Improving Latvian Morphosyntactic Parsing with Pretrained Encoders and Analyzer-Constrained Decoding

Arturs Znotins
Institute of Mathematics and Computer Science, University of Latvia
arturs.znotins@lumii.lv

Abstract

We present a systematic evaluation of Latvian morphosyntactic parsing with pretrained transformer encoders in a unified joint architecture for tagging, lemmatization, and dependency parsing. We benchmark multilingual and Latvian-specific models and show that language-specific adaptation, even with modest in-language data, substantially improves performance. We further demonstrate that factored morphological modeling improves robustness and that integrating a Latvian morphological analyzer through constrained decoding yields consistent gains in XPOS tagging and lemmatization. The best system achieves new state-of-the-art results, reaching 95.22% XPOS accuracy, 98.72% lemma accuracy, and 93.19% LAS.

Keywords: Latvian, morphosyntactic parsing, tagging, parsing, encoders, morphological analyzer

1. Introduction

Morphosyntactic parsing—joint part-of-speech tagging with morphological features and dependency parsing—is a core task in natural language processing (NLP). Accurate morphosyntactic annotation is particularly important for morphologically rich languages, where syntactic relations and lexical meaning are strongly influenced by inflectional morphology.

Latvian is a highly inflected Baltic language whose grammatical structure exhibits complex interactions between morphology and syntax, making automatic morphosyntactic analysis both challenging and practically valuable. In Latvian linguistic practice, annotation typically relies on a language-specific tagset (XPOS) that encodes detailed morphological distinctions beyond the universal UPOS and FEATS layers defined in the Universal Dependencies (UD) framework. The Latvian XPOS tagset contains 34 attributes and roughly 1,800 distinct tags, enabling detailed linguistic analysis but also introducing substantial sparsity for statistical and neural models.

Transformer-based encoder models have substantially advanced morphosyntactic analysis. Multilingual models such as mBERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) provide strong contextual representations, but their shared capacity across languages can limit the modeling of language-specific patterns, especially for morphologically rich or lower-resource languages. Continued pretraining on in-language data has therefore been shown to improve downstream performance.

Another challenge arises from the large label space of detailed morphological tagsets. Predicting full morphosyntactic tags as monolithic labels leads to sparsity problems, particularly for richly inflected languages. Factored modeling approaches that decompose tags into individual attributes have therefore been proposed to improve generalization (Tkachenko and Sirts, 2018; Keersmaekers and Mercelis, 2024). In addition, lexicon-based morphological analyzers can provide linguistically valid candidate analyses that guide neural predictions, especially for rare or ambiguous forms.

In this work, we study Latvian morphosyntactic parsing with pretrained transformer encoders in a unified joint framework, focusing on language adaptation, factored morphology, and analyzer-constrained decoding.

Our contributions are as follows:

- We evaluate pretrained encoder models for Latvian joint morphosyntactic tagging and dependency parsing.
- We show that Latvian-specific adaptation achieves strong performance even with limited pretraining data.
- We show that factored modeling of morphological features improves robustness over monolithic tag prediction, especially in limited-data settings.
- We show that integrating a Latvian morphological analyzer yields gains even for strong pretrained models.

The resulting system achieves new state-of-the-art performance for Latvian morphosyntactic parsing while providing practical tools for large-scale corpus annotation and linguistic research. The trained models and source code are publicly available.[1]

[1] https://github.com/LUMII-AILab/lvnlp

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 11724–11734
11–16 May 2026.

2. Related Work

Latvian morphosyntactic processing has traditionally relied on hybrid systems combining rule-based linguistic resources with statistical disambiguation. The most widely used tool, LVTagger (Paikens et al., 2013), integrates a lexicon-based morphological analyzer with a conditional Markov model (CMM) disambiguator and remains a strong baseline for XPOS tagging and lemmatization, achieving 93.44% and 98.41% accuracy. The system is actively maintained and integrated into pipelines such as NLP-PIPE (Znotins and Cirule, 2018). Later work explored neural taggers using analyzer-informed features and BERT-based models for Latvian UD parsing (Paikens, 2016; Znotins and Barzdins, 2020).

A central resource for Latvian morphology is the lexicon-based analyzer described by Paikens et al. (2024), which links word forms to lemmas and full morphological analyses using inflectional paradigms and stem-change rules. The analyzer covers more than 118,000 lexemes derived primarily from the Latvian online dictionary *Tēzaurs.lv* (Spektors et al., 2016) and provides candidate analyses that can guide or constrain neural predictions.

The Latvian Treebank (LVTB) was originally annotated using a hybrid dependency–constituency grammar model (Barzdins et al., 2007; Nespore et al., 2010; Pretkalnina et al., 2011) and later converted to the Universal Dependencies (UD) framework (Pretkalnina et al., 2018). With Latvian included in UD (Zeman et al., 2021), the language has become accessible for multilingual NLP research and can be processed by general-purpose toolkits such as UDPipe (Straka, 2018), Stanza (Qi et al., 2020), UDify (Kondratyuk and Straka, 2019), and Trankit (Nguyen et al., 2021).

