Proceedings of 1st Workshop on Language Technologies for Historical and Ancient Languages, pages 130–135
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Voting for POS Tagging of Latin Texts: Using the Flair of FLAIR to Better Ensemble Classifiers by Example of Latin

Manuel Stoeckel, Alexander Henlein, Wahed Hemati, Alexander Mehler
Text Technology Lab, Goehte-University Frankfurt
manuel.stoeckel@stud.uni-frankfurt.de, {henlein, hemati, mehler}@em.uni-frankfurt.de
https://www.texttechnologylab.org

Abstract

Despite the great importance of the Latin language in the past, there are relatively few resources available today to develop modern NLP tools for this language. Therefore, the EvaLatin Shared Task for Lemmatization and Part-of-Speech (POS) tagging was published in the LT4HALA workshop. In our work, we dealt with the second EvaLatin task, that is, POS tagging. Since most of the available Latin word embeddings were trained on either few or inaccurate data, we trained several embeddings on better data in the first step. Based on these embeddings, we trained several state-of-the-art taggers and used them as input for an ensemble classifier called LSTMVoter. We were able to achieve the best results for both the cross-genre and the cross-time task (90,64 % and 87,00 %) without using additional annotated data (closed modality). In the meantime, we further improved the system and achieved even better results (96,91 % on classical, 90,87 % on cross-genre and 87,35 % on cross-time).

Keywords: Part-of-Speech Tagging, Statistical and Machine Learning Methods, Corpus (Creation, Annotation, etc.)

1. Introduction

EvaLatin is the first evaluation campaign totally devoted to the evaluation of NLP tools for Latin (Sprugnoli et al., 2020). For this purpose, two tasks have been released (i.e. Lemmatization and Part of Speech (POS) tagging), each of which is divided into three subgroups: classical, cross-genre and cross-time. In this work we describe an approach to the task of EvaLatin regarding POS tagging, that is, the task of assigning each token in a text its part of speech. A part of speech is a category of words with similar grammatical properties. For many natural language processing (NLP) tasks, such as information retrieval, knowledge extraction or semantic analysis, POS tagging is a crucial pre-processing step. However, in morphologically rich languages such as Latin, this task is not trivial due to the variability of lexical forms. In order to perform POS tagging automatically, it has to be understood as a sequence labeling problem, where an output class is assigned to each input word so that the length of the input sequence corresponds to the length of the output sequence.

There already exist approaches for POS tagging for Latin (Gleim et al., 2019; vor der Brück and Mehler, 2016; Eger et al., 2016; Eger et al., 2015; Straka and Straková, 2017; Kestemont and De Gussem, 2016; Kondratyuk and Straka, 2019; Manjavacas et al., 2019). These approaches mostly utilize the increasingly popular neural network based methods for POS-tagging – by example of Latin. Part of this contribution is to extend this work and to train state-of-the-art neural network based sequence labeling tools (Straka and Straková, 2017; Lample et al., 2016; Akbik et al., 2019a; Kondratyuk and Straka, 2019) for Latin.

These neural network based sequence labeling tools usually require pre-trained word embeddings (e.g. Mikolov et al. (2013a) or Pennington et al. (2014)). These word embeddings are trained on large unlabeled corpora and are more useful for neural network sequence labeling tools if the corpora are not only large but also from the same domain as the documents to be processed. Therefore another part of this contribution is to create word embeddings for Latin for different genres and epochs. Since Latin is a morphologically rich language, sub-word-embeddings (Grave et al., 2018; Heinzerling and Strube, 2018) must be created to reflect its morphological peculiarities.

The various sequence labeling tools provide different results, making it advisable to combine them in order to bundle their strengths. For this reason LSTMVoter (Hemati and Mehler, 2019) was used to create a conglomerate of the various tools and models (re-)trained here.

To simplify the above mentioned process of training embeddings and sequence labeling tools on the one hand and creating an ensemble thereof, we developed a generic pipeline architecture which takes a labeled corpus in CoNLLU format as input, trains the different taggers and finally creates an LSTMVoter ensemble. The idea is to make this architecture available for the solution of related tasks in order to systematically simplify the corresponding training pipeline.

