Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2026”

June 24–26, 2026

Extending Retag to Detect Conversion Errors in Syntactic Annotation: SynTagRus as a Case Study

Daniil Timchenko
MIPT
Moscow, Russia
danitim4@gmail.com

Andrei Movsesian
IITP RAS
Moscow, Russia
andrei.movsesian@gmail.com

Abstract

The Universal Dependencies (UD) project has grown rapidly through semi-automatic conversion of existing treebanks, but ensuring the quality of converted annotations remains a challenge. Manual verification does not scale, and existing automatic methods cannot distinguish conversion errors from inherent annotation complexity. We present a method that addresses this gap by training two parsers on a token-aligned parallel corpus: one on the original annotation and one on its UD conversion. By requiring correct predictions from the parser trained on the original annotation, our approach isolates errors specifically introduced during conversion while filtering out cases where the construction is simply difficult to parse. We demonstrate the effectiveness of this method on the SynTagRus corpus and its UD counterpart. To enable a direct comparison, we created a fully token-aligned version of the two corpora, resolving differences in tokenization and ellipsis representation. We also proposed a simple method for aligning syntactic relations across the two corpora, addressing the fact that relations involving the same token do not always correspond due to differences in annotation schemes. Our analysis identified several hundred errors in the test set. These comprise six distinct types of conversion errors, three of which persist in the current converter, and ten groups of annotation inconsistencies between the old and new corpus parts. Our method offers a practical, scalable tool for conversion error detection and is applicable to any language pair with aligned original and converted annotations.

Keywords: conversion error detection, annotation error detection, Retag, SynTagRus, syntax, Universal Dependencies

DOI: 10.29003/2075-7182-2026-24-644-655

Расширение метода Retag для выявления ошибок конвертации в синтаксической разметке на примере корпуса СинТагРус

Даниил Тимченко
МФТИ
Москва, Россия
danitim4@gmail.com

Андрей Мовсесян
ИППИ РАН
Москва, Россия
andrei.movsesian@gmail.com

Аннотация

Проект Универсальных зависимостей (UD) активно развивается, в том числе благодаря полуавтоматической конвертации существующих корпусов с синтаксической разметкой, что требует анализа качества конвертации. Ручная проверка не масштабируется, а существующие автоматические методы не позволяют отделить ошибки конвертации от ошибок, вносимых самим методом. Мы предлагаем подход, решающий эту проблему. А именно, мы обучаем две модели автоматической синтаксической разметки на двух корпусах, выравненных на уровне токенов: одну на исходном корпусе, другую на сконвертированном. Анализируя только те случаи, где модель, обученная на исходном корпусе, дает верное предсказание, наш подход выделяет ошибки, обусловленные именно конвертацией, отсеивая при этом случаи, сложные для анализа моделью. Эффективность метода демонстрируется на корпусе СинТагРус. Для прямого сравнения мы создали полностью выравненную на уровне токенов версию двух корпусов, устранив расхождения в токенизации и представлении эллипсиса. Мы также предложили простой метод согласования синтаксических отношений между двумя корпусами, поскольку отношения, приписанные одному и тому же токену, не всегда соответствуют друг другу из-за различий в схемах аннотации. Наш подход выявил несколько сотен ошибок в тестовой выборке. Среди них — шесть различных

типов ошибок конвертации (три из которых сохраняются в текущей версии конвертера) и десять групп рассогласований в разметке между старой и новой частями корпуса. Предложенный метод представляет собой эффективный и масштабируемый инструмент для выявления ошибок конвертации и может быть применен к любому корпусу при наличии выравненных оригинальной и сконвертированной разметок.

Ключевые слова: обнаружение ошибок конвертации, обнаружение ошибок разметки, Retag, СинТагРус, синтаксис, Универсальные зависимости

1 Introduction

The Universal Dependencies (UD) project (de Marneffe et al., 2021) is a valuable resource for multilingual research in computational linguistics, providing over 200 treebanks in more than 150 languages under a unified annotation scheme. The project continues to grow: each new UD release adds both new languages and new corpora for already represented languages. Many of these treebanks are not annotated natively in UD but converted from pre-existing annotation schemes (Sung and Shin, 2025; Cecchini et al., 2020; Branco et al., 2022).

However, the conversion process is inherently error-prone, so ensuring annotation quality in converted treebanks is therefore an important practical problem. Manual verification does not scale to corpora containing tens of thousands of sentences, and standard quality assessment methods such as inter-annotator agreement are not applicable, since the converted annotation typically exists as a single version. Automatic error detection methods designed for finding internal inconsistencies within a single corpus exist, but they often require manual filtering of errors since not all of them are genuine conversion errors.

In this paper, we propose a method that addresses this gap by exploiting the availability of both the original corpus and its converted version. We train two parsers on a token-aligned parallel corpus and use the disagreement between two parser on a held-out test set to identify errors attributable to the conversion. Conditioning the error analysis on the original annotation allows us to filter out constructions that are difficult for parsing in general and isolate errors introduced specifically by the conversion procedure.