Recent work has demonstrated strong gains from pretrained transformer encoders, but prior studies show that monolingual or language-adaptive pretraining often improves performance for Latvian NLP tasks (Znotins and Barzdins, 2020; Ulčar and Robnik-Šikonja, 2021). In previous work, we introduced a family of transformer encoders trained on a 6.4B-token Latvian corpus (Znotins, 2026), covering modern encoder architectures such as RoBERTa (Liu et al., 2019), DeBERTaV3 (He et al., 2023), and ModernBERT (Warner et al., 2025), including long-context variants, and showed that these Latvian-specific models achieve state-of-the-art or competitive results across a broad range of Latvian benchmarks.

Joint modeling of morphology and syntax has long been shown to improve parsing for morphologically rich languages, as morphology and syntax are better learned jointly than in isolation (Bohnet and Nivre, 2012; Tsarfaty et al., 2013). Empirical results from joint morphosyntactic parsers show that shared modeling improves accuracy and generalization in languages with complex inflectional systems. Modern systems typically employ biaffine dependency parsers (Dozat and Manning, 2017) combined with contextual encoder representations. In addition, symbolic linguistic resources such as morphological analyzers can be used to constrain decoding or rescore predictions, improving consistency and handling rare forms more robustly (Keersmaekers and Mercelis, 2024; Inoue et al., 2022). Our work follows this direction by integrating a Latvian morphological analyzer into a joint neural morphosyntactic parser.

3. Methodology

3.1. Data

We use the Latvian Universal Dependencies Treebank included in the UD v2.17 release (Nivre et al., 2020). The current release (v2.17) contains approximately 19,000 manually annotated sentences and 330,000 tokens drawn from a variety of contemporary written sources.

For encoder model training, we collect a 1B-token text corpus (see Table 1) from the Latvian National Corpus Collection (LNCC) (Saulite et al., 2022). This corpus is used for experiments on modest-data pretraining, while still providing linguistic and topical diversity representative of modern Latvian usage.

3.2. Pretrained Transformer Encoder Models

We evaluate publicly available pretrained BERT-like encoder models that support Latvian and compare their effectiveness on Latvian morphosyntactic tasks. We further examine pretraining and adaptation under limited-data conditions using an approximately 1B-token Latvian text corpus from the Latvian National Corpus Collection (LNCC). In this setting, we continue pretraining selected multilingual encoders for 3 epochs with a context size of 512 and a batch size of 512. In addition, we train RoBERTa-style models from scratch on the same corpus for 10 epochs, using Byte-Pair Encoding (BPE) with a 32k vocabulary for subword segmentation and the same context size and batch size.

3.3. Joint Modeling of Morphosyntactic Parsing

All encoder models are evaluated within the same unified multi-task architecture. For each sentence, we obtain all hidden states of the transformer encoder, including the embedding output and all

Name	# Tokens	Genre / Type
Ziņas	400M	News articles (web, media)
Tīmeklis2020	300M	Web texts (general, web)
LVK2022 (Levāne-Petrova et al., 2023)	123M	Balanced corpus of modern Latvian
Likumi (Darģis, 2022a)	116M	Legal acts (specialized)
Vikipēdija (Darģis, 2022b)	28M	Wikipedia articles
Saeima (Auzina et al., 2018)	24M	Parliamentary debates (spoken-edited)
Disertācijas	23M	PhD theses (academic)
Emuāri	8M	Blog texts (web)

Table 1: Overview of selected Latvian text corpora from the LNCC

contextual layers. Subword representations are aligned to words by average pooling over the corresponding subword pieces, yielding word-level representations. Token-level predictions are produced using shallow feed-forward classification heads with layer normalization and dropout.

For each prediction task, we use a separate learned scalar mixture over the available layer representations. The model computes a weighted average of all encoder layers, allowing each task to exploit different levels of contextual abstraction. After predicting UPOS, we incorporate a learned embedding of the predicted tags into the representation space. This embedding is appended to the layer representations and made available to subsequent tasks (XPOS, FEATS, lemmatization, and dependency parsing) through their respective layer mixtures. The resulting task-specific representations are passed to shallow feed-forward classification heads.

We use a learning rate of 6×10^{-4} for the task-specific heads and a ten times smaller learning rate for the encoder during fine-tuning. Reported results are averaged over five random seeds. We use a batch size of 16 and train for 15 epochs.

Morphological Tagging. Universal POS (UPOS), language-specific POS (XPOS), and morphological features (FEATS) are predicted using separate classification heads applied to the shared encoder representations. Following recent work in Universal Dependencies, morphological features are modeled in a factored manner rather than as a single combined tag, mitigating sparsity arising from the large number of possible feature combinations. Conditioning XPOS predictions on UPOS helps maintain consistency between coarse-grained and fine-grained morphosyntactic categories, which is particularly important for Latvian's rich inflectional morphology.

Lemmatization. For lemmatization, we adopt the pattern-based approach proposed by Straka (2018), which expresses each *form* $\rightarrow$ *lemma* pair as a transformation rule. Instead of predicting lemmas directly, the model predicts edit operations that transform the word form into the corresponding lemma. Following this formulation, we use multiple prediction heads to model different types of transformations, including casing changes, prefix edits, suffix edits, and absolute replacements. This approach allows the model to generalize across recurring morphological patterns while keeping the prediction space compact.

