Ancient Greek time periods (archaic, classical and post-classical). We could also make use of a morphological lexicon from GLAUx which was specifically developed to be compatible with the treebank data (see 3.3).

In contrast, while for Latin various treebank project exists and some effort has recently been undertaken to homogenize them (Gamba and Zeman, 2023), these efforts have only been published very recently and we were not aware of them when we wrote this paper. We therefore instead made use of the largest dataset present in the Universal Dependencies (UD) project (Nivre et al., 2020) that was relatively diverse, viz. the PROIEL treebank (Haug and Jøhndal, 2008), consisting of 205K tokens, including the Vulgate New Testament, a late classical work by Palladius as well as more classical texts (by Caesar and Cicero). This dataset was therefore substantially smaller (we used the 'train' subset, consisting of 178K tokens, and the 'test' subset, 14K tokens). The lexicon we used was also not specifically developed to be compatible with this treebank (see 3.3). On the other hand, this allowed us to compare results for a situation that is rather typical for low-resource languages, where large datasets and standardized resources are typically absent.

As for our morphological tagging approach, our basic method was relatively simple: we fine-tuned pre-trained transformer models to predict either one or multiple labels (see 3.2) consisting of part-of-speech and morphological information. For Greek, we used *electra-grc* (Mercelis and Keersmaekers, 2022b), a small ELECTRA model trained on the GLAUx corpus, allowing us to use a model that was trained on a corpus with a data standard that was consistent with our tagging dataset. For Latin, we used LaBERTa, a base-size RoBERTa model offering state of the art performance for Latin morphological tagging (Riemenschneider and Frank, 2023). Since our data was tokenized into subwords, the training and predictions were always based on the final subwords of the token. We fine-tuned all models for a fixed number of 10 epochs, using a batch size of 16 and a learning rate of 5e-5.

3.2 Splitting tags

We evaluate the impact of predicting a single tag containing the part-of-speech proper and all morphological information (we call this approach *MonoTag* in what follows), vs. predicting each morphological feature separately. We compared two methods to perform the latter task: the simplest way is to train a tagging model for each feature (*MultiTag*). We then calculate the probability of a morphological tag as the product of the probabilities of each individual feature, and select the tag with the highest probability – this is the *Multiclass Multilabel model* described in Tkachenko and Sirts (2018). While this approach is statistically rather naïve, given that the probabilities of the various features are not independent,[3] it yielded decent results on the Greek and Latin datasets evaluated by them.

Another approach is to employ multi-task learning, as was done by Riemenschneider and Frank (2023) for Greek. In this approach (*MultiTag-MultiTask*), we do not train separate models for each feature, but rather train them all together. To achieve this, we use a shared encoder with for each feature a classification head on top. In this way, the model should generalize better and capture how the various morphological features interrelate during the training phase due to the shared loss function. Additionally, this method is computationally more efficient and less prone to overfitting.

Figure 1-3 visualize the three approaches.

Figure 1: MonoTag approach.

[3] To give just one example, Greek possesses several feminine words that have an identical ending in the genitive singular and the accusative plural, viz. -ας. Obviously in such a case the probabilities of the features 'case' and 'number' are highly dependent on each other.

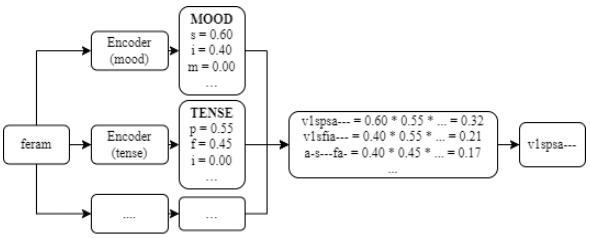

Figure 3: MultiTag approach.

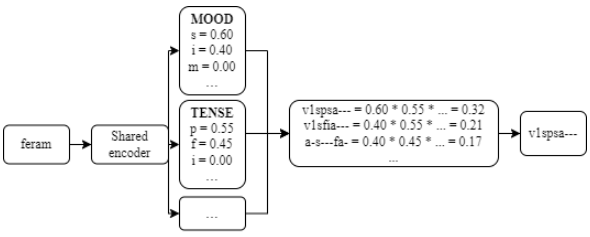

Figure 2: MultiTag-MultiTask approach.

3.3 Morphological lexica

We test the impact of employing an external lexicon consisting of inflected forms and their possible morphological analyses. For Greek, we used a lexicon from the GLAUx project, which was based on the morphological analysis tool Morpheus (Crane, 1991) and of which its output was converted and homogenized in order to be compatible with the morphological tagging of GLAUx. For Latin, we analyzed all forms in the test data with LEMLAT (3.0) (Passarotti et al., 2017). Since the output of this analyzer was not compatible with the UD annotation of PROIEL, we created a script in order to convert it to the latter format using a number of rules.