The article is organized as follows: Section 2 describes the data sets we used to train our word embeddings. Section 3 describes the training process of the taggers and how they were integrated into our system. In Section 4, we present and discuss our results, while Section 5 provides a summary of this study and prospects for future work.

2. Datasets

This section gives a brief overview about the datasets supplied for EvaLatin as well as other corpora we used for the *closed modality* run of the POS task.

Current state-of-the-art sequence labeling systems for POS tagging make use of word embeddings or language models (Akbik et al., 2018; Bohnet et al., 2018; Gleim et al., 2019, *LMs*). These tools are usually trained and evaluated on high-resource languages; making use of the availability of large unlabeled corpora to build feature-rich word em-

beddings. This leads to an ever-increasing ubiquitousness of embeddings for all kinds of languages.
Unfortunately, the number of available, high-quality corpora for Latin is stretched thin; historically the Latin Wikipedia has often been used as a corpus for training word embeddings (Grave et al., 2018; Heinzerling and Strube, 2018). But the Latin Wikipedia is composed of modern texts written by scholars of different backgrounds, which cannot properly reflect the use of Latin language throughout history. Thus we compiled a corpus of historical, Medieval Latin texts covering different epochs which is presented in the following section.

2.1. Historical Corpora

An overview of the corpora used is shown in table 1. It lists each corpus together with its numbers of sentences, tokens and characters and provides a summary of the overall corpus with the total number and unique counts. In addition to the corpus published for EvaLatin, we added other publicly accessible corpora: the Universal Dependencies Latin (Nivre et al., 2016a, UD_Latin) corpora UD_Latin-PROIEL (Haug and Jøhndal, 2008), UD_Latin-ITTB (Cecchini et al., 2018) and UD_Latin-Perseus (Bamman and Crane, 2011a), the Capitularies (Mehler et al., 2015) and the Cassiodorus Variae (Variae, 2020). But the main bulk of text comes from the Latin text repository of the eHumanties Desktop (Gleim et al., 2009; Gleim et al., 2012) and the CompHistSem (Cimino et al., 2015) project comprising a large number of Medieval Latin texts.[1] For all corpora we extracted the plain text without annotations and compiled a single corpus called *Historical Latin Corpus* (HLC).

Corpus	Sentences	Tokens	Chars
UD-Perseus	2 260	29 078	1 444 884
Cassiodor. Variae	3 129	135 352	748 477
EvaLatin	14 009	258 861	1 528 538
Capitularies	15 170	477 247	2 432 482
UD-PROIEL	18 526	215 175	1 157 372
UD-ITTB	19 462	349 235	1 771 905
CompHistSem	2 608 730	79 136 129	384 199 772
Total	2 665 840	80 129 332	389 576 106
Unique		971 839	434

Table 1: Plain text corpora statistics.

3. System Description

3.1. Embeddings

While there are some word embeddings and language models trained on Latin texts, these are either trained on small, but higher-quality datasets (eg. Nivre et al. (2016b), trained on the Latin part of the UD corpus; Sprugnoli et al. (2019), trained on the 1 700 000 token *Opera Latin* corpus), or larger datasets which suffer from poor OCR quality (eg. Bamman and Crane (2011b) trained on noisy data) or are of modern origin (eg. Grave et al. (2018) and Heinzerling and Strube (2018) trained on Wikipedia). Therefore we trained our own embeddings[2] on the HLC of Section 2.1 to obtain high quality word embeddings for our sequence labeling models. In the following sections we describe the type of embeddings we used and their hyperparameters adjusted during training.

3.1.1. Word Embeddings

wang2vec (Ling et al., 2015) is a variant of *word2vec* embeddings (Mikolov et al., 2013a; Mikolov et al., 2013b) which is aware of the relative positioning of context words by making a separate prediction for each context word position during training.

