Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization methods

CONFERENCE PAPER · JULY 2015

3 AUTHORS:

Steffen Eger
Goethe-Universität Frankfurt am Main
19 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Tim vor der Brück
Lucerne University of Applied Sciences and …
32 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Alexander Mehler
Goethe-Universität Frankfurt am Main
123 PUBLICATIONS **553** CITATIONS
SEE PROFILE

Lexicon-assisted tagging and lemmatization in Latin: A comparison of six taggers and two lemmatization methods

Steffen Eger
Text Technology Lab
Universität Frankfurt

Tim vor der Brück
Text Technology Lab
Universität Frankfurt

Alexander Mehler
Text Technology Lab
Universität Frankfurt

{steeger,vorderbr,mehler}@em.uni-frankfurt.de

Abstract

We present a survey of tagging accuracies — concerning part-of-speech and full morphological tagging — for several taggers based on a corpus for medieval church Latin (see www.comphistsem.org). The best tagger in our sample, Lapos, has a PoS tagging accuracy of close to 96% and an overall tagging accuracy (including full morphological tagging) of about 85%. When we 'intersect' the taggers with our lexicon, the latter score increases to almost 91% for Lapos. A conservative assessment of lemmatization accuracy on our data estimates a score of 93-94% for a lexicon-based lemmatization strategy and a score of 94-95% for lemmatizing via trained lemmatizers.

1 Introduction

Part-of-speech (PoS) tagging is a standard task in natural language processing (NLP) in which the goal is to assign each word in a sentence its (possibly complex) part-of-speech label. While part-of-speech tagging for English is well-researched, morphologically rich languages like some Slavic languages or classical languages such as ancient Greek or Latin have received considerably less attention. Often-cited problems for the latter class of languages include relatively free word-order and a high degree of inflectional variability, leading to data sparseness problems.

In this work, we survey tagging accuracies (part-of-speech as well as full morphological tagging) for several part-of-speech taggers based on a corpus of Latin texts.

The *corpus*, which was built as part of the *Computational Historical Semantics* (CompHistSem) project[1], comprises about 15 500 sentences as exemplified in Table 1. The aim of CompHistSem is to develop an historical semantics based on medieval Latin texts that allows for fine-grained analyses of word meanings starting from richly annotated corpora. The application scenario of the current study is to meet this annotation requirement by means of open access tools.

Our corpus is based on the capitularies, the amalarius corpus as partly available via the Patrologia Latina[2] and three further texts from the MGH[3] corpus (Visio Baronti, Vita Adelphii, Vita Amandi). Each token of the corpus has been manually annotated with a reference to an associated lexicon entry as described below (cf. Mehler et al. (2015)). In this way, full morphological features are available for all tokens. Our *lexicon* has been compiled from several sources such as LemLat and from rule-based lexical expanders. We describe its composition in more depth in Section 2.

The *taggers* we survey include three relatively new taggers (Lapos, Mate, and the Stanford tagger) as well as two taggers originating in an earlier tagging tradition (TnT, TreeTagger). In addition, we report results for two tagger variants available in the OpenNLP package. All taggers are trained on our corpus. In accordance with Moore's law describing scientific/technological progress over time, we find that more recent tagger classes substantially outperform their predecessor generation. The best tagger in our sample, Lapos, has a PoS tagging accuracy of close to 96% and an overall tagging accuracy (including full morphological tagging) of about 85%. When we 'intersect' the taggers with our lexicon, the latter score increases to almost 91% for Lapos. Concerning lemmatization, we lemmatize words on the basis of the taggers' outputs. We employ two dif-

[1] www.comphistsem.org

[2] http://patristica.net/latina

[3] MGH is the acronym of Monumenta Germaniae Historica, the German Central Institute for Middle Age research (deutsches Zentralinstitut zur Erforschung des Mittelalters).

Proceedings of the 9th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities, pages 105–113,
Beijing, China, July 30, 2015.

