Extending Retag to Conversion Error Detection: A Case Study on SynTagRus Morphology

Andrei Movsesian[1], **Daniil Timchenko**[2]
[1]Institute for Information Transmission Problems (IITP RAS)
[2]Moscow Institute of Physics and Technology (MIPT)
Correspondence: Andrei Movsesian (andrei.movsesian@gmail.com)

Abstract

Linguistically annotated corpora are often converted between annotation schemes, but errors introduced during conversion can compromise their reliability. While annotation error detection is a well-studied topic, conversion error detection remains largely unexplored. We adapt the Retag method, which is traditionally used for finding annotation errors, to identify conversion errors by comparing model performance on original and converted versions of the same corpus, aligned at the token level. Applying this approach to the SynTagRus corpus converted to Universal Dependencies, we achieve high-precision detection of conversion errors in morphological annotation. Our analysis reveals systematic errors in distinction of auxiliary verbs, pronouns, numerals, and multi-word named entities, and uncovers previously undocumented annotation inconsistencies between different sections of the corpus. The method can be applied to any converted dataset for which an aligned source is available, providing an efficient way to target conversion errors for manual correction without exhaustive inspection.

Keywords: conversion error detection, annotation error detection, Retag, Universal Dependencies, SynTagRus, morphology, sequence tagging

1. Introduction

When converting a corpus from one annotation scheme to another, such as when creating Universal Dependencies (UD) resources from existing treebanks (de Marneffe et al., 2021), annotation errors can arise from two distinct sources. First, inaccuracies may be inherited from the source annotation. Second, new errors can be introduced by the conversion process itself.

For teams developing and maintaining converted corpora, distinguishing between these two error types is a practical necessity because they demand different responses. A conversion error points to a flaw in the conversion algorithm itself; fixing it once improves all future conversions. As for the source-side errors, the best course of action is to trace the error to its source and, when possible, notify the original maintainers. Fixing the error upstream ensures that future conversions inherit the correction automatically.

While the conceptual distinction between source and conversion errors is straightforward, isolating them in practice is not. Standard methods for evaluating converted data can reveal that errors exist, but they struggle to distinguish between different error types. Annotation error detection (AED) methods based on machine learning models (Klie et al., 2023) are especially problematic in this context: when applied to converted corpora, they conflate source errors, conversion errors, and the model's own prediction inaccuracies, producing a noisy signal that hinders targeted debugging and improvement.

In this paper, we address the specific problem of isolating conversion errors. We propose a simple but effective method that leverages the original annotations as a direct point of comparison. For that we align the source and converted corpora at the token level and run the same AED model on both corpora. By examining cases where their annotations diverge, we can identify discrepancies where a conversion error is the most probable explanation.

We demonstrate this method on the conversion of the SynTagRus corpus (Boguslavsky et al., 2024) to UD, focusing on morphological annotations. Our case study shows that this approach successfully pinpoints conversion errors that would otherwise be buried in the output of standard AED, enabling their systematic analysis and correction.

2. Related Work

Work on evaluating converted corpora can be grouped into three broad approaches, each with different goals and limitations.

Manual inspection. The most direct approach to conversion quality assessment is manual annotation of a sample. For example, Borges Völker et al. (2019) manually annotated 50 sentences of the Hamburg Dependency Treebank according to UD guidelines and compared them with automatic conversion output. Cecchini et al. (2018) manually annotated 994 sentences of the Index Thomisticus Treebank to compare two conversion algorithms.

Proceedings of the Ninth Workshop on Universal Dependencies (UDW 2026), pages 305–314
May 16, 2026.

While such efforts can estimate overall conversion quality and identify some errors, they are labor-intensive and do not scale to corpus-level debugging. More fundamentally, manual inspection alone cannot systematically distinguish whether a discovered error originated in the source or was introduced during conversion.

Downstream task evaluation. A common proxy for annotation quality is to measure performance on tasks that depend on the annotation, such as parsing (Johannsen et al., 2015; Türk et al., 2019), morphological tagging (Pyysalo et al., 2015; McCarthy et al., 2018), or language modeling (Akkurt et al., 2024). Improvements in downstream performance are taken as evidence of improved annotation quality. This approach is useful for tracking progress across corpus versions, but it provides only an aggregate signal and do not produce interpretable error diagnoses at the instance level. Moreover, the drop in model's performance does not necessarily mean that there are errors. For example, the annotation could simply become more difficult for the model to analyze.