Concretely, we employed these lexica as follows: if an inflected form occurred in the lexicon, the possible tags that could be predicted were constrained to the ones corresponding to this form. To avoid the problem that some words may have analyses that are not present in this lexicon, we also added all forms from the training data and their tags to it. Our lexica covered the test data very well: for both languages only 0.4% of the forms in the test data were not present in the lexicon.

Figure 4 illustrates the integration of a lexicon in the *MultiTag-MultiTask* approach (in *MonoTag* and *MultiTag*, the integration happens analogically).

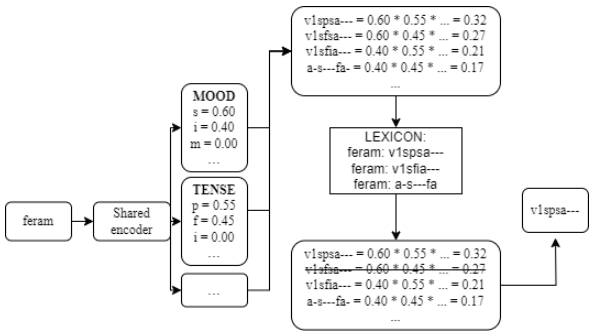

Figure 4: Integrating a lexicon in the tagging process.

3.4 Constraining the outcome space

When predicting the various features individually, one risk is that linguistically nonsensical feature combinations could be predicted (e.g. a passive noun). While the use of a lexicon may already reduce this problem to a great extent (since the possible combinations are limited to the ones occurring in the lexicon for a specific form), the problem potentially remains for forms that are not present in it. We therefore experiment with two approaches adding additional constraints on the tag outcomes: firstly, we restrict the possible tags that could be predicted to the ones occurring in the training data. A disadvantage of this approach is that if a feature combination does occur in the test data but not in the training data, it can never be predicted. [4] We therefore also tried a second approach, which consists of adding an external list of linguistically valid feature combinations for Greek and Latin to the list of tags occurring in the training data, based on a number of constraints that we defined for both languages (e.g. nouns cannot receive the feature voice, the future tense cannot occur in the subjunctive mood). In this way, all feature combinations that could logically occur in Greek and Latin could in theory be predicted.

Figure 5 illustrates the addition of constraints to the outcome space in the *MultiTag-MultiTask* (in *MonoTag* and *MultiTag* this again happens analogically), which can either come from the training data or an external list (e.g. in Figure 5, an external list has determined that the *s*[ubjunctive] mood and *f*[uture] tense are not compatible).

[4] Although this does not occur very often, for Latin there were 12 tokens and for Greek 18 where this was the case.

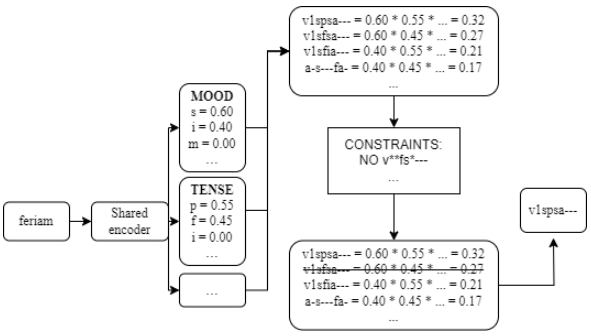

Figure 5: Constraining the outcome tag space.

4 Results

4.1 Parameter comparison

4.1.1 Greek

Tables 1-3 show the results for the three training approaches described in section 3.2 (*MonoTag*, *MultiTag*, *MultiTag-MultiTask*) for Greek. Firstly, it is clear that the use of a lexicon has a positive effect across all three approaches, allowing for a 15-20% error reduction. These differences are also statistically significant: with McNemar's test, $p<0.01$ in all cases when comparing the models with and without lexicon. Taking the *MonoTag* approach as an example, the lexicon corrected 1569 tagging mistakes, although it also introduced 240 new mistakes. An example of the former case is (1), in which φθιμένῃ (*phthiménēi*) was originally tagged as a present participle, but corrected by the lexicon to an aorist participle (the correct analysis). Given that the present of the same verb would be superficially similar (φθι(ν)ομένῃ *phthi(n)oménēi*), intricate knowledge of Greek verbal morphology is necessary to predict that it is an aorist, which the transformer model was not able to pick up. In particular, the lexicon was a valuable asset to handle Greek verbal morphology in a better way than the transformer model was able to do: verbs consisted of 26.5% of the mistakes when no lexicon was employed, but 22.0% when a lexicon was employed, the largest difference among all parts of speech.