We apply this method to the SynTagRus/SynTagRus-UD pair, where SynTagRus-UD was derived by automatic conversion from the SynTagRus dependency corpus. To achieve direct comparison, we construct a token-aligned version of the two corpora, including restored elliptical constructions. Since the corpus was converted in two stages by different converter versions, we stratify the error analysis by corpus part, which allows us to distinguish persistent conversion errors from annotation inconsistencies introduced by the change of converter. The identified errors are analyzed linguistically and accompanied by practical recommendations for improving the corpus.

2 Related Work

Many Universal Dependencies treebanks are created by automatic conversion from existing annotation schemes, using methods ranging from tree transducers (Borges Völker et al., 2019) to rule-based pipelines (Arnardóttir et al., 2023) and multi-layer conversion (Peng and Zeldes, 2018). The conversion of SynTagRus to UD is described in (Droganova and Zeman, 2016), which details the structural transformations and label mappings applied during the conversion process. Cross-treebank parsing experiments to assess consistency among the four Russian UD treebanks were later conducted in (Droganova et al., 2018). That work focuses on comparing different treebanks of the same language, whereas we compare the original and converted versions of the same corpus, which allows us to trace errors directly to specific conversion rules.

A number of methods have been proposed for detecting annotation errors within a single corpus. The variation nuclei approach (Boyd et al., 2008) flags cases where the same pair of words appears in identical local contexts but receives different dependency labels, suggesting that one annotation is erroneous. This method was later extended for UD corpora and shown to be effective for English, French, and Finnish (de Marneffe et al., 2017). A complementary approach (Alzetta et al., 2017) detects systematic errors by measuring the reliability of individual dependency arcs, focusing on error patterns that a parser would replicate during training. A broad survey and reimplementation of 18 error detection methods across different annotation tasks is provided in (Klie et al., 2023); our method can be seen as an extension of

the Retag approach described there, which uses a model trained on the same data to flag tokens where the prediction disagrees with the gold label.

Several studies use trained models as an indirect measure of annotation quality. Cross-treebank parsing accuracy has been interpreted as an indicator of annotation consistency (Droganova et al., 2018). An alternative approach is to regenerate sentences from their UD annotations and flag cases where the output differs from the original text (Lapalme, 2021). More recently, large language models have been used to compare two versions of a Turkish UD treebank by prompting the model to recover surface forms from annotations (Akkurt et al., 2024).

A common limitation of all the approaches mentioned is that they do not distinguish between genuine errors and model-related errors. These methods assess annotation quality globally through aggregate metrics or sentence-level comparison, but do not identify which specific annotation decisions are responsible for the errors. This results in labor-intensive process of manually filtering these errors to find the relevant ones. Our method addresses this limitation by conditioning error detection on the annotation of the original corpus.

3 Data Preprocessing

This study uses two morphologically and syntactically annotated corpora of Russian to demonstrate the proposed method: SynTagRus and its version within the Universal Dependencies project.

3.1 SynTagRus and SynTagRus-UD

SynTagRus[1] is a deeply annotated corpus of Russian developed by the Computational Linguistics Laboratory of IITP RAS (Boguslavsky et al., 2024). The corpus is an autonomous component of the Russian National Corpus, which provides a corpus search interface.[2] Its source texts include contemporary fiction, popular science articles, newspaper and journal publications, as well as online news texts. The corpus features multi-level annotation, but the present study uses only the morphological and syntactic layers.

SynTagRus-UD[3] is a version of SynTagRus within the UD project, available since the v1.3 release (Droganova and Zeman, 2016). The conversion from SynTagRus to UD involved changes at both the structural and labeling levels. At the structural level, prepositions, conjunctions, and auxiliary verbs (which serve as syntactic heads in SynTagRus) were demoted to dependent positions following the UD principle that relations hold primarily between content words. At the labeling level, the 68 SynTagRus dependency relation types were mapped onto 37 universal UD types and 11 subtypes; only 16 universal types have a direct equivalent, while the rest require morphosyntactic context for disambiguation. Punctuation, which is not tokenized in the original SynTagRus, was added as independent nodes in the dependency tree. Elliptical constructions, explicitly represented in SynTagRus, were transferred to the enhanced dependency structure and not present in the basic structure.

3.2 Corpus Alignment

The alignment of the two corpora follows the procedure described by other authors (Timchenko and Movsesian, 2024), where SynTagRus was converted to the CoNLL-U format, punctuation and multiword expressions were unified, and sentence pairs were matched by token identity. To align sentences with ellipsis, the authors used enhanced dependency structure of SynTagRus-UD. That procedure aligned most of the sentences; here we describe the additional steps that brought coverage to 100%.

The sentences that were not aligned in the mentioned work fall into two groups. The first involves negative pronominal constructions (*некуда, нечего, незачем*, etc.), which are single tokens in SynTagRus-UD but split into *не* and a pronominal component in SynTagRus. The second group consists of sentences where the number of elliptical nodes differs between the two corpora. For these, redundant elliptical tokens in SynTagRus were removed, and the affected dependency arcs were reattached to preserve a valid tree, with priority given to nodes that have a corresponding counterpart in the UD corpus.