Cross-corpus comparison. A third line of work compares annotations across different corpora that follow the same guidelines. Droganova et al. (2018) compared syntactic annotations across Russian UD corpora (including SynTagRus), proposing to analyze differences in parser attachment scores to discriminate between parser errors and genuine annotation inconsistencies. Aggarwal and Zeman (2020) introduced a symmetric measure for comparing part of speech (PoS) annotation consistency between corpora and demonstrated its use in identifying specific annotation differences. While these methods are effective, they are designed for comparing independently annotated corpora. In the conversion setting, however, we have an additional resource: the source corpus itself, which allows us to condition our analysis on the original annotation when isolating conversion-induced changes.

Like downstream evaluation approaches, we use a model (an AED classifier) to surface potential errors. But unlike prior work, we apply the same model to both the source and converted corpora and compare its outputs. This comparison cancels out the model's own prediction errors, revealing discrepancies that are attributable to the conversion process.

3. SynTagRus Alignment

3.1. Text Alignment

SynTagRus (Boguslavsky et al., 2024) is a fully disambiguated corpus of Russian featuring multiple annotation layers, including morphology and syntax. Its development began in 1998 and as of 2026 it consists of more than 111,000 sentences (more than 1,585,000 words excluding punctuation marks) from 1392 texts across various genres. These genres include 20th-21st century fiction, contemporary popular science literature, opinion journalism, biographies, and news feeds. SynTagRus is an autonomous part of the Russian National Corpus, with search functionality available via its online interface.[1] For the purposes of conversion error detection, the corpus data was converted from its native XML format into the CoNLL-U format. This conversion was purely structural and did not alter morphological annotation.

The initial conversion of SynTagRus to the Universal Dependencies format was developed in 2016 as part of the UD v1.3 release (Droganova and Zeman, 2016). This first version contained texts annotated up to 2014, along with two texts annotated in 2015 (more than 61,000 sentences in total). As the UD guidelines evolved, the corpus annotation was subsequently corrected.

With the release of UD v2.9 in 2021, portions of SynTagRus annotated up to 2020 were incorporated, providing additionally over 25,000 sentences. However, the older segments of the corpus were not reconverted so changes in the original SynTagRus annotation of old texts were not reflected in the UD version. To align the two resources, we therefore compiled two distinct versions of SynTagRus (from 2016 and 2021) and aligned the texts with the UD version (SynTagRus-UD, v2.16).

3.2. Token Alignment

After aligning the texts, the number of texts and sentences in SynTagRus and SynTagRus-UD became identical. However, the number of tokens differed due to differences in annotation principles. To align them, we adopted the notion of tokens as defined in UD and adjusted the annotation of SynTagRus accordingly. The main differences in tokenization are as follows:

1. **Punctuation marks.** UD treats each punctuation mark as a token and assigns a special PUNCT tag to them, while SynTagRus do not annotate them. For alignment we copy the punctuation marks annotation from SynTagRus-UD to SynTagRus.
2. **Multi-word expressions (MWEs).** UD guidelines prohibit spaces within a token's form, while SynTagRus contains numerous MWEs (e.g., *kak by to ni bylo* 'anyhow', *ni razu* 'never'). During the initial conversion to UD, these MWEs were split into separate tokens using specially devised rules. Subsequently, in 2022,

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

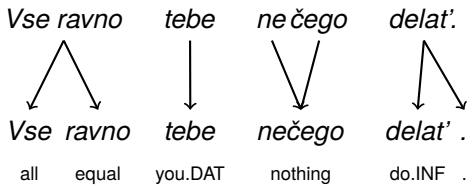
similar splitting rules were created for the original SynTagRus annotation to enhance the search functionality of the Russian National Corpus (Boguslavsky et al., 2024) and were applied during our alignment process.