Form	Lemma	PoS-tag	Sub.-cats.
Ex	ex	AP	
frugibus	frux	NN	gender=f, case=abl. number=pl
terrae	terra	NN	gender=f, case=gen. number=sg,
corpus	corpus	NN	gender=n, case=nom. number=sg,
nostrum	noster	PRO	gender=n, case=nom. number=sg,
sustentatur	sustento	V	number=sg, person=3 mood=ind, voice=pass. tense=present

Table 1: Sample sentence ('from the fruits of the earth our body is sustained') in our corpus and its annotation.

ferent lemmatization strategies: we either look up the current lemma in the lexicon given the word form as well as the predicted tag information (lexicon-based lemmatization) or we lemmatize on the basis of statistical lemmatizers/string transducers trained on our corpus. A conservative assessment of lemmatization accuracy estimates a score of 93-94% for the lexicon-based strategy and a score of 94-95% for the trained lemmatizers.

This work is structured as follows. Section 2 describes our lexicon. Section 3 outlines related work, on part-of-speech tagging and resources for Latin. Section 4 describes our lemmatization module and Section 5 the tagging systems we survey. In Section 6, we outline results and we conclude in Section 7.

2 Lexicon

Our lexicon named Collex.LA (Mehler et al., 2015) consists both of manually created lexicon entries as well as of automatically extracted entries from several freely available Web resources, in particular AGFL (Koster and Verbruggen, 2002), LemLat (Passerotti, 2004), Perseus Digital Library (Smith et al., 2000), Whitaker word list[4], Thomisticum[5] (Busa, 1980; McGilivray et al., 2009), Ramminger word list[6], and several others. In total it consists of 8 347 062 word forms, 119 595 lemmas and 104 905 superlemmas.[7] A superlemma is a special kind of lemma that unifies several writing variants. The lexicon distribution over different parts of speech is given in Table 2. Each lexicon entry consists of word form, part-of-speech, and lemma. Depending on the part-of-speech of the entry, additional grammatical features can be provided. For instance, each verb entry contains its mood, voice, number, person, verb type (transitive or intransitive), tense and the conjugation class. Pronouns are annotated with a pronoun type that further differentiates pronouns into demonstrative, interrogative, personal, reflexive, relative, possessive, indefinite, intensive, and correlative pronouns. Analogously, additional grammatical features are provided for nouns, adverbs and adjectives. In total, there are currently 17 different grammatical features defined. Our lexicon can be accessed via the website `collex.hucompute.org`.

3 Related work

PoS tagging is a long-standing NLP task and (modern) classical approaches to solving it include Hidden Markov models, conditional random fields (CRFs), averaged perceptrons, structured SVMs, and max margin Markov networks (Nguyen and Guo, 2007). For highly inflectional languages, the problem of large tagsets arises, which leads to serious data sparsity issues, besides tractability problems. Tufis (1999) addresses this via a multi-stage tagging approach in which tagging is initially performed with a reduced tagset. Müller et al. (2013) show that even higher-order CRFs can be used for large tagsets when approximations are employed. Boros et al. (2013) use feed forward neural networks, which can arguably better smooth probabilities, for this problem. In a non-contextual task setting, Toutanova and Cherry (2009) show that, for morphologically rich languages, *lemmatization* and part-of-speech tagging may mutually

[4]URL: `http://archives.nd.edu/whitaker/dictpage.htm`

[5]URL: `http://www.corpusthomisticum.org/tl.html`

[6]`http://www.neulatein.de`

[7]The lexicon is currently extended by additionally exploring the Latin Wiktionary as a resource.

Part-of-speech	#Word forms	#Lemmas	#Superlemmas
verb (V)	4 646 369	11 556	8 666
adjective (ADJ)	2 693 333	24 020	21 155
normal noun (NN)	654 194	40 906	34 096
anthroponym (NP)	229 299	26 241	25 898
named entity (NE)	68 276	5 387	4 821
adverb (ADV)	40 771	10 625	9 594
pronoun (PRO)	6 377	139	113
ordinal number (ORD)	3 349	116	87
cardinal number (NUM)	1 835	104	75
distributive number (DIST)	1 216	44	44
foreign material (FM)	1 023	91	32
conjunction (CON)	383	122	103
preposition (AP)	341	104	87
interjection (ITJ)	199	110	109
non word (XY)	69	14	14
particle (PTC)	28	16	11

Table 2: Distribution of the lexicon entries over the different parts of speech.

inform each other. Lee et al. (2011) show that tagging and *dependency parsing* may mutually inform each other in such a setup, too.