(1) καίτοι **φθιμένῃ** μέγα κἀκοῦσαι τοῖς ἰσοθέοις σύγκληρα λαχεῖν. (Soph. Ant. 836-7)
kaítoi ***phthiménēi*** *mega kakoûsai toîs isothéois súgklēra lakheîn.*
"Yet it is great **for someone who died** to earn a fate equal to that of the gods."

Nevertheless, there were some new mistakes that the lexicon introduced. These were typically

Lexicon	Accuracy
No	0.963 (210635)
Yes	0.969 (211964)

Table 1: Greek tagger results (*MonoTag*), with accuracy and N correct predictions.

Lexicon	Tag constraints	Accuracy
No	None	0.964 (210891)
	Training data	0.967 (211569)
	Tag list	0.967 (211529)
Yes	None	0.972 (212567)
	Training data	0.972 (212596)
	Tag list	0.972 (212592)

Table 2: Greek tagger results (*MultiTag*), with accuracy and N correct predictions.

Lexicon	Tag constraints	Accuracy
No	None	0.964 (210805)
	Training data	0.964 (210950)
	Tag list	0.964 (210929)
Yes	None	0.970 (212169)
	Training data	0.970 (212177)
	Tag list	0.970 (212176)

Table 3: Greek tagger results (*MultiTag-MultiTask*), with accuracy and N correct predictions.

cases in which the lexicon was not strictly incorrect, but simply inconsistent with the data. For example, in (2), ὠμόφρονος *ōmóphronos* was tagged without the lexicon as an adjective (as it appears in the data) but with a lexicon as a noun. Since it has an adjectival meaning but morphologically it shares characteristics with nouns (having no gender inflection), both analyses could be argued to be correct, especially since there were no strict annotation guidelines in the data we used (see 3.1) to handle such cases.

(2) σίγα, τέκνον, μὴ κινήσῃς ἀγρίαν ὀδύνην πατρὸς **ὠμόφρονος**. (Soph. Trach. 975-6)
síga, téknon, mḕ kinḗsēis agrían odúnēn patrós ***ōmóphronos***.
"Be quiet, child, so that you will not stir the savage pain of your **savage-minded** father."

Comparing the three training approaches, the *MultiTag* approach performs slightly better than the *MonoTag* approach. In the best case (when also combined with constraints on the tag outcomes, see below), this allows for a 10% error reduction both with (96.9% to 97.2% accuracy) and without

(96.3% to 96.7%) lexicon. These differences are also statistically significant (p<0.01 with McNemar's test in both cases). Taking the lexicon-based approach, without any constraints on the outcome tags, as an example, the *MultiTag* method was able to correct 1898 mistakes but unfortunately also introduced 1295 new mistakes. One obvious advantage of this approach is with scarcely attested tag combinations: for example, tag combinations that occur 50 times or less in the training data constitute 6.5% of mistakes with the *MonoTag* approach (448 in total) but 4.3% with the *MultiTag* approach (268). It introduced quite a large number of new mistakes, however. An example is (3), in which βέλτιστ᾽ *béltist'* (literally 'best') was tagged correctly as a masculine singular vocative by the *MonoTag* approach but as a neuter (plural) vocative by the *MultiTag* approach. Obviously in this case the morphological features are highly dependent on each other: if βέλτιστ᾽ *béltist'* is analyzed as a vocative (used in appellative contexts), it is much more likely that it refers to a masculine than a neuter entity. Cases such as this one might explain why the statistically 'naïve' approach of predicting each feature individually, assuming independence between these features, may return worse results than predicting one tag containing all morphological information.

(3) μὴ δὴ πράγματ᾽, ὦ **βέλτιστ**᾽, ἔχε· (Men. Dysc. 338)
*mḕ dḕ prágmat', ô **béltist'**, ékhe.*
"Don't worry, **my dear friend**."

It would be expected that the multi-task model would improve in such cases (see Section 3.4), however, as can be judged from Tables 2-3, the multi-task models consistently performed slightly worse than the separately trained models. Taking again the lexicon-based approach without any constraints on the tag outcomes as an example, while the multi-class model corrected 1163 of the mistakes of the separately trained model, it also introduced 1561 new mistakes. It is difficult to explain why this is the case: the general qualitative characteristics of the errors of the multi-class models were similar to those of the separately trained models (see 4.2), but simply quantitatively more numerous.