GloVe embeddings (Pennington et al., 2014) are trained on *global* word-word co-occurrence statistics across an entire corpus rather than considering *local* samples of co-occurrences.

3.1.2. Sub-word Embeddings

fastText embeddings (Grave et al., 2018) are trained on *character n-grams* of words rather than words themselves. They are able to capture character-based information which may be related to morphological information in addition to distributional information.

Byte-Pair Embeddings (Heinzerling and Strube, 2018, BPEmb) are composed of sub-word token embeddings. They utilize a vocabulary of character sequences which are induced from a large text corpus using a variant of byte-pair encoding for textual data (Sennrich et al., 2016). We used the *SentencePiece's*[3] implementation of the byte-pair algorithm to encode the HLC (see Section 4).

3.1.3. FLAIR Language Model

Current methods for sequence labeling use *language models* (LMs) trained on large unlabeled corpora to obtain *contextualized embeddings*, achieving state-of-the-art performance in POS tagging and named entity recognition for English, German and Dutch (Peters et al., 2018; Akbik et al., 2018). Some recent sequence labeling models with strong performance leverage *FLAIR character language models* (Akbik et al., 2018; Akbik et al., 2019b). These models are available through the *FLAIR framework* (Akbik et al., 2019a) which, since its first release, has been expanded with character language models for various languages by the NLP community, but none for Latin. Thus, we trained our own Latin character language model on the HLC of Section 2.1.

3.2. Taggers

In the following sections we briefly describe the taggers we have selected for our evaluation.

3.2.1. MarMoT

MarMoT is a generic CRF framework (Mueller et al., 2013). It implements a higher order CRF with approximations such that it can deal with large output spaces. It can also be trained to fire on predictions of lexical resources and on word embeddings.

[1] The texts are available via www.comphistsem.org or the eHumanities Desktop (hudesktop.hucompute.org).

[2] http://embeddings.texttechnologylab.org

[3] https://github.com/google/sentencepiece

3.2.2. anaGo

anaGo is a neural network-based sequence labeling system. It is based on the Glample Tagger (Lample et al., 2016), which combines a bidirectional *Long Short-term Memory* (LSTM) with *Conditional Random Fields* (CRF).

3.2.3. UDPipe

UDPipe provides a trainable pipeline for tokenization, tagging, lemmatization and dependency parsing. It offers 94 pre-trained models of 61 languages, each of which has been trained on UD Treebank (Nivre et al., 2016a) datasets. The POS model itself is based on MorphoDiTa (Straková et al., 2014) and can be easily trained on new data; no additional embeddings or features are required.

3.2.4. UDify

UDify is a single BERT-based (Devlin et al., 2018) model which was trained on 124 treebanks of 75 different languages for tagging, lemmatization and dependency parsing as well. Besides a pre-trained BERT model, the pipeline does not require any other features to be trained on new data.

3.2.5. FLAIR

Utilizing the FLAIR language model introduced above, we trained a BiLSTM-CRF sequence tagger using pooled contextualized embeddings (Akbik et al., 2019b, PCEs). PCEs are *aggregated* during the tagging process to capture the meaning of underrepresented words, which have already been seen by the tagger previously in contexts that are more specified.

3.2.6. Meta-BiLSTM

The **Meta-BiLSTM** tagger (Bohnet et al., 2018) combines two separate classifiers using a meta-model and achieves very good results on POS tagging. Each intermediate model is trained on the sequence labeling task using a different view of sentence-level representations, namely word and character embeddings. Then, a meta-model is trained on the same task while using the hidden states of the two other models as its input.

3.2.7. LSTMVoter

LSTMVoter (Hemati and Mehler, 2019) is a two-stage recurrent neural network system that integrates the optimized sequence labelers from our study into a single ensemble classifier: in the first stage, we trained and optimized all POS taggers mentioned so far. In the second stage, we combined the latter sequence labelers with two bidirectional LSTMs using an attention mechanism and a CRF to build an ensemble classifier. The idea of LSTMVoter is to learn, so to speak, which output of which embedded sequence labeler to use in which context to generate its final output.