[1] https://ruscorpora.ru/page/corpora-datasets

[2] https://ruscorpora.ru/en/search/syntax

[3] https://github.com/UniversalDependencies/UD_Russian-SynTagRus

All modifications were applied exclusively to SynTagRus; the UD corpus is preserved in its original form with the exception of preprocessing extended dependency structures. About 200 sentences required manual correction. The result is a fully aligned parallel corpus: every sentence in SynTagRus-UD has a counterpart in SynTagRus with identical tokenization. Table 1 presents the corpus statistics.

Parameter	Total corpus	New part (2015–2020)	Old part
Number of sentences	87,337	25,447	61,890
Number of tokens	1,517,783	410,101	1,107,682
Number of elided tokens	2,197	757	1,440

Table 1: Characteristics of the parallel corpus (SynTagRus/SynTagRus-UD).

3.3 Data Division

The data is split into training, validation, and test sets following the original SynTagRus-UD partition. It is known that the old and new parts of the corpus have different annotation due to the use of different converter versions (Timchenko and Movsesian, 2024). To account for this, we additionally stratify the data by the period of conversion, yielding six parallel datasets: the full corpus, the old part, and the new part, each for both SynTagRus and SynTagRus-UD.

4 Method

The Retag method (van Halteren, 2000) detects annotation errors by training a model on the corpus of interest and flagging tokens where its predictions disagree with the gold annotation. We extend it to detect conversion errors by training two parsers on SynTagRus and SynTagRus-UD and comparing inconsistencies between parsers' predictions and gold annotation. If a parser trained on SynTagRus consistently handles a construction correctly, while a parser trained on SynTagRus-UD fails, the failure is likely related to the conversion process. The comparison requires that every dependency arc in one corpus can be unambiguously mapped to an arc in the other; this is achieved through token-level alignment (Section 3.2) and arc direction alignment (Section 4.2).

Klie et al. (2023) report that, within the range of reasonably performing models, varying the underlying model primarily shifts the precision/recall trade-off of errors detected by Retag rather than the types of patterns surfaced. We therefore expect a stronger model to mainly affect the noise level of the found conversion errors, leaving the relative distribution of errors across linguistic categories largely unchanged.

Nevertheless, we use UDPipe 2 (Straka, 2018) as the underlying model that has one of the strongest published parsing performance for SynTagRus-UD. UDPipe 2 is a multi-task model that utilizes pre-trained contextual embeddings from a multilingual BERT model and jointly predicts morphological features and dependency relations. The model operates on pre-tokenized input: tokens are taken directly from the gold annotation, and the model predicts only morphological features and dependency relations. As a result, predicted dependency trees are always defined over the same token set as the gold annotation. All hyperparameters are kept at their default values.

We train models in two configurations: syntax-only (predicting HEAD and DEPREL) and joint morphosyntax (additionally predicting UPOS and FEATS). Each configuration is trained on six datasets described in Section 3.3, yielding 12 models in total. We report Unlabeled Attachment Score (UAS) and Labeled Attachment Score (LAS) as the primary evaluation metrics, additionally computing them separately for elided tokens (UAS_{ell} and LAS_{ell}).

4.1 Handling Ellipsis

Elided tokens require special treatment because they lack surface forms. We explore two strategies for representing them as input to the model. In the first, each elided token is replaced with the BERT mask token (`[MASK]`), allowing the model to infer its syntactic role from context. In the second, word forms are restored based on the lemma information available in SynTagRus; in rare cases where the elided word is unknown, the mask token is retained.

4.2 Aligning Syntactic Relations

An additional step is required to ensure that the comparison of dependency labels is meaningful. SynTagRus and UD use different structural conventions: in SynTagRus, prepositions, conjunctions, and auxiliary verbs serve as syntactic heads, whereas in UD they are dependents. As a result, some dependency pairs that require comparison have different tokens as their dependents in the corresponding corpora.

To address this, we apply a preliminary transformation to the SynTagRus annotation. Whenever token i is the head of token j in one corpus while token j is the head of token i in the other, we reverse the arc direction for these two tokens in the SynTagRus annotation to match the UD convention. After this transformation, the arc directions are consistent across the two corpora, and the dependency labels can be compared at token level.

5 Results

5.1 Parsing Performance

Table 2 compares the two ellipsis handling strategies described in Section 4.1 on syntax-only models. Restoring word forms substantially improves parsing performance for elliptical constructions across all datasets while leaving overall UAS and LAS virtually unchanged since the number of sentences with ellipsis is relatively small. The mask token strategy yields comparable overall results and has the advantage of being applicable to any corpus, since it does not require word forms synthesis. We adopt word form restoration for the remaining experiments.