Finally, the effect of adding constraints to the possible tag outcomes is rather mixed. If the possible tags are restricted to those occurring in the training data, this has a somewhat visible positive effect for the *MultiTag* model when no lexicon is employed (about an 8% error reduction) and a tiny positive effect with a lexicon as well (about a 0.5% error reduction) – note that these constraints only apply for forms that do not occur in the lexicon (since the lexicon already acts as a constraint for the other forms), which are only 3% of all errors of this model (215/6220), so a large error reduction is not expected. For the multi-task model, the differences are barely visible. Focusing on the *MultiTag* approach with a lexicon, constraining the tag outcomes to the ones occurring in the training data corrected 29 mistakes while not introducing a single new mistake. Most of these 29 mistakes were impossible feature combinations: for example, in (4) ἐξόπιστο *eksópisto* 'from behind' was predicted as an adverb with the aorist tense, presumably because the tagger was conflicted between an adverbial and a verbal analysis (since -το *-to* is a common verbal ending).

(4) εἰ σπόδρ᾽ ἐπιτυμεῖς τὴ γέροντο πυγίσο, τὴ σανίδο τρήσας **ἐξόπιστο** πρώκτισον. (Aristoph. Thesm. 1123-4)
*ei spódr' epitumeîs tḗ géronto pugíso, tḗ sanído trḗsas **eksópisto** prṓktison.*
"If you desperately want, have anal sex with the old man, make a hole in the board and penetrate him **from behind**."

Restricting the possible tag combinations to the ones occurring in the training data has an obvious disadvantage: if the feature combination does not occur in the training data, it cannot be predicted. For Greek this occurs very rarely due to the size of the training data, but there are still 18 tokens in the test where this is the case (typically containing very rare features: 13/18 cases have dual number, which died out in an early stage in Greek). As argued in Section 3.4, adding an external list of possible tag combinations might help in these cases. Unfortunately, as can be judged from the numbers in Tables 2-3, in all cases this has a very small net negative effect instead. Again focusing on the *MultiTag* approach with a lexicon as an example, in all the 18 cases mentioned above a wrong tag was still predicted, while there were 4 new mistakes. Apparently the possible tags list was a little too permissive, introducing feature combinations that we would not expect to occur in the corpus and which were then erroneously applied in some cases. For example, one form (ἅπις *âpis*) was analyzed as a nominative masculine singular personal pronoun,

which was present in the possible tags list but we would not expect to actually occur in Greek texts.[5]

4.1.2 Latin

Tables 4-6 show the results for the three training approaches (*MonoTag*, *MultiTag*, *MultiTag-MultiTask*) for Latin.

In contrast to Greek, adding a morphological lexicon does not seem to have a positive effect – in some cases even a slightly negative one, although the difference in absolute numbers is minimal. Taking the *MultiTag* model with the possible tags constrained by the training data as an example, even though the lexicon corrected 84 mistakes, it unfortunately also introduced 102 new ones. Many of these new mistakes involved proper nouns (36 out of 102), where the vocabulary of LEMLAT seemed to be incomplete. For example, the proper noun *Furio* (here in the dative case) is included in the lexicon as an adjective, or a verb form. Note that these are valid options, but the proper noun analysis should have been included as well.

In comparison with the Ancient Greek tagger, the multi-task model again falls just short of the simpler *MultiTag* approach. For Latin, the model corrects 202 mistakes, while it introduces 212 new mistakes. Again, it is difficult to explain why, since as for Greek, no general categories can be found in the newly introduced errors.

For the addition of constraints, we observe that constraining the output to combinations that occur in the training data has a positive effect on the *MultiTag* model, while the effect is much smaller for the *MultiTag-MultiTask* model. When we take the lexicon into account as well, the constraint options yield no differences at all.

As for Greek, the use of an external list of possible tags had a net negative effect on the result. More precisely, of the 12 tokens in the test data that had a tag that did not occur in the training data, only 1 received the correct tag (*primis*, an ablative masculine plural adjective without the degree feature). Meanwhile, the list introduced 13 new errors. Again, these were mainly cases where the list of possible tags was too permissive: for example, for the form *mi* (a dative of *ego*, *I*) the tagger predicted that it was in the vocative case, which would not be possible for a first person personal pronoun.

Lexicon	Accuracy
No	0.936 (13191)
Yes	0.933 (13151)

Table 4: Latin tagger results (*MonoTag*), with accuracy and N correct predictions.

Lexicon	Tag constraints	Accuracy
No	None	0.932 (13131)
	Training data	0.937 (13210)
	Tag list	0.937 (13198)
Yes	None	0.936 (13192)
	Training data	0.936 (13192)
	Tag list	0.936 (13192)

Table 5: Latin tagger results (*MultiTag*), with accuracy and N correct predictions.