Dependency Parsing. Dependency structures are predicted using a biaffine graph-based parser (Dozat and Manning, 2017). Each word is scored against all other words to determine its most probable syntactic head and the corresponding dependency relation. The biaffine scorer jointly models arc and relation scores. To ensure well-formed syntactic trees, we decode the structure using a maximum spanning tree algorithm under a single-root constraint, equivalent to the Chu–Liu/Edmonds algorithm (Chu, 1965; Edmonds, 1967).

Joint Optimization. All tasks are trained jointly using a multi-task objective. The overall loss combines objectives from different task groups, such as dependency parsing, monolithic tagging tasks, and factored tagging components. A simplified formulation is:

$$\mathcal{L} = \lambda_{\text{arc}} L_{\text{arc}} + \lambda_{\text{tag}} \sum_{t \in T} L_t + \lambda_{\text{fact}} \frac{1}{|F|} \sum_{f \in F} L_f$$

where L_{arc} denotes the dependency arc prediction loss, $\sum_{t \in T} L_t$ represents losses for monolithic tagging tasks (e.g., predicting a complete tag as a single label), and $\sum_{f \in F} L_f$ corresponds to losses for factored tag components predicted by separate classification heads. The coefficients λ control the relative contribution of each task group in the joint objective.

In the final configuration, each task is assigned weight 1. For factored tagging tasks, this weight

is distributed equally across all feature heads. For lemmatization, we use a higher weight $\lambda_{\text{lemma}} = 4$, which empirically improved performance as lemma rule prediction benefited from a higher effective learning rate.

Evaluation. We report standard UD metrics computed with the official CoNLL evaluation script. Morphosyntactic performance is measured by token-level accuracy for UPOS, XPOS, FEATS (Universal Morphological Features such as case, number, and gender), and lemmas. Dependency parsing quality is assessed using UAS (Unlabeled Attachment Score) and LAS (Labeled Attachment Score), which measure the accuracy of both the syntactic head and the relation label.

To provide a more granular analysis, we also report CLAS (Content Labeled Attachment Score), MLAS (Morphology-aware Labeled Attachment Score), and BLEX (Bi-Lexical Dependency Score), which respectively emphasize content-bearing words and the integration of morphological features or lemmas within the dependency structure. Results are summarized in Table 2.

3.4. Factored vs. Monolithic Morphological Tagging

Morphological annotation schemes often define tags as combinations of multiple features (e.g., part of speech, number, case, tense). This results in a large number of possible tag combinations, many of which occur rarely in training data. Such sparsity makes learning monolithic tag representations difficult, particularly for morphologically rich languages.

We compare two alternative formulations for morphological tag prediction. The first is a single-head model (*MonoTag*) that treats each full tag as a single class in a multiclass classification setting. This approach guarantees internal tag consistency but suffers from severe data sparsity and limited generalization to unseen feature combinations.

Alternatively, we consider a factored formulation (*MultiTag*), where individual morphological features are predicted independently using separate classification heads. This design substantially reduces the output space and enables parameter sharing across features. Although this formulation assumes conditional independence between feature predictions—a simplification that is not linguistically realistic—it has nonetheless demonstrated strong empirical performance for morphologically rich languages (Tkachenko and Sirts, 2018; Keersmaekers and Mercelis, 2024).

To ensure valid tag outputs, we apply feature compatibility constraints during decoding, masking morphologically incompatible feature combinations according to the tagset specification.

Formally, the monolithic (*MonoTag*) approach models each tag as a single label, whereas the factored (*MultiTag*) formulation treats tagging as a multi-label classification problem, where the log-probability of a complete tag is computed as the sum of the log-probabilities of its feature components:

$$\log p(t \mid h) = \sum_{f \in F} \log p(f \mid h)$$

where h is the contextual representation of the word and F denotes the set of morphological features defining the tag.

We apply this comparison to both language-specific XPOS tags and universal morphological feature annotations, evaluating whether factored prediction improves generalization and robustness under data sparsity.

3.5. Incorporating a Morphological Analyzer

To further improve morphological prediction accuracy, we integrate the Latvian morphological analyzer (Paikens et al., 2024) during decoding. Morphological analyzers provide lexicon-based analyses that help guide neural models toward linguistically valid tag combinations, especially for ambiguous or rare word forms (Keersmaekers and Mercelis, 2024; Inoue et al., 2022). When an analyzer analysis is selected, we also use its lemma instead of the lemma produced by the model's rule-based lemmatizer.

The decoding procedure is as follows. For each token, the analyzer first produces a set of candidate analyses for the observed word form. Each candidate includes a lemma and a full morphological interpretation. We map these analyses to the same representation used by the model, including the corresponding XPOS tag and universal morphological features.

Next, we score each analyzer candidate using the model's predicted probabilities over morphological features and select the highest-scoring analyzer-supported analysis. In parallel, the model also produces its own unconstrained prediction by selecting the most likely feature values without consulting the analyzer.

Because the analyzer may miss valid analyses, we do not apply it as a hard constraint. Instead, we use a soft backoff strategy: we compare the best analyzer candidate with the model's greedy prediction and accept the analyzer candidate only if its score is within a threshold γ of the model prediction. Otherwise, we keep the model prediction. The threshold γ is tuned on the development set.

Lemma replacement is applied only when the analyzer candidate is selected. If the model prediction is retained, we also keep the model's predicted

lemma.