3. **Syntactic agglomerates** such as *nečego* 'nothing' and *negde* 'nowhere'. This case is the opposite of MWEs: in SynTagRus, such tokens are split into two for syntactic reasons. For example, *negde* is annotated as *ne* 'no' and *gde* 'where'. During alignment, we merged these token pairs and assigned the morphological features of the right-hand token.

4. **Ellipsis.** In SynTagRus, omitted words are restored in syntactically disconnected elliptical sentences. They function as normal nodes in the syntactic structure, but their word form is left empty. In SynTagRus-UD, this information is not present in the basic structure. However, since our focus is on morphological annotation, we do not need to align the elided tokens, and therefore they were omitted.

5. **Other differences.** The original conversion process introduced some errors into SynTagRus-UD (e.g., token duplication and using full form in abbreviation). For the purpose of strict token-by-token alignment, these errors were replicated semi-automatically in the SynTagRus data.

Figure 1 shows the main alignment operations. In the end, the aligned version of SynTagRus contained exactly the same number of sentences and tokens as SynTagRus-UD from UD v2.16.



'All the same, you have nothing to do.'

Figure 1: Token alignment between SynTagRus (top) and SynTagRus-UD (bottom), showing three main operations: MWE split, token merge, and punctuation addition.

4. Method

4.1. Conversion Error Detection

Our approach to conversion error detection extends the Retag method for annotation error detection (van Halteren, 2000; Klie et al., 2023). While Retag identifies potential annotation errors by comparing original annotations with model predictions on a single dataset, we adapt this methodology to detect errors introduced during dataset conversion.

Given a source dataset src and its converted version cnv, we train two independent taggers θ_{src} and θ_{cnv} on src and cnv respectively. This yields four annotations for each token t: the original annotation y_t^{src}, the converted annotation y_t^{cnv}, and the model predictions $\hat{y}_t^{src} = \theta_{src}(t)$ and $\hat{y}_t^{cnv} = \theta_{cnv}(t)$. We categorize each token into one of four cases based on agreement patterns:

1. **Full agreement**: $y_t^{src} = \hat{y}_t^{src}$ and $y_t^{cnv} = \hat{y}_t^{cnv}$. This suggests both the original annotation and conversion are correct.

2. **Potential conversion error**: $y_t^{src} = \hat{y}_t^{src}$ but $y_t^{cnv} \neq \hat{y}_t^{cnv}$. The model agrees with the original annotation but disagrees with the converted version, indicating a likely error introduced during conversion.

3. **Source-side error**: $y_t^{src} \neq \hat{y}_t^{src}$ but $y_t^{cnv} = \hat{y}_t^{cnv}$. This may occur if: (1) the conversion makes the annotation task easier for the model, or (2) an annotation error in src was manually corrected in cnv.

4. **Noisy error**: $y_t^{src} \neq \hat{y}_t^{src}$ and $y_t^{cnv} \neq \hat{y}_t^{cnv}$. Both models disagree with their respective annotations, suggesting inherent ambiguity, source-side error, or otherwise difficult token for the models to process. These provide no reliable signal about conversion quality.

Focusing on case 2 allows us to obtain a high-precision candidate set for manual inspection, reducing the search space for conversion errors.

4.2. Conversion Errors and Data Inconsistency

The SynTagRus-UD corpus consists of two subcorpora converted at different times: an older subcorpus and a newer subcorpus processed with an updated conversion script. Although both subcorpora underwent manual validation after conversion, numerous inconsistencies between them remain unresolved. These residual inconsistencies dominate the error landscape, making it difficult to identify genuine conversion errors (Timchenko and Movsesyan, 2024).

To distinguish between these inconcistencies and actual errors introduced by the conversion process we stratify our analysis by creating three variants: **full** (the entire corpus), **new** (the newer subcorpus only), and **old** (the older subcorpus only). For a given triple of annotations $(y_t^{src}, y_t^{cnv}, \hat{y}_t^{cnv})$, we count how frequently it is flagged as a potential

conversion error in each variant. The resulting distribution helps to identify the underlying nature of the inconsistency:

1. **Data inconsistency**: the triple appears as an error only in the full variant, indicating that the new and old variants use different but internally consistent annotation conventions. When merged, these differences surface as errors. We analyze these errors for two reasons. First, old and new variants ideally should follow the same annotation guidelines so their divergence is itself a conversion issue. Second, conditioning on src and the ability to train the model on the full variant reduces noise compared to standard Retag approaches (e.g., using θ_{cnv}^{old} to annotate cnv^{new} or vice versa), providing a cleaner detection of annotation inconsistencies.
2. **Conversion error**: the error appears in the full variant and at least one of the old/new variants. We further distinguish two subtypes:
   - **Genuine conversion error**, caused by a bug in the conversion script. The error may appear in the old variant only, the new variant only, or both.
   - **Conversion-induced model error**: the conversion script is correct, but the converted corpus introduces new features not present in src. Unlike noisy errors (where $y_t^{src} \neq \hat{y}_t^{src}$), here θ_{src} predicts the tag correctly while θ_{cnv} fails. These errors are systematic and interpretable: a linguist can trace them to specific differences in annotation conventions. Since the inventory of PoS tags and morphological features for Russian is almost identical between UD v1 (old) and UD v2 (new), conversion-induced model errors typically appear in all three variants (old, new, and full).

4.3. Training and Evaluation

For training the models we chose UDPipe 2.1 (Straka, 2018), a widely used tool for processing Universal Dependencies data. We trained separate UDPipe models with default hyperparameters for each of the six subcorpora (three variants for both source and converted datasets), using the predefined train/dev/test splits provided in the UD release. Since each variant in SynTagRus-UD already follows the standard 80/10/10 split, no resplitting was necessary. Table 1 shows the statistics for all splits and variants.

Since our focus is exclusively on morphological annotation, we disabled all other UDPipe components (lemmatization and parsing) during training.

Table 1: Corpus statistics by variant and split. Since we fully aligned SynTagRus and SynTagRus-UD, the statistics is correct for both of them.

Variant	Split	Sentences	Tokens
Old	Train	48,815	870,434
	Dev	6,584	118,484
	Test	6,491	117,320
	Total	61,890	1,106,238
New	Train	20,816	334,123
	Dev	2,322	34,833
	Test	2,309	40,386
	Total	25,447	409,342
Full	Train	69,631	1,204,557
	Dev	8,906	153,317
	Test	8,800	157,706
	Total	87,337	1,515,580

All analysis is conducted on test set data, as taggers tend to overfit on training and development sets, potentially biasing the error signal. While the current evaluation serves to demonstrate the validity of our approach, exhaustive conversion error detection would employ cross-validation or repeated random subsampling to ensure every token is evaluated as part of some model's test set, thereby achieving full corpus coverage.

The test set used is identical to the standard split in SynTagRus-UD. We did not perform any prior error analysis on this set before the study. Instead, our goal was to manually verify the errors identified by our algorithm and compare them against standard AED approach.

5. Results and Discussion

5.1. Models Performance

Table 2 shows accuracy metrics for all variants of SynTagRus and SynTagRus-UD calculated separately for parts of speech (UPOS), other morphological features (FEATS), and at tag level (UPOS + FEATS). The model trained on SynTagRus has the best performance on the full variant, while the model trained on SynTagRus-UD performs best on the old variant, so adding new data did not increase the performance, indicating data inconsistencies between the variants.

Table 3 shows how many errors correspond to each of the error type defined in section 4.1. The least frequent errors are source-side errors, and conversion error is the only error type that increases on the full variant, again suggesting data inconsistency. In the next sections we will only analyze potential conversion errors.

Table 2: Accuracy metrics for all variants of SynTagRus and SynTagRus-UD (%). Best results for each corpus are in bold.

Field	SynTagRus			SynTagRus-UD		
	Old	New	Full	Old	New	Full
UPOS	99.35	99.13	**99.35**	**99.14**	98.93	98.90
FEATS	98.03	97.54	**98.15**	**97.50**	96.87	96.52
UPOS + FEATS	97.79	97.19	**97.90**	**97.30**	96.60	96.26

Table 3: Error type distribution for each variant and each metric of (SynTagRus, SynTagRus-UD) pair (%).