Lexicon	Tag constraints	Accuracy
No	None	0.936 (13193)
	Training data	0.937 (13203)
	Tag list	0.937 (13200)
Yes	None	0.934 (13168)
	Training data	0.934 (13168)
	Tag list	0.934 (13168)

Table 6: Latin tagger results (*MultiTag-MultiTask*), with accuracy and N correct predictions.

4.2 Error analysis

In this section, we will analyze the remaining errors of two high-performing models, viz. the model with split tags, lexicon and morphological tags for both languages. We will do this by analyzing a random sample of 100 errors for both languages. In appendix, we also provide plots analyzing more general qualitative characteristics of the tagging errors, viz. the accuracy by morphological feature (appendix A) and by text type (appendix B).

Error	Proportion
Mistake gold data	41%
Data consistency	15%
Syntactic structure	11%
Mistake lexicon	10%
Various	24%

Table 7: Error analysis for Greek.

[5] Note that first and second person personal pronouns were never gendered in our corpus, since Greek makes no morphological gender distinctions. The only personal pronouns that can be gendered are reflexive third person personal pronouns, but these never occur in the nominative case.

4.2.1 Greek

A quantitative description of the mistakes we found is presented in Table 7. Strikingly, a very large part (41%) of them were actually cases where the gold data was incorrectly annotated and the tagger was correct, suggesting that the actual accuracy of the tagger is even higher than 97% (although it could also be the case that some analyses labeled as 'correct' were in fact wrongly annotated in the gold data as well). An additional 15% of errors were issues of data consistency, typically related to part-of-speech, where the boundaries between part-of-speech can be fluid and there are no consistent choices in the training/test data, as was already discussed above.

Moving to the actual errors, 11% of cases can be explained because the transformer model understood the syntactic structure of the sentence incorrectly. For example, in (5), βασιλήιον *basilḗion* was analyzed as a noun by the tagger. The noun βασιλήιον *basilḗion*, meaning 'palace', certainly exists, but in this case it is clearly an adjective 'royal' modifying the noun τεῖχός *teîkhós* 'fortress' (if it was a noun, it would not fit in the sentence context, given that the subject slot of ἐδέδμητο *edédmēto* 'it was built' is already taken up by τεῖχός *teîkhós*).

(5) ἐν τῷ τεῖχός τε ἐδέδμητο **βασιλήιον** τοῦτο τὸ δὴ Δορίσκος κέκληται… (Hdt. 7.59.1)
*en tôi teîkhós te edédmēto **basilḗion** toûto tó dḕ Dorískos kéklētai…*
"at which that **royal** fortress was built which was called Doriscus…"

10% of errors were simply related to mistakes in the tagger lexicon: even though it had a net positive effect, fixing these mistakes could therefore further improve the results. The remaining 24% of errors were rather diverse. Interestingly, in 6% of cases the correct morphological analysis could only be made by logical inferences. For example, in (6) δακρύων *dakrúōn* was analyzed as a noun instead of the participle of δακρύω *dakrúō* 'to cry', which it could theoretically be: in that case θάλασσαν δακρύων *thálassan dakrúōn* would mean 'sea of tears'. While we could plausibly expect such an expression in e.g. a poetic context, it is much more logical that δακρύων *dakrúōn* means 'crying' in this context rather than that the farmer would curse his own massive torrent of tears. Obviously such logical inferences are easy to make for humans, but pose a challenge for a tagger.

(6) γεωργός τις ἰδὼν ναῦν ἐν θαλάσσηι κυμαινομένην καὶ βυθῶι πεμπομένην, κατηρᾶτο τὴν θάλασσαν **δακρύων**. (Aes. Fab.)
*geōrgós tis idṑn naûn en thalássēi kumainoménēn kaí buthôi pempoménēn, katērato tḗn thálassan **dakrúōn**.*
"A farmer, seeing a ship being tossed on the waves and being sent into the deep sea, cursed the sea **while crying**."

Some other errors include cases related to the coreference chain (5, e.g. the gender of a pronoun was incorrectly determined, because the entity that the pronoun refers to occurs in another sentence), to the diversity of the Greek corpus (3, e.g. dialectal forms that were difficult to determine correctly) and general problems related to data sparsity (2), to damage/corruption to the actual text (2), 1 case clearly related to the issue that the morphological features were independently predicted (see 4.1), 1 case of true ambiguity (i.e. both the gold and the predicted tag can be argued to be correct, depending how the sentence is interpreted) and finally 3 cases where we did not find any explanation for.

Error	Proportion
Data consistency	45%
Mistake gold data	24%
Syntactic structure	16%
Various	15%

Table 8: Error analysis for Latin.