Concerning lexical resources for Latin, to our knowledge, there are concurrently three freely available resources for Latin: Perseus (Smith et al., 2000; Bamman and Crane, 2007), Proiel (Haug and Jøhndal, 2008), and the Index Thomisticus (IT) (Busa, 1980; McGilivray et al., 2009). Perseus and Proiel cover the more classical Latin era, while IT focuses on the writings of Thomas Aquinas. All resources indicate lemma and various part-of-speech information for its tokens. IT in addition provides dependency information. Concerning size, Perseus is the smallest resource with roughly 3 500 sentences, and Proiel and IT each contain about 13 000–14 000 Latin sentences.

4 Lemmatization

On our corpus, we learn a character-level string transducer as a component model of our tagger. This lemmatizer is trained on pairs of strings $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x}$ is a full form (e.g., *amavisse* 'have loved') and $\mathbf{y}$ its corresponding lemma (e.g., *amo* 'love'). Learning a statistical lemmatizer has the advantage that it can cope with OOV words and may adapt to the distribution of the corpus. Our lemmatization module is LemmaGen (Juršič et al., 2010). LemmaGen learns 'if-then' rules from $(\mathbf{x}, \mathbf{y})$ pairs as indicated. To transduce/lemmatize a new input form, rules (and their exceptions) are ordered, and the first condition that is satisfied fires the corresponding rule.

5 Part-of-speech taggers

Here, we briefly sketch the taggers we survey in Section 6. All taggers outlined are language-independent and general-purpose taggers.

The **TreeTagger** (Schmid, 1994) implements a tagger based on decision trees. Despite its simple architecture, it seems to enjoy considerable popularity up until recently. Concurrently, two freely available TreeTagger taggers for Latin are available.[8] **TnT** (Brants, 2000) implements a trigram Hidden Markov tagger with a module for handling unknown words. It has been shown to perform similarly well as maximum entropy models. **Lapos** (Tsuruoka et al., 2011) is a 'history based' tagging model (this model class subsumes maximum entropy Markov model) incorporating a lookahead mechanism into its decision-making process. It has been reported to be competitive with globally optimized models such as CRFs and structured perceptrons. **Mate** (Bohnet and Nivre, 2012) implements a transition based system for joint part-of-speech tagging and dependency parsing reported to exhibit high performance for richly

[8] See `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`.

inflected languages, where there may be considerable dependence between morphology and syntax, as well as for more configurational languages like English. The **OpenNLPTagger** is an official Apache project and provides three different tagging methods: *maximum entropy*, *perceptron* and *perceptron sequence* (cf. (Ratnaparkhi, 1996; Collins, 2002)) for maximum/perceptron based entropy tagging). We evaluated the maximum entropy and the perceptron approach.[9]

The **Stanford tagger** (Toutanova et al., 2003) implements a bidirectional log-linear model that makes broad use of lexical features. The implementation lets the user specifically activate and deactivate desired features.

We use default parametrizations for all taggers[10] and trained all taggers on a random sample of our data of about 14 000 sentences and test them on the remainder of about 1 500 sentences.

6 Results

6.1 Tagging

Contrary to some of our related work, we view the morphological tagging problem for Latin as a multi-label tagging problem in which each tagging task (PoS, case, gender, etc.) is handled *independently*. To compensate for this naïvety, we subsequently 'intersect' the resulting tag decisions with our lexicon, which considerably improves performance, as we show.

Table 3 shows accuracies (fraction of correctly tagged words) on each tagging subtask. The almost consistently best tagger is Lapos, with a slight margin over Mate and the Stanford tagger. TnT's and particularly OpenNLP's and the TreeTagger's performance are substantially worse. For example, overall tagging accuracy (indicating the probability that a system is jointly correct on *all* subtasks) of Lapos is about 2.9% higher than that of TnT and about 6.6% higher than that of the TreeTagger. When we 'intersect' the taggers' outputs with our lexicon — i.e., we retrieve the closest lexicon classification for the input form in question if the form is in the lexicon[11] — all performance values increase substantially, on the order of about 5-6 percentage points (see Table 3). Individual increases (for Lapos) for each subtask are outlined in Table 5.[12]

Figure 1 shows the learning curve (accuracy as a function of training set size) for the three selected taggers Lapos, Mate, and the TreeTagger for the category 'PoS' (similar curves for the other tagging subtasks). Apparently, the more recent tagger generation generalizes substantially better than the older approaches, exhibiting much higher accuracies especially at small training set sizes.