Variant	Field	Error Type		
		Conversion	Source	Noisy
Old	POS	0.4	0.2	0.5
	FEATS	1.1	0.6	1.4
	POS+FEATS	1.1	0.6	1.6
New	POS	0.4	0.2	0.6
	FEATS	1.5	0.8	1.7
	POS+FEATS	1.5	0.9	1.9
Full	POS	0.6	0.2	0.5
	FEATS	2.2	0.5	1.3
	POS+FEATS	2.2	0.6	1.5

5.2. Errors in Parts of Speech

To detect potential conversion errors in parts of speech, we use the following procedure:

1. Collect $(y^{src}_{PoS}, y^{cnv}_{PoS}, \hat{y}^{cnv}_{PoS})$ triples as defined in section 4.1 for each variant of the (SynTagRus, SynTagRus-UD) pair.

2. Count how frequently each triple is flagged as a potential conversion error.

3. Exclude any triples for which there are no variants with absolute frequency of at least 10 and relative frequency of at least 2%. Here relative frequency is the count of conversion errors for $(y^{src}_{PoS}, y^{cnv}_{PoS}, \hat{y}^{cnv}_{PoS})$ divided by the total number of $(y^{src}_{PoS}, y^{cnv}_{PoS})$ pairs. We filter by both raw counts and relative frequencies to exclude rare triples: for example, if a given $(y^{src}_{PoS}, y^{cnv}_{PoS})$ pair appears only once in the test set and that token is misclassified, the relative frequency for the corresponding triple would be 100%, highlighting categories where the model's performance is least reliable.

4. Group the errors as defined in section 4.2 and analyze them manually.

The results of our analysis is present in Table 4. In the following sections we will analyze each conversion error type separately.

5.2.1. Data Inconsistency

Both cases of data inconsistency are similar and involve words *uže* 'already', *eŝe* 'else/yet', *počti* 'almost', *takže* 'also', and *čut'* 'hardly'. In SynTagRus they are all considered as particles. However, in the old variant of SynTagRus-UD they were all converted to adverbs, while in the new variant they are particles.

5.2.2. Genuine Conversion Errors

In SynTagRus, auxiliary verbs are marked as verbs. In that sense, the triple (V, AUX, VERB) can be attributed to conversion-induced model errors. However, additional ambiguity introduced by the annotation principles of UD does not mean that there are no errors. In SynTagRus-UD, only the lexeme *byt'* 'to be' receives the POS tag AUX. At the same time, it can function as a main verb, and some of these cases are mislabeled, as in Example (1).

(1) Kogda mne bylo sem'-vosem' let...
*AUX

'When I was seven or eight years old...'

The (A, PROPN, ADJ) triple represents a case of erroneous annotation of multi-word named entities in the older version of SynTagRus-UD. According to UD guidelines, adjectives within such entities should not be marked as proper nouns. For example, in the multi-word expression *Plehanovskaja akademija* 'Plekhanov academy', the

Table 4: PoS conversion errors (counts with % in parentheses). Percentages are relative to the total number of tokens with given pair of PoS in two annotations. Model errors were separated from conversion errors after manual inspection.

y^{src}_{PoS}	y^{cnv}_{PoS}	$\hat{y}^{cnv}_{PoS}$	Old	New	Full	Total
			Data inconsistency			
PART	PART	ADV	1 (0.0%)	0 (0.0%)	103 (2.1%)	104
PART	ADV	PART	0 (0.0%)	1 (25.0%)	68 (13.7%)	69
			Genuine conversion errors			
V	AUX	VERB	32 (3.6%)	9 (2.2%)	56 (4.3%)	97
A	PROPN	ADJ	23 (19.5%)	0 (0.0%)	34 (28.8%)	57
NUM	NUM	ADJ	8 (0.4%)	1 (0.3%)	45 (2.0%)	54
S	DET	PRON	13 (1.5%)	2 (0.6%)	30 (2.4%)	45
NUM	ADJ	NUM	13 (2.1%)	0 (0.0%)	17 (2.6%)	30
			Conversion-induced model errors			
S	PROPN	NOUN	39 (1.0%)	23 (2.1%)	55 (1.1%)	117
ADV	NUM	ADV	4 (9.5%)	3 (8.3%)	23 (29.5%)	30
S	PART	PRON	7 (12.1%)	4 (13.8%)	11 (12.6%)	22

first token should be marked as an adjective. This error was corrected in UD v2.17 by the maintainers of SynTagRus-UD.