4.2.2 Latin

Our results (see Table 8) largely reflect similar problems to the ones for Greek. While the data contained less wrongly annotated forms than the Greek data (24%), an even larger proportion of the mistakes related to annotation conventions (45%). In this latter category, a very large proportion of problems (28/45) involved double- (23) and triple- (5, meaning no gender at all in the PROIEL annotation) gendered forms. In the error analysis, we considered a form to be triple-gendered if it does have a case and a number, but no gender. An example is (7), in which *multis* (which theoretically can be all three genders) agrees with *regionibus.* Since the PROIEL treebank is not very consistent in which cases forms are considered double/triple-gendered, it is not surprising that the tagger

analyzed it as feminine (as *regionibus* is), even though it was triple-gendered in the gold data.

(7) et **multis** regionibus Samaritanorum evangelizabant (Acts 8:25)
"and they preached the gospel to **many** villages of the Samaritans"

As for Greek, some errors were related to the transformer model misinterpreting the syntactic structure of the sentence (16%), while mistakes caused by errors in the lexicon are more rare (only 2% – specifically cases where the lexicon was incomplete, such as *Furio* as described in 4.1.2). As for the other problems (13%), they are rather analogous to the problems found for Greek, so we will not discuss them here.

5 Conclusions

The aim of this paper was to investigate whether transformer models need special adaptations to morphologically tag highly inflectional languages with data sparsity, using Ancient Greek and Latin as a test case. We show that, although the most simple approach – i.e. finetuning a transformer model on tags containing all morphological information – already performs decently, special adaptations tailored to the typological nature of these languages can still further improve tagging accuracy.

Firstly, the use of a morphological lexicon had a clear positive effect on Greek tagging accuracy. On Latin, conversely, the effect was negative in most cases. This can largely be explained by the quality of the respective lexica: the Latin lexicon contained a relatively large number of cases (primarily proper nouns) where not all possible analyses for a given token were recorded in the lexicon, and therefore introduced new tagging errors. Nevertheless, the proportion of errors that the Latin lexicon corrected (84/881, or about 10%) was still relatively modest. There are multiple explanations why a morphological lexicon might be less necessary than for Greek: this might be because Greek could be morphologically more complex, or because the pretrained transformer for Latin was trained on much more data than for Greek, or because the Latin data was simply more homogeneous.

Training separate models for each individual morphological feature had a positive, although very modest effect for both languages. Surprisingly, however, multi-task learning did not further improve the results, but had a (slight) detrimental effect instead. We were not able to explain why this was the case. In the future, however, we plan to experiment with other methods to combine the outputs of the individual feature models, as described in Tkachenko and Sirts (2018).

As for constraining the tag outcomes to the ones occurring in the training data, this had a very slight positive effect for Greek and no effect for Latin. Further adding a linguistically-based list of possible tags did have a slight negative effect for both languages, however. This was caused by a too permissive list of combinatory possibilities, so that feature combinations were predicted that could not co-occur. This is therefore a consequence of the quality of the concrete external list we used, and since it is only through such a list that feature combinations can be predicted that do not occur in the training data, we still generally recommend using this technique.

An error analysis revealed where there was room for further improvement. For both languages, data errors and consistency issues made up a very large proportion of errors. Most improvement can therefore not be made through more sophisticated machine learning algorithms, but by simply improving the quality of the data. Some other errors (e.g. related to logical inferencing or co-references across sentences) would also be hard to solve by the current generation of NLP techniques. A more promising category of errors were related to co-dependence of morphological and syntactic analysis. In this case, joint syntactic parsing and tagging may offer a possible solution.

Finally, we should note that, while this paper focused on Greek and Latin, the techniques we explore are not solely tied to these historical languages, given that there are many other inflectional languages with sparse datasets. We therefore hope that the solutions offered here could also inspire researchers working on similar languages.

For the sake of reproducibility and to allow other researchers to make use of the resources this study produced, all the code and datasets we used can be found on GitHub (see 'Supplementary Material').

6 Limitations

There are some limitations inherent to the experiments carried out in this paper. Firstly, to avoid having to compare too many models, we chose one specific method to employ transformer models for tagging, viz. finetuning the transformer

network. As mentioned in Section 2, various alternative methods exist, and it would be interesting to compare which of them works best for our data. Similarly, for each language model we chose one pretrained transformer model, instead of comparing several of them. This, again, was in order to avoid having to run too many experiments, as well as the fact that the available transformer models for Greek and Latin differ on too many parameters (transformer architecture, data that it was trained on, tokenizer, training method etc.) so that a fair comparison could not be made.