6.2 Lemmatization

Lemma accuracy is indicated in Table 4. As we mentioned, we employ two lemmatization strategies based on the taggers' outputs: either the lemma is retrieved from the lexicon given the predicted part-of-speech and the morphological tags. Alternatively, we train LemmaGen string transducers as outlined in Section 4, one for each part-of-speech. Once the taggers have predicted a part-of-speech we apply the corresponding lemmatizer for this word-class. Note that both strategies tendentially imply a loss of accuracy due to errors committed in a previous step, viz., tagging; however, even a falsely tagged form may receive correct lemmatization, e.g., when tag mismatch is between 'neighboring' parts-of-speech such as noun and proper noun. We find that, across the different taggers, lemma accuracy is about 93-94% for the lexicon based strategy and about 94-95% for the learned lemmatizers. Scores for the lexicon are lower, e.g., because the lexicon can simply not store all sorts of lemma information (e.g., numbers such as '75', '76', etc.), which is an instance of the OOV problem.[13] Moreover, the lexicon tends to suffer more strongly from free lemma variations (e.g., *honos* and *honor* as equivalent alternatives). In contrast, the learned lemmatizers can adapt to the actual form-lemma distribution in the respective corpus. Due to the free variation problem as indicated and since we also count lower/upper-

[9] Unfortunately, the documentation of these methods is not very detailed, which leaves the methodology of the tagger rather unclear. The application of the *sequence perceptron* method led to an exception during the training phase. Therefore, this method could not be evaluated.

[10] For the Stanford tagger, we include the features bidirectional5words, allwordshapes(-1,1), generic, words(-2,2), suffix(8), biwords(-1,1).

[11] We measure closeness in terms of the number of matching categories.

[12] We note that a simple majority vote additionally slightly increases performance values. Integrating in this way Lapos, Mate and the Stanford Tagger leads to a PoS accuracy of 95.97%; adding TnT leads to 95.94%; finally, integrating all systems leads to 95.88%.

[13] E.g., for Lapos, adding a rule for numbers increases accuracy to 94.61% for the lexicon-based lemmatization.

	Lapos	TnT	Mate	TreeTagger	Stanford	OpenNLP	
						Max.Entr	Perceptron
PoS	**95.86**	95.16	95.67	92.00	95.55	93.83	92.92
case	**94.64**	92.86	94.56	88.17	94.58	90.71	90.23
degree	**97.55**	97.09	97.40	92.40	97.30	95.55	94.52
gender	**96.09**	95.35	95.84	90.64	95.83	93.81	92.57
mood	**98.28**	97.73	98.13	93.33	98.12	96.04	94.55
number	97.19	96.90	97.04	95.16	**97.23**	95.52	94.92
person	99.25	98.87	**99.27**	94.07	99.18	97.51	95.64
tense	**98.53**	98.17	98.41	93.60	98.43	96.68	95.34
voice	**98.79**	98.52	98.74	94.43	98.67	97.95	96.93
OVERALL	**85.03**	82.63	84.25	79.71	84.35	78.16	75.87
OVERALL+LEX	**90.74**	88.33	90.55	86.38	90.29	84.58	84.03

Table 3: Tag accuracies in % for different systems and different categories.

case differences as errors, the reported numbers may be seen as conservative estimates of lemma accuracy.

Cat.	Acc.	Increase
PoS	96.10	+0.25
case	94.79	+0.15
degree	97.85	+0.30
gender	96.40	+0.32
mood	98.71	+0.47
number	97.89	+0.72
person	99.45	+0.20
tense	98.90	+0.37
voice	99.10	+0.31

Table 5: Tag accuracies in % for Lapos+Lexicon. The column 'Increase' indicates the increase over not consulting the lexicon.

6.3 Error analysis

Table 6 shows a fine-grained precision and recall analysis for Lapos, across each of the possible part-of-speech labels in our tagset (for the category 'PoS'), indicating that among the frequent parts-of-speech particularly adjectives (ADJ) and proper names (NE and NP) are hard to classify.