4. Experiments

In this section we discuss our experiments and outline the parameters used to train each of the models. After the end of the task's evaluation window we were able to fine-tune our models using the gold-standard evaluation dataset. All of our experiments were conducted according to the *closed modality* of the second EvaLatin task, i.e. no additional labeled training data was used.

Tool	Classical	Cross-Genre	Cross-Time
LSTMVoterV1^{e}	93,24 %	83,88 %	81,38 %
FLAIR^{e†}	**96,34 %**	**90,64 %**	83,00 %
LSTMVoterV2^{e}	95,35 %	86,95 %	**87,00 %**
UDPipe	93,68 %	84,65 %	86,03 %
UDify	95,13 %	86,02 %	87,34 %
Meta-BiLSTM†	96,01 %	87,95 %	82,32 %
FLAIR†	96,67 %	**90,87 %**	83,36 %
LSTMVoterV3†	**96,91 %**	90,77 %	**87,35 %**

Table 2: F1-scores (macro-average) for the different test datasets. All tools were trained according to the *closed modality*. † denotes models that were trained using our embeddings, while ^{e} denotes models which were submitted during the tasks evaluation window.

4.1. Training

4.1.1. Embeddings

For each of the methods mentioned in Section 3.1.1 we created 300 dimensional word embeddings by

- setting the window size to 10 for wang2vec and training for 50 epochs,
- using default parameters in the case of fastText and by training it for 100 epochs,
- choosing a window size of 15 with default parameters for GloVe and training for 100 epochs.

We encoded the HLC by means of the byte-pair algorithm, experimented with different vocabulary sizes $c \in \{5\,000, 10\,000, 100\,000, 200\,000\}$ and trained 300 dimensional GloVe embeddings on them using the same hyperparameters for GloVe as with the plain text corpus.

For our FLAIR language model we choose our parameters according to the recommendations of Akbik et al. (2018) and set the hidden size of both forward and backward language models to 1024, the maximum character sequence length to 250 and the mini-batch size to 100. We trained the model until after 50 epochs the learning rate annealing stopped with a remaining perplexity of 2,68 and 2,71 for the forward and backward model, respectively.

4.1.2. Taggers

We trained a BiLSTM-CRF sequence tagger using FLAIR with pooled contextualized embeddings together with our language model. We added all our word and subword embeddings as features for up to 150 epochs and used learning rate annealing with early stopping. In our experiments the byte-pair embeddings with the smallest vocabulary size of 5 000 performed best. We choose one hidden LSTM layer with 256 nodes and default parameters otherwise.

The Meta-BiLSTM tagger was trained with our GloVe embeddings using default parameters. UDPipe was trained with the default settings on the data set. POS was trained independently of the lemmatizer, as this achieved better results. The UDify BERT model was also only trained on POS, while all other modules were removed. This concerned a variant of BERT-Base-Multilingual[4] which also processed Latin data.