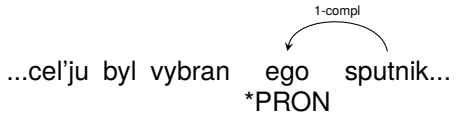
Errors in (NUM, NUM, ADJ) and (NUM, ADJ, NUM) triples are because of adjectival ordinal numerals. In SynTagRus all numerals which do not contain letters are numerals. In SynTagRus-UD these cases were converted to adjectives, but inconsistently. Most errors are in the old variant. Example (2) shows annotation of one such case, where both numerals should be adjectives.

(2) ...načinaja s 1 sentjabrja 2006 goda.
*NUM ADJ

'...starting from September 1, 2006.'

(S, DET, PRON) errors are due to incomplete heuristics used to disambiguate between substantive (PRON) and attributive (DET) pronouns. It commonly occurs in the old variant, but the errors in the new variant are similar. For example, the disambiguation is incorrect when attributive pronoun is a direct object of a noun, marked in SynTagRus by the first completive (1-compl) syntactic relation, as in Example (3).

(3)



'...the target was its satellite...'

5.2.3. Conversion-induced Model Errors

These errors mostly occur in cases where annotation principles of SynTagRus-UD introduce additional ambiguity. (S, PROPN, NOUN) errors do not occur in SynTagRus because proper nouns there are marked as nouns (S).

(ADV, NUM, ADV) errors occur in borderline cases with words like *mnogo* 'many' and *malo* 'few' which are ambiguous in SynTagRus-UD (cardinal numerals or adverbs) but marked as adverbs in SynTagRus.

(S, PART, PRON) errors occur in nominal sentences with zero copula, in which demonstrative *eto* 'this' is converted from noun to particle in SynTagRus-UD.

5.3. Errors Below the Threshold

To conserve space, we analyzed only the most frequent errors and excluded 73 additional error types, comprising 1018 erroneous tokens in total across the three variants. Most of these errors are duplicates of the frequent ones (9 types, 577 tokens). For example, the triple (V, AUX, VERB) has a less frequent counterpart (V, VERB, AUX) with the same underlying explanation.

Another common category consists of noisy errors in ambiguous cases (44 types, 378 tokens), where the annotation in SynTagRus and SynTagRus-UD is identical, but the model prediction differs. One example is the triple (A, ADJ, VERB), where the choice between a short adjective and a past participle sharing the same form (e.g., *urezannom* 'truncated') is ambiguous in the absence of syntactic dependents. While each of the 43 triples occurs relatively rarely, these nevertheless represent noisy errors for which the pre-

diction of θ_{src} is actually correct. One possible explanation is the difference in feature distributions between the two corpora, though further investigation is needed for these cases, as well as for the remaining rare cases (20 types, 63 tokens).

5.4. Errors in Other Morphological Features

For other morphological features we follow the same procedure as for parts of speech, but with three modifications:

1. For simplicity, we analyzed only those morphological categories which are present in both SynTagRus and SynTagRus-UD: Animacy, Aspect, Case, Degree, Gender, Mood, Number, Person, and VerbForm.

2. Since different parts of speech have different morphological feature inventories, and UDPipe model predicts PoS and all other features as two separate tags, for each category we counted only those errors for which $y^{src}_{PoS} = \hat{y}^{src}_{PoS}$ and $y^{cnv}_{PoS} = \hat{y}^{cnv}_{PoS}$.

3. For better grouping of errors, instead of counting triples $(y^{src}_t, y^{cnv}_t, \hat{y}^{cnv}_t)$, we included part of speech and counted triples $([y^{src}_{PoS}, y^{src}_t], [y^{cnv}_{PoS}, y^{cnv}_t], [\hat{y}^{cnv}_{PoS}, \hat{y}^{cnv}_t])$.

The counts for these errors are present in Table 5.

The triple (A,Inan; DET,Inan; DET,∅) represents a data inconsistency error. In SynTagRus, demonstratives like *ètot* 'this' are annotated as adjectives and receive animacy marking in the accusative case (e.g., animate accusative *ètogo* vs. inanimate accusative *ètot*). When converting to SynTagRus-UD, these adjectives become determiners (DET), and the animacy feature is carried over from the original annotation. However, the old version of SynTagRus-UD fails to preserve animacy for some determiners due to a conversion error, resulting in missing animacy features (∅).