Table 7 shows the agreements in PoS prediction for the taggers of our test scenario. The agreement between the best-performing taggers Mate and Lapos is very high (98%), while the agreement of the low performing taggers to all other taggers is rather low (mostly below 95%). This is the case even when the latter taggers are compared among each other, which indicates that they commit quite different types of errors.

PoS	Precision	Recall	F_1
NN	95.89	95.50	95.69
V	96.81	96.61	96.71
CON	98.30	97.17	97.73
PRO	98.05	96.22	97.13
$,	100.00	100.00	100.00
AP	98.38	95.33	96.83
ADJ	83.95	88.07	85.96
$.	100.00	100.00	100.00
ADV	88.59	93.91	91.17
NUM	97.00	97.59	97.29
NP	92.87	84.49	88.48
NE	67.56	82.41	74.25
$(	100.00	98.22	99.10
FM	80.89	94.77	87.28
ORD	82.29	75.23	78.60
ITJ	78.26	100.00	87.80
XY	73.33	84.61	78.57
PTC	0.00	0.00	0.00
DIST	33.33	100.00	50.00

Table 6: Precision, recall and F_1 measure across the possible PoS tags in our corpus. PoS ordered by corpus frequency.

	Lapos	TnT	Mate	TreeTagger	Stanford
Lexicon	93.87	93.74	**93.90**	93.49	93.85
LemmaGen	**95.30**	94.85	95.06	94.74	94.99

Table 4: Lemma accuracy in % for 5 selected taggers based on either lexicon-based lemmatization or using the learned LemmaGen transducer.

	Lapos	TnT	Mate	Tree-Tagger	Stanford	OpenNLP	
						Max.Entr	Perceptron
Lapos	100	97	98	94	98	96	94
TnT	97	100	97	95	97	96	94
Mate	98	97	100	93	97	95	94
Tree-Tagger	94	95	93	100	93	92	91
Stanford	98	97	97	93	100	95	94
Op.NLP/MaxEntr.	96	96	95	92	95	100	95
Op.NLP/Perceptron	94	94	94	91	94	95	100

Table 7: Agreement of different taggers in %.

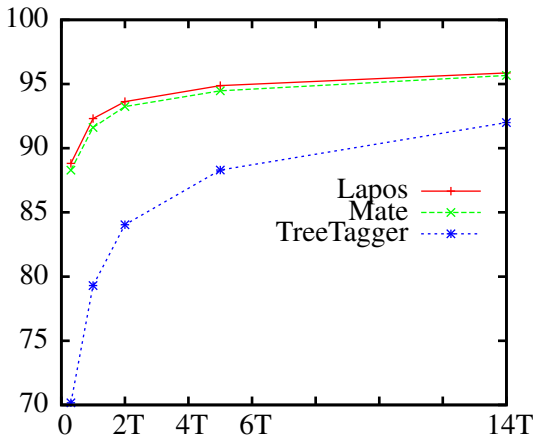

Figure 1: Accuracy as a function of training set size (300, 1000, 2000, and 5000 sentences) for Lapos, Mate, and the TreeTagger.

Our evaluation showed that in 98.90% of the cases at least one of the taggers predicted the correct part-of-speech (oracle prediction), indicating that a tagger combination could theoretically lead to accuracy values far above the 95.86% of the best performing system Lapos.

We further investigate the distribution of errors common to all taggers, shown in Figure 2.

Our analysis shows that prepositions are often confused with adverbs, because several Latin word forms can be prepositions in one context and adverbs in another. Since a preposition is almost always attached to a noun, and an adverb almost always to a verb, one possible approach to overcome this problem could be to estimate attachment probabilities of words by analyzing large Latin corpora.

A further common error is that the part-of-speech tags for nouns, adjectives and pronouns are frequently confounded by the taggers, since the associated word endings are similar and quite a few word forms are homographs with both an adjective and noun reading. In addition, the word order in Latin is relatively free. Thus, an adjective can follow or precede the modified noun, which impedes a disambiguation by statistical context analysis.