[4] https://github.com/google-research/bert/blob/master/multilingual.md

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	SCONJ	VERB	X
Classical															
Meta	90 %	99 %	93 %	85 %	99 %	97 %	**98 %**	97 %	76 %	99 %	97 %	97 %	89 %	97 %	75 %
UDPipe	85 %	98 %	91 %	64 %	99 %	96 %	88 %	95 %	69 %	98 %	95 %	95 %	85 %	95 %	89 %
UDify	87 %	99 %	92 %	88 %	99 %	96 %	00 %	96 %	74 %	99 %	96 %	97 %	91 %	97 %	00 %
FLAIR	91 %	**99 %**	**95 %**	86 %	**99 %**	**97 %**	91 %	**97 %**	78 %	**100 %**	97 %	97 %	**93 %**	98 %	82 %
VoterV1	83 %	98 %	90 %	67 %	99 %	96 %	70 %	94 %	69 %	99 %	95 %	95 %	86 %	95 %	00 %
VoterV2	88 %	99 %	93 %	84 %	**99 %**	97 %	96 %	96 %	74 %	99 %	96 %	97 %	90 %	97 %	**95 %**
VoterV3	**91 %**	99 %	**95 %**	**88 %**	**99 %**	97 %	96 %	97 %	**78 %**	99 %	**98 %**	**98 %**	92 %	**98 %**	90 %
Cross-Genre															
Meta	79 %	96 %	85 %	57 %	97 %	94 %	77 %	90 %	67 %	97 %	96 %	80 %	75 %	91 %	—
UDPipe	69 %	93 %	80 %	13 %	**98 %**	92 %	79 %	86 %	55 %	98 %	96 %	86 %	75 %	87 %	—
UDify	73 %	97 %	80 %	50 %	98 %	89 %	00 %	88 %	55 %	**98 %**	95 %	87 %	79 %	88 %	—
FLAIR	**82 %**	97 %	**87 %**	**80 %**	98 %	**94 %**	**91 %**	**93 %**	64 %	97 %	96 %	87 %	78 %	**94 %**	—
VoterV1	66 %	95 %	81 %	29 %	98 %	92 %	70 %	86 %	**71 %**	98 %	95 %	85 %	73 %	86 %	—
VoterV2	73 %	97 %	84 %	50 %	98 %	93 %	77 %	88 %	74 %	**98 %**	96 %	86 %	78 %	89 %	—
VoterV3	79 %	**97 %**	86 %	**80 %**	98 %	93 %	80 %	92 %	71 %	**98 %**	**97 %**	**87 %**	**80 %**	93 %	—
Cross-Time															
Meta	74 %	97 %	**72 %**	42 %	**90 %**	89 %	60 %	89 %	29 %	**100 %**	84 %	65 %	70 %	86 %	—
UDPipe	70 %	97 %	68 %	36 %	90 %	89 %	50 %	93 %	97 %	**100 %**	82 %	**98 %**	72 %	86 %	—
UDify	74 %	98 %	68 %	**46 %**	90 %	87 %	00 %	**95 %**	97 %	**100 %**	85 %	93 %	**76 %**	88 %	—
FLAIR	74 %	**98 %**	71 %	44 %	90 %	86 %	75 %	90 %	50 %	**100 %**	85 %	52 %	72 %	89 %	—
VoterV1	69 %	97 %	68 %	38 %	90 %	89 %	55 %	88 %	29 %	**100 %**	81 %	55 %	70 %	86 %	—
VoterV2	73 %	98 %	69 %	43 %	90 %	89 %	**100 %**	94 %	**97 %**	**100 %**	84 %	95 %	74 %	88 %	—
VoterV3	**75 %**	98 %	**73 %**	43 %	90 %	**89 %**	46 %	94 %	96 %	**100 %**	**86 %**	81 %	74 %	**89 %**	—

Table 3: F-Scores (micro-average) for each tool per tag and dataset. Model names are abbreviated: VoterV*i* denotes LSTMVoter V*i* and Meta denotes the Meta-BiLSTM model. Bold entries mark the best values prior to rounding.

For LSTMVoter we used a 40-10-40-10 split of the training data in line with Hemati and Mehler (2019). Using the first 40-10 split, all taggers from Section 3.2 were trained and their hyperparameters were optimized. The second split was then used to train LSTMVoter and to optimize its hyperparameters. We created the following ensembles:

V1: MarMoT and anaGo.

V2: MarMoT, anaGo and UDify, UDPipe.

V3: MarMoT, anaGo, UDify, UDPipe and FLAIR.

4.2. Results

An overview of the results of our taggers is provided by Table 2, while a more detailed report listing the performance of each tool for each POS and data type is given by Table 3. The first three rows of Table 2 show our submissions during the EvaLatin evaluation window. The best model for the classical and cross-genre sub-task is the FLAIR BiLSTM-CRF tagger with 96,34 % and 90,64 % while the LSTMVoter V2 model performs best on the cross-time sub-task with 87,00 %. With these results we placed first among other closed modality EvaLatin participants for both out-of-domain tasks and second for the Classical sub-task.