	Mask token				Restored word forms			
Dataset	**UAS**	**LAS**	**UAS_{ell}**	**LAS_{ell}**	**UAS**	**LAS**	**UAS_{ell}**	**LAS_{ell}**
SynTagRus-UD	94.58	92.73	78.68	70.96	94.57	92.74	87.13	82.72
SynTagRus-UD-Old	95.15	93.82	72.82	65.64	95.07	93.75	85.13	81.54
SynTagRus-UD-New	94.03	92.37	83.12	75.32	94.18	92.54	85.71	81.82
SynTagRus	94.83	91.23	84.56	76.84	94.76	91.19	92.65	85.29
SynTagRus-Old	95.19	91.36	80.51	72.31	95.24	91.45	90.77	84.62
SynTagRus-New	94.11	90.00	90.91	80.52	94.15	90.13	94.81	85.71

Table 2: Comparison of ellipsis handling strategies (syntax-only models).

Table 3 presents the evaluation results for the syntax-only and joint morphosyntax models across the six datasets.

Dataset	**Regime**	**UAS**	**LAS**	**UAS_{ell}**	**LAS_{ell}**
SynTagRus-UD	Syntax	94.57	92.74	87.13	82.72
	Joint	94.62	92.85	88.24	82.72
SynTagRus-UD-Old	Syntax	95.07	93.75	85.13	81.54
	Joint	95.12	93.89	85.13	81.03
SynTagRus-UD-New	Syntax	94.18	92.54	85.71	81.82
	Joint	94.08	92.58	92.21	87.01
SynTagRus	Syntax	94.76	91.19	92.65	85.29
	Joint	94.74	91.06	93.01	87.50
SynTagRus-Old	Syntax	95.24	91.45	90.77	84.62
	Joint	95.24	91.39	92.31	87.69
SynTagRus-New	Syntax	94.15	90.13	94.81	85.71
	Joint	94.17	90.15	96.10	87.01

Table 3: Evaluation results for syntax-only and joint morphosyntax models with restored ellipsis word forms. UAS_{ell} and LAS_{ell} are computed on elided tokens only.

Joint training provides marginal gains over syntax-only models, confirming that the addition of morphological tagging does not substantially improve syntactic quality in a multi-task setting. This is consistent with the findings of previous studies (Zhou et al., 2020).

Models trained on the old part outperform those trained on the full corpus. The differences are consistent, confirming the annotation heterogeneity between the two periods and supporting the decision to analyze conversion quality separately for the old and new parts.

Based on these observations, the error analysis in the following sections is conducted on the syntax-only models.

5.2 Syntactic Analysis

All tables in this section are filtered by the same thresholds: we retain only error patterns where at least one of the three corpus versions (full, old, new) contains at least 10 errors and where the error rate exceeds 1% in at least one version.

The analysis proceeds in three stages. First, we group the errors by (SynTagRus gold relation, UD gold relation, UD predicted relation) triples and sort them by the absolute number of errors in the *new* part of the corpus. Since the new part was processed by the current converter, errors found here may represent active conversion issues. Second, we exclude the cases already examined and sort the remainder by the number of errors in the *old* part, which may reveal issues specific to the earlier converter version. Third, we sort the remaining cases by the number of errors on the *full* corpus.

5.2.1 Errors in the New Part

Table 4 presents the error triples sorted by the number of errors in the new part of the corpus. Three of the five groups (предик, опред, атриб) represent conversion errors; the remaining two (аппоз, разъяснит) are model errors.

$\text{deprel}_{\text{SynTagRus}}$	$\text{deprel}_{\text{SynTagRus-UD}}$	$\text{pred}_{\text{SynTagRus-UD}}$	full	%	old	%	new	%	total
предик	nsubj:pass	nsubj	99	8.8	78	8.4	30	15.1	207
опред	acl	amod	48	4.5	29	3.6	17	6.4	94
аппоз	appos	flat:name	36	3.2	15	1.6	15	9.1	66
разъяснит	appos	parataxis	17	77.3	0	0.0	14	66.7	31
атриб	nmod	obl	71	2.5	56	2.5	11	1.9	138

Table 4: Error triples sorted by the number of errors in the new part. These represent potential conversion errors in the current converter.

5.2.2 Errors in the Old Part

Table 5 presents the error triples sorted by the number of errors in the old part of the corpus, after excluding cases already discussed in Table 4. Four of the seven groups (1-компл→obl/obj, примыкат, root) represent conversion errors; the remaining four are model errors.

$\text{deprel}_{\text{SynTagRus}}$	$\text{deprel}_{\text{SynTagRus-UD}}$	$\text{pred}_{\text{SynTagRus-UD}}$	full	%	old	%	new	%	total
1-компл	obl	obj	109	3.3	55	2.1	4	0.6	168
обст	obl	nmod	58	1.2	40	1.0	7	0.7	105
примыкат	appos	parataxis	44	80.0	37	71.2	0	0.0	81
1-компл	obj	obl	149	2.9	30	0.8	6	0.4	185
сравнит	obl	advcl	17	7.5	21	12.7	0	0.0	38
аппоз	appos	nmod	18	1.6	12	1.2	4	2.4	34
root	cop	root	18	36.0	11	52.4	1	3.5	30

Table 5: Error triples sorted by the number of errors in the old part (after excluding cases from Table 4).