Verbs are sometimes erroneously classified as nouns, due the fact that gerund forms, annotated as verbs in the corpus, can syntactically function as nouns and have strong ending similarity with nouns.

Analogously to PoS tagging, errors in morphological tagging can occur, if the same word form can be associated to different morphological feature values of the same type, which is the case for quite a lot word forms in ablative and dative as well as for word forms in accusative and nominative plural.

Finally, some words in our corpus are annotated inconsistently. For example, ordinal numbers are sometimes tagged as adjective instead with the tag ORD that is actually intended for such numbers.

6.4 Comparison with other work

Several other papers document PoS tagging accuracies for Latin corpora. For example, Bamman

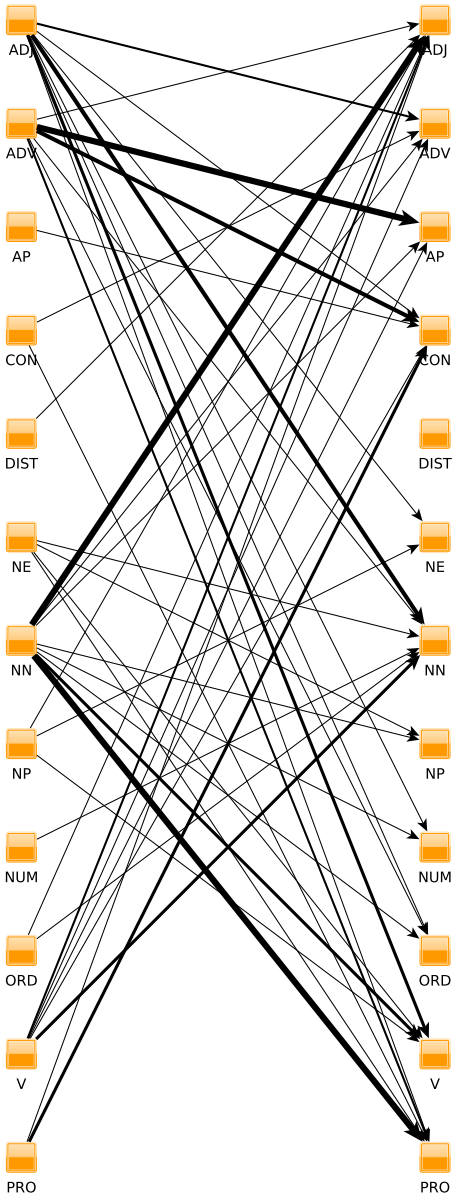

Figure 2: A bipartite graph showing the distribution of errors common to all taggers. The partition with the correct parts-of-speech are on the left side of the figure while the erroneously predicted parts of speech are displayed on the right side. The thickness of an arrow leading from the correct part-of-speech to the incorrectly predicted part-of-speech is proportional to the number of times that such an error was made by the taggers.

and Crane (2008) report a PoS tagging accuracy of 95.11% and full morphological analysis accuracy of 83.10% for the TreeTagger on Perseus. Passarotti (2010) indicates numbers of 96.75% and 89.90%, respectively, on the IT data base using an HMM-based tagger. Lee et al. (2011) introduce a joint model for morphological disambiguation and dependency parsing, achieving a PoS accuracy of 94.50% on Perseus. Müller and Schütze (2015) give a best result of 88.40% for full morphological analysis on Proiel, using a second-order CRF and features firing on the suggestions of a morphological analyzer. Of course, none of these results are directly comparable — not only because different variants of Latin are considered but also because training set sizes and annotation standards differ across corpora. For instance, while Perseus has 12 different PoS labels, our corpus has 19, making PoS tagging a priori more difficult on our corpus in this respect, irrespective of which tagging technology is employed.