With fine-tuning after the release of the gold-standard annotations (while still following closed modality rules) we were able to increase all our results significantly by means of the third variant (V3) of our LSTMVoter ensemble model, while the performance of the fine-tuned FLAIR tagger only increased marginally.

5. Conclusion

We presented our experiments and results for the EvaLatin task on POS tagging. We trained and optimized various state-of-the-art sequence labeling systems for the POS tagging of Latin texts. Current sequence labeling systems require pre-trained word embeddings. In our experiments we trained a number of such models. In the end a combination of tools, which were integrated into an ensemble classifier by means of LSTMVoter, led to the best results. The reason for this might be that the LSTMVoter combines the strengths of the individual taggers as much as possible, while at the same time not letting their weaknesses get too many chances. The best model submitted during the evaluation window for the classical and cross-genre sub-task was the FLAIR BiLSTM-CRF tagger with 96,34 % and 90,64 % while the LSTMVoter V2 model performed at this time best on the cross-time sub-task with 87,00 %. With these results we placed first among other closed modality EvaLatin participants for both out-of-domain tasks and second for the classical sub-task. With fine-tuning after the release of the gold-standard annotations we were able to increase all our results significantly with the help of LSTMVoter V3. However, it is rather likely that we reached the upper bound of POS tagging for classic texts, because the inter-annotator agreement for POS tagging seems to be limited by a number in the range of 97 %–98 % (Brants, 2000; Plank et al., 2014). Our results for cross-genre and cross-time are top performers in EvaLatin, but they still offer potential for improvements. Future work should develop models that are specialized for each genre and time period. This also regards the inclusion of additional information such as lemma-related and morphological features to a greater extent, since Latin is a morphologically rich language.

The data and the code used and implemented in this study are available at `https://github.com/texttechnologylab/SequenceLabeling`; the embeddings are available at `http://embeddings.texttechnologylab.org`. All presented tools are accessible through the TextImager (Hemati et al., 2016) interface via the GUI [5] and as REST services[6].

[5] `textimager.hucompute.org`

[6] `textimager.hucompute.org/rest/doku/`

6. Bibliographical References

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019a). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Akbik, A., Bergmann, T., and Vollgraf, R. (2019b). Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Bamman, D. and Crane, G. (2011a). The ancient greek and latin dependency treebanks. In *Language technology for cultural heritage*, pages 79–98. Springer.

Bamman, D. and Crane, G. R. (2011b). Measuring historical word sense variation. In Glen Newton, et al., editors, *Proceedings of the 2011 Joint International Conference on Digital Libraries, JCDL 2011, Ottawa, ON, Canada, June 13-17, 2011*, pages 1–10. ACM.

Bohnet, B., McDonald, R., Simões, G., Andor, D., Pitler, E., and Maynez, J. (2018). Morphosyntactic tagging with a meta-BiLSTM model over context sensitive token encodings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2642–2652, Melbourne, Australia, July. Association for Computational Linguistics.

Brants, T. (2000). Inter-annotator agreement for a German newspaper corpus. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece, May. European Language Resources Association (ELRA).

Cecchini, F. M., Passarotti, M., Marongiu, P., and Zeman, D. (2018). Challenges in converting the *Index Thomisticus* treebank into universal dependencies. *Proceedings of the Universal Dependencies Workshop 2018 (UDW 2018)*.

Cimino, R., Geelhaar, T., and Schwandt, S. (2015). Digital approaches to historical semantics: new research directions at frankfurt university. *Storicamente*, 11.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eger, S., vor der Brück, T., and Mehler, A. (2015). Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization methods. In *Proceedings of the 9th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH 2015)*, Beijing, China.

Eger, S., Gleim, R., and Mehler, A. (2016). Lemmatization and morphological tagging in German and Latin: A comparison and a survey of the state-of-the-art. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC 2016.