5.2.3 Errors in the Full Corpus

Table 6 presents the 10 most frequent error triples sorted by the number of errors on the full corpus, after excluding cases from Tables 4 and 5. These cases share a characteristic pattern: high error counts on the full corpus but low counts on both the old and new parts individually, indicating annotation inconsistency between the two converter versions.

6 Discussion

The error filtering procedure identified 22 error triples that meet our thresholds across Tables 4, 5, and 6. Of these, six represent model errors: cases where the gold UD annotation is correct but the model fails to predict it. While not conversion errors, these cases are informative: they highlight constructions where

$\text{deprel}_{\text{SynTagRus}}$	$\text{deprel}_{\text{SynTagRus-UD}}$	$\text{pred}_{\text{SynTagRus-UD}}$	full	%	old	%	new	%	total
количест	nummod:gov	nummod	77	24.5	7	4.1	2	1.4	86
сравн-союзн	case	mark	61	93.9	3	60.0	0	0.0	64
присвяз	xcomp	obl	54	76.1	5	50.0	0	0.0	59
1-компл	xcomp	obl	48	3.5	6	0.6	0	0.0	54
обст	obl:tmod	obl	45	80.4	4	100.0	1	1.9	50
2-компл	iobj	obl	43	6.4	6	1.4	0	0.0	49
1-компл	iobj	obl	41	8.4	4	1.3	0	0.0	45
агент	obl:agent	obl	31	81.6	1	100.0	0	0.0	32
аппоз	flat:name	appos	29	4.9	8	1.7	6	4.5	43
2-компл	xcomp	obl	27	18.0	2	2.2	0	0.0	29

Table 6: Error triples sorted by the number of errors on the full corpus (after excluding cases from Tables 4 and 5).

the UD scheme introduces distinctions that are difficult for the parser to learn. The remaining 16 triples divide into conversion errors and annotation inconsistencies, which we analyze below.

6.1 Conversion Errors

Table 7 summarizes the six conversion errors identified across Tables 4 and 5. They fall into two groups: errors present in both parts of the corpus, which persist in the current converter, and errors concentrated in the old part, which have been largely resolved.

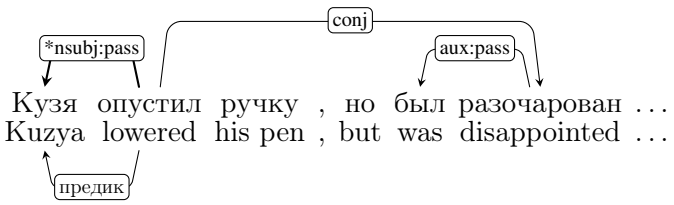
Error	Correct relation	Scope	Total
предик → nsubj:pass	nsubj	both parts	207
опред → acl	amod	both parts	94
атриб → nmod	obl	both parts	138
1-компл → obl	obj	mainly old	168
примыкат → appos	parataxis	old only	81
root → cop	root	old only	30

Table 7: Summary of conversion errors. "Scope" indicates whether the error is present in both corpus parts or predominantly in the old part. "Total" is the number of errors across all three corpus versions.

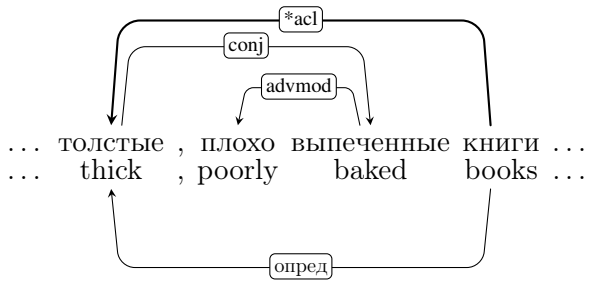
6.1.1 Errors Present in Both Parts

предик → nsubj:pass instead of nsubj. This error arises in sentences with two coordinated predicates, one in the active voice and one in the passive voice. Example 1 illustrates this. The gold annotation assigns `nsubj:pass` to the subject, apparently because the converter detects the presence of a passive construction somewhere in the sentence and propagates the `:pass` suffix to the subject of the active predicate.

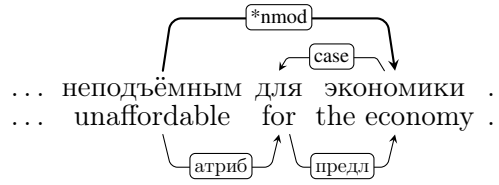
(1)



опред → acl instead of amod. In UD, `amod` marks simple adjectival modifiers, while `acl` is reserved for clausal modifiers of nouns. The converter apparently distinguishes the two by checking whether the modifier has any dependents: if it does, the relation is assigned `acl`. However, dependents such as `conj` (in coordinated adjective sequences) or `parataxis` (in parenthetical insertions) do not make the modifier clausal in the UD sense. Example 2 illustrates this case.