To improve robustness, we exclude several analyzer attributes from this procedure: noun type, verb transitivity, and adjective type. These features are often lexical or context-dependent and are predicted more accurately by the contextual model. We also disable analyzer-side guessing for unseen forms, since guessed analyses may introduce noise.

This strategy allows us to benefit from analyzer constraints while preserving the model's ability to handle forms that are missing from the analyzer or not reliably analyzed by it.

4. Results

4.1. Encoder model evaluation

Table 2 summarizes results for all evaluated models. The best overall base model is *lv-deberta-base*, which reaches 94.94% XPOS and 93.19% LAS. It is followed closely by *lv-roberta-base*. Among publicly available multilingual encoders, *XLM-RoBERTa-large* is the strongest model, with 94.04% XPOS and 92.45% LAS. Overall, the results show that Latvian-specific pretraining provides a clear advantage over general multilingual encoders, especially for fine-grained morphosyntactic prediction.

To test whether limited Latvian pretraining data is already beneficial, we continued pretraining several multilingual encoders on the 1B-token LNCC corpus. In all cases, adaptation improves both XPOS and LAS. For example, *XLM-RoBERTa-base* improves from 93.36% to 94.20% XPOS and from 91.51% to 92.50% LAS. Similarly, *GTE-multilingual-MLM-base* improves from 93.46% to 94.17% XPOS and from 91.71% to 92.41% LAS. These gains show that even modest amounts of in-language continued pretraining substantially improve multilingual encoders for Latvian.

We also trained monolingual Latvian encoders from scratch on the same 1B-token corpus. Despite the much smaller pretraining data size compared with the 6.4B-token models, the resulting systems are already highly competitive. In particular, *lv-roberta-base-lnnc* reaches 94.40% XPOS and 92.79% LAS, approaching the performance of the much larger Latvian-specific models and outperforming general multilingual base encoders. This shows that strong Latvian morphosyntactic performance can be achieved even with relatively limited monolingual pretraining data, provided that the corpus remains reasonably large and diverse.

The results highlight the importance of pretraining data scale and diversity. While the best performance is achieved by monolingual models trained on large Latvian corpora (6.4B tokens), the 1B-token LNCC models already yield highly competitive results. This is further supported by the strong performance of *hplt-bert-base-lvs*, trained on large cleaned web data (Burchell et al., 2025). Overall, both continued pretraining of multilingual encoders and monolingual pretraining are viable approaches, trading off efficiency and performance depending on data availability.

Applying morphological-analyzer-constrained decoding to *lv-deberta-base* yields the best results overall, improving XPOS from 94.94% to 95.22% and lemma accuracy from 98.01% to 98.72%. The same trend holds across other models, with consistent improvements in XPOS and lemma accuracy. For *lv-roberta-base-lnnc*, XPOS increases from 94.40% to 94.77% and lemma accuracy from 97.78% to 98.52%. For *XLM-RoBERTa-large*, XPOS improves from 94.04% to 94.51% and lemma accuracy from 97.87% to 98.44%. The largest gains are observed for *XLM-RoBERTa-base*, where XPOS increases from 93.36% to 94.12% and lemma accuracy from 97.46% to 98.23%. Overall, improvements are consistent across models, with larger gains for weaker encoders.

The proposed models also outperform off-the-shelf NLP toolkits that support Latvian morphosyntactic parsing. Among these, *Trankit* with *XLM-Roberta-large* is the strongest baseline, reaching 91.22% XPOS and 91.25% LAS, but it remains below the best jointly trained models. *Stanza* performs substantially worse on both metrics. *LVTagger* remains a strong traditional baseline for Latvian morphological analysis, reaching 93.44% XPOS, but it does not match the best neural models. Overall, the results show that Latvian-specific pretraining, combined with analyzer-constrained decoding, sets a new state-of-the-art for Latvian morphosyntactic parsing.

4.2. Learning Curve

To further analyze data efficiency, we examine the learning behavior of the best-performing model, *lv-deberta-base*, under varying amounts of training data (Figure 1). We randomly sample subsets of the training corpus containing 500, 1k, 2k, 4k, 8k, and 16k (100%) sentences. We evaluate both a full joint setup that predicts all tasks and corresponding single-task setups. For XPOS and universal features, we compare monolithic classification against factored prediction, and additionally evaluate morphological-analyzer-constrained decoding.

Factored XPOS modeling yields clear and consistent improvements over monolithic (*Mono*) classification, especially in low-resource conditions (500–2k sentences), where sparsity effects are most pronounced. While the performance gap narrows as more training data becomes available, factored modeling remains the more robust and reliable approach across all data sizes.