Gleim, R., Waltinger, U., Ernst, A., Mehler, A., Esch, D., and Feith, T. (2009). The eHumanities Desktop – an online system for corpus management and analysis in support of computing in the humanities. In *Proceedings of the Demonstrations Session of the 12th Conference of the European Chapter of the Association for Computational Linguistics EACL 2009, 30 March – 3 April, Athens*.

Gleim, R., Mehler, A., and Ernst, A. (2012). SOA implementation of the eHumanities Desktop. In *Proceedings of the Workshop on Service-oriented Architectures (SOAs) for the Humanities:Solutions and Impacts, Digital Humanities, Hamburg*.

Gleim, R., Eger, S., Mehler, A., Uslu, T., Hemati, W., Lücking, A., Henlein, A., Kahlsdorf, S., and Hoenen, A. (2019). A practitioner's view: a survey and comparison of lemmatization and morphological tagging in german and latin. *Journal of Language Modeling*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Haug, D. T. and Jøhndal, M. (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.

Heinzerling, B. and Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

Hemati, W. and Mehler, A. (2019). LSTMVoter: chemical named entity recognition using a conglomerate of sequence labeling tools. *Journal of Cheminformatics*, 11(1):7, Jan.

Hemati, W., Uslu, T., and Mehler, A. (2016). Textimager: a distributed uima-based system for nlp. In *Proceedings of the COLING 2016 System Demonstrations*. Federated Conference on Computer Science and Information Systems.

Kestemont, M. and De Gussem, J. (2016). Integrated sequence tagging for medieval latin using deep representation learning. *Journal of Data Mining and Digital Humanities*.

Kondratyuk, D. and Straka, M. (2019). 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, abs/1603.01360.

Ling, W., Dyer, C., Black, A. W., and Trancoso, I. (2015). Two/too simple adaptations of word2vec for syntax problems. In Rada Mihalcea, et al., editors, *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1299–1304. The Association for Computational Linguistics.

Manjavacas, E., Kádár, Á., and Kestemont, M. (2019). Improving lemmatization of non-standard languages with joint learning. *CoRR*, abs/1903.06939.

Mehler, A., vor der Brück, T., Gleim, R., and Geelhaar, T. (2015). Towards a network model of the coreness of texts: An experiment in classifying latin texts using the ttlab latin tagger. In Chris Biemann et al., editors, *Text Mining: From Ontology Learning to Automated text Processing Applications*, Theory and Applications of Natural Language Processing, pages 87–112. Springer, Berlin/New York.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. In Yoshua Bengio et al., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

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.

Nivre, J., De Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., et al. (2016a). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Nivre, J., de Marneffe, M., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016b). Universal dependencies v1: A multilingual treebank collection. In Nicoletta Calzolari, et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Plank, B., Hovy, D., and Søgaard, A. (2014). Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 742–751, Gothenburg, Sweden, April. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Sprugnoli, R., Passarotti, M., and Moretti, G. (2019). Vir is to moderatus as mulier is to intemperans lemma embeddings for latin. In *Sixth Italian Conference on Computational Linguistics*, pages 1–7. CEUR-WS. org.

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).

Straka, M. and Straková, J. (2017). Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada, August. Association for Computational Linguistics.

Straková, J., Straka, M., and Hajic, J. (2014). Open-source tools for morphology, lemmatization, pos tagging and named entity recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18.

Variae. (2020). Latin Text Archive (LTA) Version of the CompHistSem Working Group of the Corpus "Variae" by Flavius Magnus Aurelius Cassiodorus based on Cassiodori Senatoris Variae, rec. Theodorus Mommsen, Berlin: Weidmann, 1894 (MGH Auct. Ant. 12). Retrieved from the critical edition and prepared by the BMBF project "Humanist Computer Interaction under Scrutiny" (https://humanist.hs-mainz.de/en/). Available at https://www.comphistsem.org/texts.html.

vor der Brück, T. and Mehler, A. (2016). TLT-CRF: A lexicon-supported morphological tagger for Latin based on conditional random fields. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, LREC 2016.