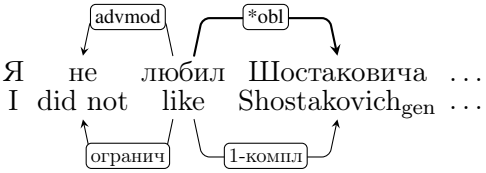
(2)


атриб → nmod instead of obl. In UD, `nmod` is used for nominal modifiers of nouns, while `obl` marks oblique dependents of predicates. In many of the flagged cases, the dependent is a prepositional phrase that modifies a predicative head (an adjective, verb, or pronoun) and therefore should be `obl` rather than `nmod`. Example 3 illustrates this.

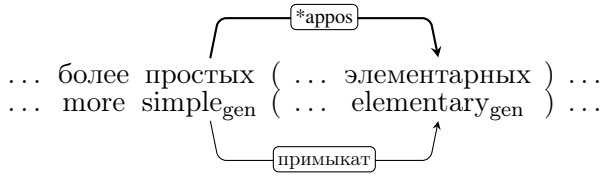
(3)


6.2 Errors Concentrated in the Old Part

1-компл → confusion between obl and obj The converter assigns `obj` only when the dependent bears accusative case; all other complements of the SynTagRus relation 1-компл receive `obl`. This misclassifies arguments of verbs that license accusative objects but appear in a different morphological form in a given context — most commonly, genitive under negation (Example 4). According to the UD guidelines, the determining factor should be whether the verb is capable of licensing an accusative object, not the morphological case in a particular sentence.

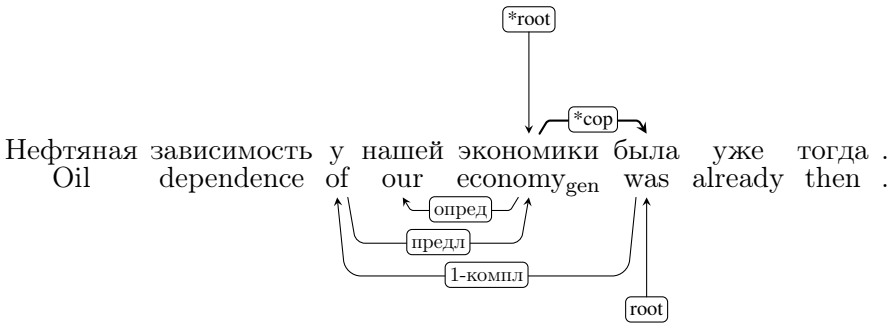
(4)


примыкат → appos instead of parataxis. In UD, `appos` is intended primarily for the relation between two nominals in apposition. The converter maps the SynTagRus relation примыкат to `appos` whenever the morphological features of the dependent and its head fully coincide, without checking whether both elements are nominal. As a result, `appos` is assigned too broadly, including cases where the dependent is a parenthetical elaboration involving adjectives, participles, or clausal elements, where `parataxis` would be more appropriate (Example 5).

(5)


root → cop instead of root. The converter treats all constructions with forms of *быть* 'to be' as copular, demoting *быть* to `cop` and promoting the nearest nominal or prepositional dependent to root. While this is correct for true copular sentences, it misanalyzes existential and locative constructions where *быть* is the main predicate of the clause (Example 6).

(6)



6.3 Annotation Inconsistencies

Table 8 summarizes the annotation inconsistencies identified in Table 6. These arise because the old and new parts of the corpus were processed by different converter versions. In every case, the new converter assigns the correct label, while the old converter produces an incorrect one.

SynTagRus relation	Old → New label	Errors (full)
количест	nummod → nummod:gov	77
сравн-союзн	mark → case	61
присвяз, 1-компл, 2-компл	obl → xcomp	129
обст	obl → obl:tmod	45
агент	obl → obl:agent	31
1-компл, 2-компл	obl → iobj	84
аппоз	appos → flat:name	29

Table 8: Summary of annotation inconsistencies between the old and new corpus parts. The "Old → New label" column shows the incorrect label assigned by the old converter and the correct label used by the new converter. "Errors (full)" is the number of errors on the full corpus attributable to the inconsistency.

The inconsistencies fall into several groups. The most frequent case involves количест → nummod:gov vs nummod: the subtype `:gov` marks constructions where the numeral governs the case of the noun, but in the old part of the corpus it is not consistently applied. A similar pattern holds for сравн-союзн → case vs mark, where subordinating conjunctions such as *чем* 'than' and *как* 'as' are correctly labeled `case` in the new part but `mark` in the old part. The relations присвяз, 1-компл, and 2-компл, all mapped to xcomp vs obl, reflect the same underlying issue: predicative and clausal complements that should receive `xcomp` are labeled `obl` in the old part. Two further groups, обст → obl:tmod vs obl and агент → obl:agent vs obl, involve missing subtypes in the old annotation. The 1-компл/iobj and 2-компл/iobj groups mirror the obj/obl issue discussed in Section 5.2.2: non-accusative core arguments receive `obl` in the old part but `iobj` in the new part. Finally, аппоз → flat:name vs appos reflects incomplete application of the `flat:name` relation in the old part, where multi-word proper names are left as `appos`.