Finally, this study was only limited to transformer-based approaches. While they are highly popular currently, there is no hard evidence that they are the best performing method for Greek and Latin morphological tagging. It would therefore be interesting to systematically investigate in the future whether they are actually the way to move forward or whether better performing approaches can be found.

Acknowledgments

Our work has been funded by grant no. G052021N of FWO/Research Council – Flanders and HBC.2021.0210 of Flanders Innovation and Entrepreneurship. We wish to thank all the annotators of the material we used in this study, as well as the three anonymous reviewers for their constructive comments which have helped to improve the quality of this paper.

References

Giuseppe G. A. Celano, Gregory Crane, and Saeed Majidi. 2016. Part of Speech Tagging for Ancient Greek. *Open Linguistics*, 2(1).

Gregory Crane. 1991. Generating and Parsing Classical Greek. *Literary and Linguistic Computing*, 6(4):243–245.

Helma Dik and Richard Whaling. 2008. Bootstrapping Classical Greek Morphology. In *Digital Humanities 2008*, pages 105–106, Oulu.

Steffen Eger, Tim vor der Brück, and Alexander Mehler. 2015. Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization models. In *Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 105-113, Beijing.

Federica Gamba and Daniel Zeman. 2023. Universalising Latin Universal Dependencies: a harmonisation of Latin treebanks in UD. In Loïc Grobol and Francis Tyers, editors, *Proceedings of the Sixth Workshop on Universal Dependencies (UDW, GURT/SyntaxFest 2023)*, pages 7–16, Washington, D.C. Association for Computational Linguistics.

Jan Hajic. 2000. Morphological Tagging: Data vs. Dictionaries. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 94-101, Seattle.

Dag Haug and Marius Jøhndal. 2008. Creating a Parallel Treebank of the Old Indo-European Bible Translations. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2008)*, pages 27–34, Marrakech.

Alek Keersmaekers. 2020. Creating a richly annotated corpus of papyrological Greek: The possibilities of natural language processing approaches to a highly inflected historical language. *Digital Scholarship in the Humanities*, 35(1):67–82.

Alek Keersmaekers. 2021. The GLAUx corpus: methodological issues in designing a long-term, diverse, multi-layered corpus of Ancient Greek. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 39–50, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels. Association for Computational Linguistics.

John Lee, Jason Naradowsky, and David A. Smith. 2011. A Discriminative Model for Joint Morphological Disambiguation and Dependency Parsing. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 885–894, Portland. Association for Computational Linguistics.

Wouter Mercelis and Alek Keersmaekers. 2022a. An ELECTRA Model for Latin Token Tagging Tasks. In Rachele Sprugnoli and Marco Passarotti, editors, *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 189–192, Marseille. European Language Resources Association.

Wouter Mercelis and Alek Keersmaekers. 2022b. electra-grc. https://huggingface.co/mercelisw/electra-grc

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille. European Language Resources Association.

Marco Passarotti, Marco Budassi, Eleonora Litta, and Paolo Ruffolo. 2017. The Lemlat 3.0 Package for Morphological Analysis of Latin. In *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pages 24–31, Gothenburg. Linköping University Electronic Press.

Céline Poudat and Dominique Longrée. 2009. Variations langagières et annotation morphosyntaxique du latin classique [Linguistic variations and morphosyntactic annotation of Latin classical texts]. In Joseph Denooz and Serge Rosmorduc, editors, *Traitement Automatique des Langues, Volume 50, Numéro 2 : Langues anciennes [Ancient Languages]*, pages 129–148, France. ATALA (Association pour le Traitement Automatique des Langues).

Frederick Riemenschneider and Anette Frank. 2023. Exploring Large Language Models for Classical Philology. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15181–15199, Toronto. Association for Computational Linguistics.

Helmut Schmid and Florian Laws. 2008. Estimation of Conditional Probabilities With Decision Trees and an Application to Fine-Grained POS Tagging. In Donia Scott and Hans Uszkoreit, editors, *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester. Coling 2008 Organizing Committee.

Pranaydeep Singh, Gorik Rutten, and Els Lefever. 2021. A Pilot Study for BERT Language Modelling and Morphological Analysis for Ancient and Medieval Greek. In *The 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, co-located with EMNLP 2021*, pages 128–137, Online. Association for Computational Linguistics.

Thea Sommerschield, Yannis Assael, John Pavlopoulos, Vanessa Stefanak, Andrew Senior, Chris Dyer, John Bodel, Jonathan Prag, Ion Androutsopoulos, and Nando de Freitas. 2023. Machine Learning for Ancient Languages: A Survey. *Computational Linguistics*, 49(3):703–747.