Base Model	UPOS	XPOS	Feats	Lemmas	UAS	LAS	CLAS	MLAS	BLEX
General Multilingual Encoders									
xlm-roberta-large	98.73±0.09	94.04±0.23	96.96±0.11	97.87±0.16	94.65±0.17	92.45±0.20	91.04±0.17	86.66±0.16	88.71±0.33
xlm-roberta-base	98.49±0.08	93.36±0.07	96.37±0.04	97.46±0.06	93.87±0.25	91.51±0.12	89.95±0.13	84.98±0.10	87.24±0.14
mdeberta-v3-base	98.70±0.06	93.75±0.10	96.72±0.06	97.49±0.10	94.38±0.07	92.14±0.09	90.71±0.12	85.96±0.23	87.82±0.22
mmBERT-base	98.49±0.15	93.39±0.23	96.40±0.23	97.49±0.16	93.91±0.22	91.43±0.27	89.81±0.23	84.84±0.10	87.15±0.12
mmBERT-small	98.27±0.15	92.71±0.21	96.03±0.13	97.20±0.11	93.30±0.22	90.73±0.21	89.01±0.26	83.68±0.23	86.05±0.26
gte-mlm-base	98.54±0.02	93.46±0.21	96.51±0.19	97.50±0.06	94.06±0.07	91.71±0.13	90.16±0.12	85.34±0.22	87.48±0.16
cis-lmu/glot500-base	98.34±0.11	92.83±0.10	96.06±0.15	97.13±0.10	93.41±0.23	90.88±0.36	89.18±0.40	83.84±0.43	86.08±0.50
BSC-LT/mRoBERTa	98.26±0.06	92.68±0.17	95.93±0.05	96.99±0.08	93.44±0.20	90.97±0.16	89.33±0.20	83.89±0.27	86.09±0.18
bert-base-multi	98.14±0.10	92.08±0.28	95.39±0.20	96.86±0.03	91.87±0.16	89.27±0.19	87.37±0.21	81.60±0.41	84.34±0.11
Existing Latvian-Specific Models									
hplt-bert-base-lvs	98.90±0.02	94.49±0.03	97.47±0.02	97.84±0.03	94.95±0.02	92.76±0.03	91.30±0.05	87.48±0.06	88.78±0.07
litlat-bert	98.69±0.03	93.86±0.10	96.91±0.07	97.53±0.04	94.55±0.08	92.31±0.05	90.84±0.06	86.41±0.11	87.97±0.04
DGurgurov/xlm-r_lvs	98.64±0.03	93.77±0.09	96.73±0.04	97.72±0.12	94.09±0.06	91.81±0.08	90.25±0.12	85.70±0.13	87.86±0.35
lvbert	98.45±0.05	92.94±0.19	96.45±0.16	96.81±0.13	93.69±0.11	91.30±0.10	89.79±0.10	84.77±0.22	86.33±0.12
Monolingual Latvian Models (6.4B Tokens)									
lv-deberta-base	**99.01**±0.05	94.94±0.10	**97.66**±0.07	98.01±0.13	**95.30**±0.11	**93.19**±0.09	**91.81**±0.09	88.20±0.05	89.49±0.23
lv-roberta-base	98.92±0.02	94.61±0.11	97.41±0.04	98.02±0.05	95.14±0.15	93.10±0.18	91.72±0.18	87.74±0.33	89.40±0.21
lv-mbert-large	98.82±0.09	94.39±0.10	97.32±0.18	97.89±0.11	94.94±0.22	92.65±0.21	91.16±0.20	87.09±0.36	88.74±0.28
lv-mbert-base	98.81±0.04	94.16±0.09	97.23±0.06	97.72±0.08	94.67±0.20	92.40±0.22	90.91±0.29	86.76±0.20	88.34±0.21
lv-mbert-mini	98.66±0.02	93.63±0.08	96.94±0.03	97.13±0.04	94.20±0.03	91.87±0.02	90.36±0.05	85.80±0.06	87.05±0.04
Monolingual Latvian Models (1B LNCC; Pretrained & Adapted)									
lv-roberta-base-lnnc	98.78±0.07	94.40±0.06	97.29±0.10	97.78±0.01	94.93±0.03	92.79±0.03	91.41±0.04	87.31±0.11	88.87±0.06
lv-roberta-small-lnnc	98.40±0.09	92.49±0.13	96.47±0.05	96.42±0.05	93.59±0.27	91.15±0.24	89.51±0.23	84.54±0.25	85.32±0.07
xlm-roberta-base-lv	98.78±0.05	94.20±0.12	97.11±0.17	98.01±0.04	94.70±0.14	92.50±0.14	91.00±0.13	86.79±0.35	88.81±0.11
gte-mlm-base-lv	98.75±0.02	94.17±0.10	96.99±0.04	97.90±0.03	94.68±0.20	92.41±0.15	90.97±0.12	86.65±0.14	88.71±0.06
mmBERT-small-lv	98.59±0.09	93.75±0.06	96.77±0.07	97.74±0.11	94.09±0.29	91.76±0.36	90.24±0.33	85.59±0.27	87.77±0.27
Morphological analyzer–constrained decoding									
lv-deberta-base	**99.01**±0.05	**95.22**±0.09	**97.66**±0.07	**98.72**±0.02	**95.30**±0.11	**93.19**±0.09	**91.81**±0.09	**88.20**±0.05	**90.41**±0.11
lv-roberta-base-lnnc	98.78±0.07	94.77±0.01	97.29±0.10	98.52±0.04	94.93±0.03	92.79±0.03	91.41±0.04	87.31±0.11	89.82±0.04
xlm-roberta-large	98.73±0.09	94.51±0.21	96.96±0.11	98.44±0.10	94.65±0.17	92.45±0.20	91.04±0.17	86.66±0.16	89.42±0.24
xlm-roberta-base	98.49±0.08	94.12±0.07	96.37±0.04	98.23±0.02	93.87±0.25	91.51±0.12	89.95±0.13	84.98±0.10	88.16±0.11
Off-the-shelf NLP Toolkits supporting Latvian morphosyntactic parsing									
Trankit (xlm-r-large)	97.61	91.22	95.18	95.83	93.63	91.25	89.78	82.69	85.58
Trankit (xlm-r-base)	97.16	90.24	94.47	95.61	92.05	89.44	87.73	79.78	83.52
Stanza	96.70	89.72	94.73	96.12	88.91	85.77	83.37	76.71	80.17
LVTagger		93.44		98.41					

Table 2: Universal Dependencies results on the Latvian UD treebank (UD v2.17) under joint multi-task fine-tuning. The best overall result in each column is highlighted in bold, and the best result within each model group is underlined.