6.4 Comparison with Annotation Error Detection

Here we illustrate the effect of using two parallel corpora. Table 9 shows the top 10 error pairs (UD gold relation, UD predicted relation) obtained by evaluating only the SynTagRus-UD-trained model, without filtering by the SynTagRus-trained model. This is the baseline that would be available without the original corpus.

These errors reveal two problems with this approach. First, while it successfully identifies conversion errors, they are difficult to analyze because the source of each error is unknown. For example, many of the 138 `nmod` → `obl` errors in Table 4 are genuine conversion errors which can be traced to the specific section of the conversion algorithm. At the same time, Table 9 contains more than 700 such errors. Although determining which of these are actual conversion errors requires additional research, they cannot be traced to the same section of the conversion algorithm, since they correspond to different syntactic relations in SynTagRus rather than solely the attributive one.

Second, many of the errors from Table 9 are model-related and not attributable to conversion process in any way. This includes confusion between adverbial modifier and subordination marker (`mark` →

Full				Old				New			
gold	**pred**	**n**	**%**	**gold**	**pred**	**n**	**%**	**gold**	**pred**	**n**	**%**
nmod	obl	393	3.2	nmod	obl	306	3.1	nmod	obl	89	4.0
obl	nmod	367	2.9	obl	nmod	275	2.8	obl	nmod	80	3.0
obj	obl	159	3.1	nsubj:pass	nsubj	78	8.4	nsubj:pass	nsubj	30	15.1
iobj	obl	158	10.9	appos	parataxis	75	6.3	root	conj	25	1.1
xcomp	obl	144	8.9	conj	parataxis	59	1.1	parataxis	conj	23	2.5
appos	parataxis	114	8.0	parataxis	conj	56	2.4	parataxis	root	22	2.4
nsubj:pass	nsubj	100	8.9	obj	nsubj	52	1.4	iobj	obl:agent	20	3.7
nummod:gov	nummod	83	21.5	parataxis	root	41	1.8	appos	parataxis	19	8.0
parataxis	conj	76	2.3	appos	nmod	40	3.3	obj	nsubj	19	1.3
obl:tmod	obl	68	81.9	mark	advmod	39	2.0	acl	amod	18	3.9

Table 9: 10 most frequent error pairs (UD gold, UD predicted) without filtering by the SynTagRus model.

`advmod`), confusion between subject and object for nouns in accusative case (`obj` → `nsubj`), and other errors which are standard for modern parsers.

7 Conclusion

We presented a method for detecting conversion errors in treebanks by training two parsers on a token-aligned parallel corpus: one on the original annotation and one on its conversion. By filtering for cases where only the conversion-trained parser fails, the method isolates errors attributable to the conversion process.

Applied to the SynTagRus/SynTagRus-UD pair, the method identified six types of conversion errors — three persisting in the current converter and three concentrated in the old part — as well as ten groups of annotation inconsistencies.

While the method allowed to detect conversion errors with high precision, it has several limitations. First, the current analysis is restricted to the test set; extending it to the training data would require a cross-validation setup with multiple models trained on complementary splits so that every sentence is evaluated exactly once. Second, we achieved full token-level alignment between SynTagRus and SynTagRus-UD, but this may not always be possible for other corpus pairs. Similarly, the arc direction alignment procedure (Section 4.2) has not been formally analyzed: it is not clear whether it covers all structural differences between corpora. Fourth, by restoring ellipsis we excluded from the analysis conversion errors involving the `orphan` relation and related constructions, since investigating these would require substantial modifications to the original SynTagRus annotation. Moreover, for corpora where ellipsis restoration is not possible, the mask token strategy would have to be used, which yields lower parsing performance for elliptical constructions; improving this strategy is a direction for future work.

Despite these limitations, the method provides a practical and scalable tool for conversion error detection. In future work, we plan to apply the method to the current version of SynTagRus, which has undergone substantial revision since the versions used for the UD conversion.

Acknowledgements

This work was supported by the RSF grant No. 24-18-00988.

References

Furkan Akkurt, Onur Gungor, Büşra Marşan, Tunga Gungor, Balkiz Ozturk Basaran, Arzucan Özgür, and Susan Uskudarli. 2024. Evaluating the quality of a corpus annotation scheme using pretrained language models. // Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, P 6504–6514, Torino, Italia, May. ELRA and ICCL.

Chiara Alzetta, Felice Dell'Orletta, Simonetta Montemagni, and Giulia Venturi. 2017. Dangerous relations in dependency treebanks. // Jan Hajič, *Proceedings of the 16th International Workshop on Treebanks and Linguistic Theories*, P 201–210, Prague, Czech Republic.