7 Conclusion

We have presented a comparative study of taggers for preprocessing (medieval church) Latin. More specifically, we applied six different part-of-speech taggers to our data and surveyed their performance. This showed that the accuracy values of recent taggers barely differ on our data and take values tightly below 96% for part of speech and around 90% for full lexicon-supported morphological tagging on our test corpus. We showed that consolidating the taggers' outputs with our lexicon can substantially increase full morphological tagging performance, indicating the value of our lexical resource for addressing the problem of rich morphology in Latin. We also surveyed lemma prediction accuracy based on the taggers' outputs and found it to be on the order of around 93-94% for a lexicon-based strategy and on the order of around 94-95% for learned string transducers. Finally, we conducted a detailed error analysis that showed that all of the taggers had problems to disambiguate between prepositions and adverbs as well as between nouns and adjectives. We hope that our survey may serve as a guideline for other researchers. In future work, we intend to investigate how our results generalize to other variants of Latin. Moreover, all trained taggers presented here are made available via the website `https://prepro.hucompute.org/`. This

also concerns our training corpus that will be made available in a way that respects copyright while allowing taggers to be trained thereon.

References

David Bamman and Gregory Crane. 2007. The latin dependency treebank in a cultural heritage digital library. In *Proceedings of the Second Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)*, pages 33–40, Prague. Association for Computational Linguistics.

David Bamman and Gregory Crane. 2008. Building a dynamic lexicon from a digital library. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 11–20, New York, NY, USA. ACM.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1455–1465, Jeju Island, Korea, July. Association for Computational Linguistics.

Tiberiu Boros, Radu Ion, and Dan Tufis. 2013. Large tagset labeling using feed forward neural networks. case study on romanian language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 692–700. Association for Computational Linguistics.

Thorsten Brants. 2000. Tnt: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 224–231, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Busa. 1980. The annals of humanities computing: The index thomisticus. *Computers and the Humanities*, 14:83–90.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, Pennsylvania.

Dag Trygve Truslew Haug and Marius Jøhndal. 2008. Creating a parallel treebank of the old indo-european bible translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.

Matjaž Juršič, Igor Mozetič, and Nada Lavrač. 2010. LemmaGen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science*, 16:1190–1214.

Cornelis H.A. Koster and E. Verbruggen. 2002. The agfl grammar work lab. In *Proceedings of FREENIX/Usenix*, pages 13–18.

John Lee, Jason Naradowsky, and David A. Smith. 2011. A discriminative model for joint morphological disambiguation and dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 885–894, Stroudsburg, PA, USA. Association for Computational Linguistics.

Barbary McGilivray, Marco Passarotti, and Paolo Ruffolo. 2009. The index thomisticus treebank project: Annotation, parsing and valency lexicon. *Traitement Automatique des Langues*, 50(2).

Alexander Mehler, Tim vor der Brück, Rüdiger Gleim, and Tim Geelhaar. 2015. Towards a network model of the coreness of texts: An experiment in classifying latin texts sing the TTLab Latin tagger. In Chris Biemann and Alexander Mehler, 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.

Thomas Müller and Hinrich Schütze. 2015. Robust morphological tagging with word representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 526–536, Denver, Colorado, May–June. Association for Computational Linguistics.

Thomas Müller, Helmut Schmid, and Hinrich Schütze. 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.

Nam Nguyen and Yunsong Guo. 2007. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 681–688, New York, NY, USA. ACM.

Marco Passarotti. 2010. Leaving behind the less-resourced status. the case of Latin through the experience of the Index Thomisticus Treebank. In Kepa Sarasola, Francis M. Tyers, and Mikel L. Forcada, editors, *7th SaLTMiL Workshop on Creation and use of basic lexical resources for less-resourced languages*, pages 27–32.

Marco Passerotti. 2004. Development and perspectives of the latin morphological analyser lemlat. *Linguistica Computazionale*, 20–21.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, Pennsylvania.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

David A. Smith, Jeffrey A. Rydberg-Cox, and Gregory R. Crane. 2000. The perseus project: a digital library for the humanities. *Literary and Linguist Computing*, 15(1).

Kristina Toutanova and Colin Cherry. 2009. A global model for joint lemmatization and part-of-speech prediction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 486–494, Stroudsburg, Pennsylvania. Association for Computational Linguistics.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 173–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Kazama. 2011. Learning with lookahead: Can history-based models rival globally optimized models? In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 238–246, Stroudsburg, PA, USA. Association for Computational Linguistics.

Dan Tufis. 1999. Tiered tagging and combined language models classifiers. In Vclav Matousek, Pavel Mautner, Jana Ocelkov, and Petr Sojka, editors, *TSD*, volume 1692 of *Lecture Notes in Computer Science*, pages 28–33. Springer.