Adding analyzer-constrained decoding has a strong positive effect on XPOS and lemma accuracy. The gains are largest in low-resource settings, where the analyzer provides valuable constraints for ambiguous or unseen forms, but remain clearly beneficial even at larger data sizes.

Across all training sizes, the joint model performs close to the corresponding single-task setups. Joint training slightly improves XPOS prediction, while dependency parsing and lemmatization remain marginally below their single-task counterparts. This suggests mild task interference, but overall indicates that shared representations are effective and that joint training provides a good balance between efficiency and performance.

Overall, the learning-curve analysis shows that (1) factored classification is preferable to monolithic XPOS prediction, particularly under limited supervision, (2) analyzer-constrained decoding provides substantial gains for fine-grained morphology and lemma prediction, especially in low-resource settings, and (3) joint multi-task training offers a practical compromise, achieving performance close to specialized single-task models.

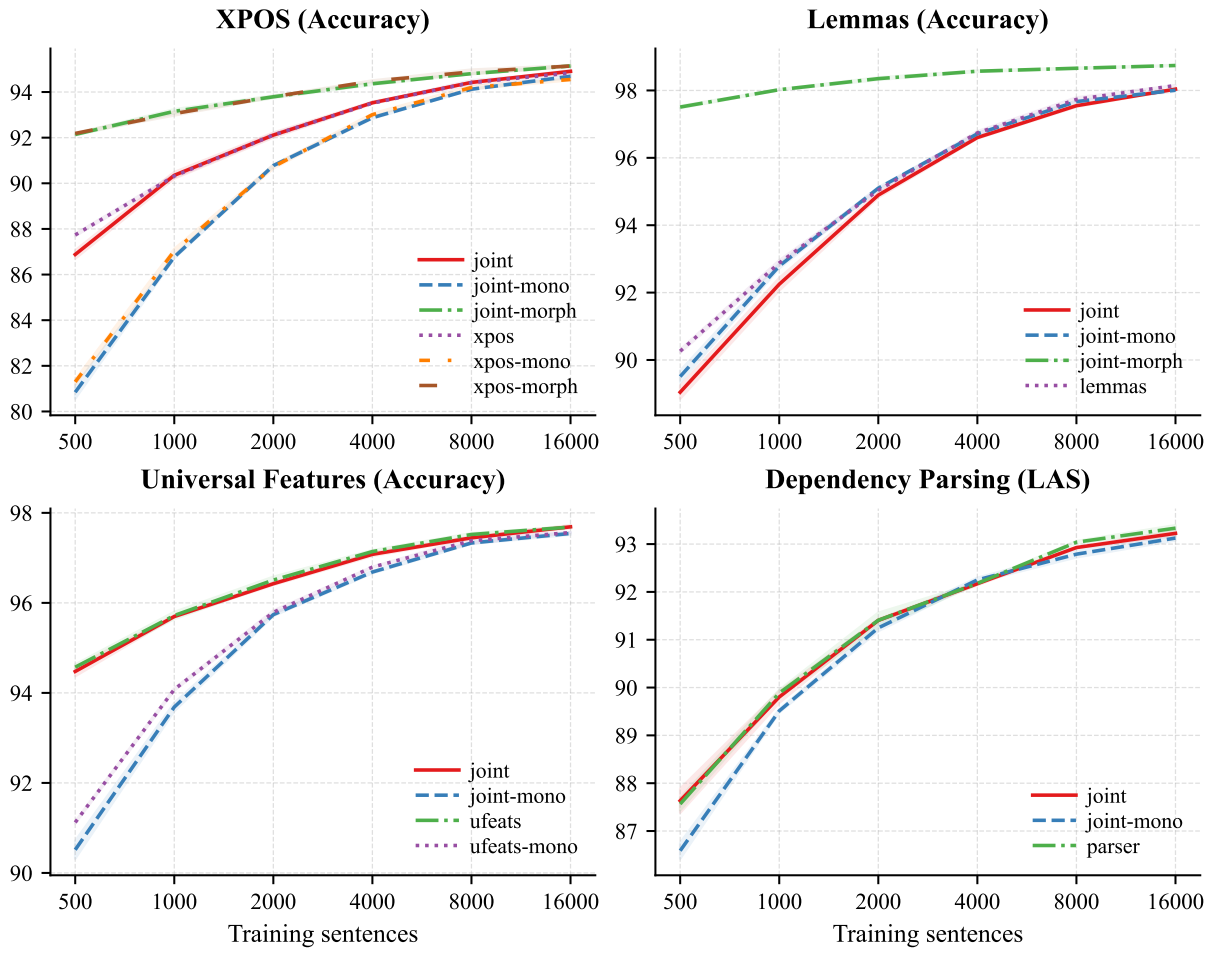

Figure 1: Learning curves across increasing training set sizes (500–16k sentences). *Joint* denotes the full multi-task setup predicting all tasks, while the other runs are single-task variants. *Mono* indicates monolithic XPOS and morphological feature classification (otherwise factored), and *Morph* indicates morphological-analyzer-constrained decoding.