Þórunn Arnardóttir, Hinrik Hafsteinsson, Atli Jasonarson, Anton Ingason, and Steinθór Steingrímsson. 2023. Evaluating a Universal Dependencies conversion pipeline for Icelandic. // Tanel Alumäe and Mark Fishel, *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, P 698–704, Tórshavn, Faroe Islands, May. University of Tartu Library.

Igor M. Boguslavsky, Pavel V. Djachenko, Evgenia S. Inshakova, Leonid L. Iomdin, Alexander V. Lazurski, Leonid G. Mityushin, Andrey A. Movsesyan, Ivan P. Rygaev, Victor G. Sizov, Svetlana P. Timoshenko, Tatyana I. Frolova, and Alexandra V. Chaga. 2024. The current state of the SynTagRus corpus. *Trudy instituta russkogo yazyka im. V.V. Vinogradova*, (4):141–169.

Emanuel Borges Völker, Maximilian Wendt, Felix Hennig, and Arne Köhn. 2019. HDT-UD: A very large Universal Dependencies treebank for German. // Alexandre Rademaker and Francis Tyers, *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, P 46–57, Paris, France, August. Association for Computational Linguistics.

Adriane Boyd, Markus Dickinson, and W Detmar Meurers. 2008. On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137.

António Branco, João Ricardo Silva, Luís Gomes, and João António Rodrigues. 2022. Universal grammatical dependencies for Portuguese with CINTIL data, LX processing and CLARIN support. // Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Jan Odijk, and Stelios Piperidis, *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, P 5617–5626, Marseille, France, June. European Language Resources Association.

Flavio Massimiliano Cecchini, Timo Korkiakangas, and Marco Passarotti. 2020. A new Latin treebank for Universal Dependencies: Charters between Ancient Latin and Romance languages. // Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, P 933–942, Marseille, France, May. European Language Resources Association.

Marie-Catherine de Marneffe, Matias Grioni, Jenna Kanerva, and Filip Ginter. 2017. Assessing the annotation consistency of the Universal Dependencies corpora. // Simonetta Montemagni and Joakim Nivre, *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, P 108–115, Pisa, Italy, September. Linköping University Electronic Press.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308, June.

Kira Droganova and Daniel Zeman. 2016. Conversion of syntagrus (the russian dependency treebank) to universal dependencies. Technical report, Technical report.—Institute of Formal and Applied Linguistics, Faculty of ....

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian ud treebanks. // *Proceedings of the 17th international workshop on treebanks and linguistic theories (tlt 2018)*, volume 155, P 53–66. Linköping University Electronic Press Linköping, Sweden.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023. Annotation error detection: Analyzing the past and present for a more coherent future. *Computational Linguistics*, 49(1):157–198, March.

Guy Lapalme. 2021. Validation of Universal Dependencies by regeneration. // Miryam de Lhoneux and Reut Tsarfaty, *Proceedings of the Fifth Workshop on Universal Dependencies (UDW, SyntaxFest 2021)*, P 109–120, Sofia, Bulgaria, December. Association for Computational Linguistics.

Siyao Peng and Amir Zeldes. 2018. All roads lead to UD: Converting Stanford and Penn parses to English Universal Dependencies with multilayer annotations. // Agata Savary, Carlos Ramisch, Jena D. Hwang, Nathan Schneider, Melanie Andresen, Sameer Pradhan, and Miriam R. L. Petruck, *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, P 167–177, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. // Daniel Zeman and Jan Hajič, *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, P 197–207, Brussels, Belgium, October. Association for Computational Linguistics.

Hakyung Sung and Gyu-Ho Shin. 2025. Second language Korean Universal Dependency treebank v1.2: Focus on data augmentation and annotation scheme refinement. // Špela Arhar Holdt, Nikolai Ilinykh, Barbara Scalvini, Micaella Bruton, Iben Nyholm Debess, and Crina Madalina Tudor, *Proceedings of the Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2025)*, P 13–19, Tallinn, Estonia, March. University of Tartu Library, Estonia.

Daniil Timchenko and Andrei Movsesian. 2024. Issledovanie kačestva konvertacii sintaksičeskoj razmetki korpusa SynTagRus v format Universal'nyh zavisimostej [A study on syntactic annotation conversion quality of the SynTagRus corpus to the Universal Dependencies format]. // *Sbornik trudov 48-j meždisciplinarnoj školy-konferencii IPPI RAN*, P 329–340, Moscow.

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. // Anne Abeille, Thorsten Brants, and Hans Uszkoreit, *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, P 48–55, Centre Universitaire, Luxembourg, August. International Committee on Computational Linguistics.

Houquan Zhou, Yu Zhang, Zhenghua Li, and Min Zhang. 2020. Is POS Tagging Necessary or Even Helpful for Neural Dependency Parsing? // Xiaodan Zhu, Min Zhang, Yu Hong, and Ruifang He, *Natural Language Processing and Chinese Computing*, P 179–191, Cham. Springer International Publishing.