Milan Straka and Jana Straková. 2020. UDPipe at EvaLatin 2020: Contextualized Embeddings and Treebank Embeddings. In Rachele Sprugnoli and Marco Passarotti, editors, *Proceedings of LT4HALA 2020 - 1st Workshop on Language Technologies for Historical and Ancient Language*s, pages 124–129, Marseille. European Language Resources Association.

Elina Stüssi and Phillip Ströbel. 2024. Part-of-Speech Tagging of 16th-Century Latin with GPT. In Yuri Bizzoni, Stefania Degaetano-Ortlieb, Anna Kazantseva, and Stan Szpakowicz, editors, *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 196–206, St. Julians, Malta. Association for Computational Linguistics.

Colin Swaelens, Ilse De Vos, and Els Lefever. 2023. Evaluating Existing Lemmatisers on Unedited Byzantine Greek Poetry. In Adam Anderson, Shai Gordin, Bin Li, Yudong Liu, and Marco C. Passarotti, editors, *Proceedings of the Ancient Language Processing Workshop*, pages 111–116, Varna. INCOMA Ltd., Shoumen, Bulgaria.

Alexander Tkachenko and Kairit Sirts. 2018. Modeling Composite Labels for Neural Morphological Tagging. In Anna Korhonen and Ivan Titov, editors, *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 368–379, Brussels. Association for Computational Linguistics.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (SPMRL): what, how and whither. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 1–12, Los Angeles. Association for Computational Linguistics.

Krzysztof Wróbel and Krzysztof Nowak. 2022. Transformer-based Part-of-Speech Tagging and Lemmatization for Latin. In Rachele Sprugnoli and Marco Passarotti, editors, *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 193–197, Marseille. European Language Resources Association.

A Tagging accuracy by morphological feature

Feature	Accuracy
Person	0.999 (218532)
Voice	0.999 (218499)
Mood	0.998 (218443)
Tense	0.997 (218113)
Number	0.996 (217953)
Degree	0.995 (217745)
XPOS	0.993 (217279)
Case	0.991 (216736)
Gender	0.989 (216292)

Table 7: Tagging accuracy by morphological feature (Greek) (N=218,787)

Feature	Accuracy
Reflex	1.000 (14091)
Polarity	1.000 (14088)
Poss	1.000 (14085)
Mood	0.998 (14064)
Person	0.997 (14055)
Aspect	0.997 (14053)
VerbForm	0.997 (14049)
Voice	0.997 (14046)
Tense	0.995 (14023)
PronType	0.995 (14015)
Degree	0.993 (13991)
Number	0.992 (13980)
Case	0.989 (13934)
UPOS	0.983 (13854)
Gender	0.972 (13699)

Table 8: Tagging accuracy by morphological feature (Latin) (N=14,091)

B Tagging accuracy by text type

Text type	Accuracy
Mythography	0.994 (167/168)
Religious History	0.986 (17172/17419)
Religious Epistle	0.985 (7224/7333)
Religious Prophecy	0.983 (1686/1715)
Paradoxography	0.982 (639/651)
Religious Narrative	0.981 (254/259)
Dialogue	0.979 (1050/1072)
Biology	0.979 (94/96)
Alchemy	0.978 (391/400)
Biography	0.976 (8958/9181)
Oratory	0.975 (14905/15289)
Epistolography	0.975 (1234/1266)
Narrative	0.974 (12255/12584)
Philosophic Dialogue	0.973 (3833/3938)
Medicine	0.973 (803/825)
Epic poetry	0.973 (36641/37657)
History	0.973 (60353/62027)
Rhetoric	0.970 (2835/2924)
Geography	0.968 (1341/1385)
Polyhistory	0.966 (6460/6686)
Philosophy	0.965 (8847/9166)
Military	0.963 (2327/2417)
Tragedy	0.957 (15679/16384)
Engineering	0.951 (1402/1474)
Scientific Poetry	0.945 (52/55)
Mathematics	0.945 (240/254)
Comedy	0.944 (4085/4327)
Language	0.913 (506/554)
Lyric poetry	0.905 (1159/1281)

Table 9: Tagging accuracy by text type (Greek)

Text	Accuracy
Jerome's Vulgate	0.952 (6588/6922)
Commentarii belli Gallici	0.941 (1989/2114)
Epistulae ad Atticum	0.921 (2871/3116)
De officiis	0.921 (820/890)
Opus agriculturae	0.898 (942/1049)

Table 10: Tagging accuracy by text type (Latin)

C Supplementary Material

All the datasets used in this study can be found on https://github.com/alekkeersmaekers/transformer-tagging. The code (including the tagger settings for the experiments described here) can be found on https://github.com/alekkeersmaekers/glaux-nlp.