Gold relation	Predicted relation	Count
obl	iobj	81
iobj	obl	79
obl	nmod	57
nmod	obl	57
advmod	discourse	37
advmod:emph	discourse	32
nmod	iobj	31
iobj	nmod	28
nsubj	root	20
root	nsubj	19
discourse	advmod	19
discourse	advmod:emph	18
advcl	acl	18
obl	dep	18
acl	advcl	17

Table 3: Most frequent dependency relation mismatches.

5. Error analysis

The most frequent dependency relation confusions (see Table 3) are between *obl* and *iobj*, followed by *obl* and *nmod*. This suggests that the model has particular difficulty distinguishing oblique dependents from indirect objects and nominal modifiers, especially in constructions where case marking and syntactic attachment allow more than one plausible interpretation. Further confusion between *nmod* and *iobj* supports the same observation, namely that argumenthood and adnominal modification are not always sharply separable on the basis of surface form alone.

The most frequent morphological confusions (see Table 4) involve verb transitivity (transitive ↔ intransitive), adjective type (relative ↔ qualificative), and gender (feminine ↔ masculine). These errors point to categories whose correct interpretation often depends on syntactic and semantic context rather than on word form alone. Transitivity mismatches

Attribute	Gold value	Predicted value	Count
Transitivity	Transitive	Intransitive	155
Adjective type	Relative	Qualificative	145
Transitivity	Intransitive	Transitive	127
Gender	Feminine	Masculine	126
Gender	Masculine	Feminine	102
Adjective type	Qualificative	Relative	101
Number (governed)	Plural	Singular	85
Number (governed)	Singular	Plural	58
Governed case	Dative	Accusative	47
Number	Singular	Plural	43
Part of speech	Particle	Adverb	40
Number	Plural	Singular	36
Case	Accusative	Genitive	33
Number	Singular	Singulare tantum	31
Noun type	Proper noun	Common noun	31

Table 4: Most frequent morphosyntactic attribute mismatches.

are the most common, suggesting that the model often struggles with verbs that can appear both with and without a direct object, or whose argument structure is difficult to recover from local context. Confusions in adjective type likewise reflect the fact that many adjectives can be interpreted either as classificatory/relational or as descriptive depending on usage. Gender errors remain highly frequent, especially in agreement contexts where the correct value must be inferred from a broader syntactic or discourse environment.

A substantial share of errors also involves governed number and governed case, especially plural $\rightarrow$ singular and singular $\rightarrow$ plural mismatches in governed number, as well as dative $\rightarrow$ accusative confusions in governed case. This indicates that the prediction of dependent features remains sensitive to head-dependent agreement and to argument structure within the clause. In addition, the presence of mistakes such as particle $\rightarrow$ adverb and proper noun $\rightarrow$ common noun shows that some errors arise at the boundary between morphosyntactic and lexical categorization, where distinctions are often subtle and context-dependent.

Additional errors involving *advmod*, *discourse*, and *advmod:emph* indicate persistent uncertainty in the analysis of clause-level particles, discourse markers, and emphasizing adverbials. These distinctions often depend on pragmatic and information-structural factors that are only weakly signaled in morphosyntax. Finally, recurring confusions between *advcl* and *acl*, as well as occasional *root* $\leftrightarrow$ *nsubj* mismatches, show that the model also encounters difficulties in determining the structural status of clausal dependents and, in some cases, the overall sentence backbone.

6. Conclusion and Future Work

This paper evaluated Latvian morphosyntactic parsing with pretrained encoders, demonstrating that language-specific adaptation provides a significant performance advantage over general multilingual models. Our results show that competitive performance can be achieved even under limited-data conditions, with models trained from scratch or adapted on 1B tokens of text outperforming much larger general-purpose multilingual encoders. The proposed unified architecture effectively handles the complexities of Latvian morphology by combining factored modeling with dependency parsing. The system offers a practical, single-model solution for joint tagging, lemmatization, and parsing, with optional morphological analyzer integration available to further improve precision for fine-grained attributes and rare forms.

Future work may explore more sophisticated methods for integrating symbolic linguistic resources directly into the model architecture beyond simple decoding constraints. Another direction is the development of more expressive lemma generation approaches that move beyond fixed edit-rule patterns to better handle irregular stems and complex word-formation processes. Finally, exploring model compression techniques, such as distillation or pruning, could significantly increase throughput for high-volume linguistic processing while maintaining the accuracy levels achieved by larger transformer encoders.

7. Acknowledgements

This work was funded by the European Union Recovery and Resilience Facility project “Language Technology Initiative” (2.3.1.1.i.0/1/22/I/CFLA/002).

8. Bibliographical References

Guntis Barzdins, Normunds Gruzitis, Gunta Nespore, and Baiba Saulite. 2007. Dependency-based hybrid model of syntactic analysis for the languages with a rather free word order. In *Proceedings of the 16th Nordic Conference of Computational Linguistics*, pages 13–20, Tartu, Estonia.

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. Association for Computational Linguistics.

Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, Farrokh Mehryary, Vladislav Mikhailov, Nikita Moghe, Amanda Myntti, Dayyán O'Brien, Stephan Oepen, Proyag Pal, Jousia Piha, Sampo Pyysalo, Gema Ramírez-Sánchez, David Samuel, Pavel Stepachev, Jörg Tiedemann, Dušan Variš, Tereza Vojtěchová, and Jaume Zaragoza-Bernabeu. 2025. An expanded massive multilingual dataset for high-performance language technologies (HPLT). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485, Vienna, Austria. Association for Computational Linguistics.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the International Conference on Learning Representations (ICLR) 2017*. Poster.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards – B. Mathematics and Mathematical Physics*, 71B(4):233–240.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Go Inoue, Salam Khalifa, and Nizar Habash. 2022. Morphosyntactic tagging with pre-trained language models for Arabic and its dialects. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.

Alek Keersmaekers and Wouter Mercelis. 2024. Adapting transformer models to morphological tagging of two highly inflectional languages: a case study on Ancient Greek and Latin. In *Proceedings of the 1st Workshop on Machine Learning for Ancient Languages (ML4AL 2024)*, pages 165–176, Hybrid in Bangkok, Thailand and online. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Gunta Nespore, Baiba Saulite, Guntis Barzdins, and Normunds Gruzitis. 2010. Comparison of the SemTi-Kamols and Tesniere's dependency

grammars. In *Human Language Technologies – The Baltic Perspective*, volume 219 of *Frontiers in Artificial Intelligence and Applications*, pages 233–240. IOS Press.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

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 *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Peteris Paikens. 2016. Deep Neural Learning Approaches for Latvian Morphological Tagging. In *Human Language Technologies - The Baltic Perspective*, volume 289. IOS Press.

Peteris Paikens, Lauma Pretkalniņa, and Laura Rituma. 2024. A computational model of latvian morphology. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 221–232.

Peteris Paikens, Laura Rituma, and Lauma Pretkalnina. 2013. Morphological analysis with limited resources: Latvian example. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 267–277, Oslo, Norway.

Lauma Pretkalnina, Gunta Nespore, Kristine Levane-Petrova, and Baiba Saulite. 2011. A Prague Markup Language profile for the SemTi-Kamols grammar model. In *Proceedings of the 18th Nordic Conference of Computational Linguistics*, pages 303–306, Riga, Latvia.

Lauma Pretkalnina, Laura Rituma, and Baiba Saulite. 2018. Deriving Enhanced Universal Dependencies from a Hybrid Dependency-Constituency Treebank. In *Text, Speech, and Dialogue*, volume 11107, pages 95–105. Springer.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Baiba Saulite, Roberts Dargis, and Normunds Gruzitis et al. 2022. Latvian National Corpora Collection – Korpuss.lv. In *Proceedings of the 13th Language Resources and Evaluation Conference (LREC)*.

Andrejs Spektors, Ilze Auzina, and Roberts Dargis et al. 2016. Tezaurs.lv: the largest open lexical database for Latvian. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Alexander Tkachenko and Kairit Sirts. 2018. Modeling composite labels for neural morphological tagging. *arXiv preprint arXiv:1810.08815*.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologically rich languages: Introduction to the special issue. *Computational Linguistics*, 39(1):15–22.

Matej Ulčar and Marko Robnik-Šikonja. 2021. Training dataset and dictionary sizes matter in bert models: the case of baltic languages. In *International conference on analysis of images, social networks and texts*, pages 162–172. Springer.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

A. Znotins and E. Cirule. 2018. Nlp-pipe: Latvian nlp tool pipeline. In *Human Language Technologies - The Baltic Perspective*, volume 307, pages 183–189. IOS Press.

Arturs Znotins. 2026. Pretraining and benchmarking modern encoders for latvian. In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages*. Association for Computational Linguistics.

Arturs Znotins and Guntis Barzdins. 2020. LVBERT: Transformer-Based Model for Latvian Language Understanding. In *Human Language Technologies - The Baltic Perspective*, volume 328, pages 111–115. IOS Press.

9. Language Resource References

Auzina, Ilze and Dargis, Roberts and Bojars, Uldis and Paikens, Peteris and Znotins, Arturs and Rabante-Busa, Guna. 2018. *Corpus of the Saeima (the Parliament of Latvia)*. IMCS at University of Latvia.

Roberts Dargis. 2022a. Corpus of legal acts of the republic of latvia (likumi). CLARIN-LV digital library at IMCS, University of Latvia.

Roberts Dargis. 2022b. Latvian wikipedia. CLARIN-LV digital library at IMCS, University of Latvia.

Levāne-Petrova, Kristīne and Dargis, Roberts and Pokratniece, Kristīne and Lasmanis, Viesturs Jūlijs. 2023. *Balanced Corpus of Modern Latvian (LVK2022)*. ISLRN http://hdl.handle.net/20.500.12574/84. CLARIN-LV digital library at IMCS, University of Latvia.

Zeman, Daniel and Nivre, Joakim and Abrams, Mitchell and Ackermann, Elia. 2021. *Universal Dependencies 2.17*. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, ISLRN http://hdl.handle.net/11234/1-4611.