The other three are model errors and are explained by the absence of proper nouns and determiners in SynTagRus.

5.5. Annotation Errors and Conversion Errors

Up until now we analyzed potential conversion errors. All of them were either genuine conversion errors or model errors introduced because of the conversion process (specific to SynTagRus-UD).

Potential conversion errors and noisy errors (as defined in section 4.1) together represent all possible annotation errors in SynTagRus-UD which can be obtained by a standard Retag method. Here we will analyze them focusing on parts of speech to see if there are any benefits in analyzing potential conversion errors separately compared to standard Retag approach where noisy errors cannot be easily excluded. Since in Retag the source annotation is not available, we only have access to the $(y^{cnv}_{PoS}, \hat{y}^{cnv}_{PoS})$ pairs ($y^{cnv}_{PoS} \neq \hat{y}^{cnv}_{PoS}$). Table 6 shows these errors with absolute frequency of at least 10 and relative frequency of at least 2% in any of the variant.

All of these frequent errors were described earlier. (PROPN, NOUN) is a conversion-induced model error, (AUX, VERB) is a conversion error, while (PART, ADV) is a data inconsistency error. The other four conversion errors were not found.

In theory, this approach identifies all model errors. However, since it is not conditioned on the source annotation, conversion errors appear among the less frequent pairs and are mixed in with noisy errors. For example, the (NUM, ADJ, NUM) conversion error was relatively frequent when conditioned on correctly predicted PoS tag in SynTagRus. However, the number of tokens with the error type (ADJ, NUM) is negligible (0.13%) relative to the total number of adjectives in the corpus.

The same holds for absolute frequencies. For example, the (A, PROPN, ADJ) conversion error type consisted of 23 tokens in the older version, almost all of which were genuine conversion errors. In contrast, for the (PROPN, ADJ) pair there are 29 such tokens. Manual verification confirmed that 6 additional tokens mostly represent noisy errors, making error analysis more difficult since they must be filtered manually. This problem also occurs for morphological features, for example, in cases of ambiguity such as genitive versus accusative in Russian.

5.6. Correcting the Errors

While we are not the maintainers of SynTagRus-UD and have proposed a tool only for efficient detection of conversion errors, it is nevertheless important to discuss how such errors can be corrected, since the ultimate goal of any error detection method is to improve the dataset.

We believe that analyzing the old conversion script is not necessary; using the new script would automatically resolve errors related to the old conversion process, including data inconsistencies.

This leaves two error types that require further attention: (V, AUX, VERB) and (S, DET, PRON). Most (V, AUX, VERB) errors stem from incorrect syntactic annotation in SynTagRus. In SynTagRus-UD, the decision to mark the verb *byt'* 'to be' as an auxiliary depends on the presence of a copular syntactic relation in SynTagRus in which the verb is the head. In many cases, this relation is marked erroneously.

Table 5: Conversion errors of morphological features other than PoS (counts with % in parentheses). Percentages are relative to the total number of tokens with given pair of features in two annotations. ∅ indicates that the feature is not present in the corresponding annotation. The feature names of SynTagRus besides PoS are displayed in UD style.

$y^{src}_{PoS}, y^{src}_{t}$	$y^{cnv}_{PoS}, y^{cnv}_{t}$	$\hat{y}^{cnv}_{PoS}, \hat{y}^{cnv}_{t}$	Old	New	Full	Total
		Animacy errors				
A,Inan	DET,Inan	DET,∅	2 (1.6%)	0 (0.0%)	105 (39.6%)	107
S,Anim	PROPN,Anim	PROPN,Inan	2 (0.1%)	38 (5.3%)	24 (0.9%)	64
S,Inan	PROPN,Inan	PROPN,Anim	6 (0.3%)	6 (2.7%)	10 (0.5%)	22
A,Neut	DET,Neut	DET,Masc	3 (0.6%)	10 (4.4%)	10 (1.4%)	23

Table 6: Annotation errors in SynTagRus-UD for parts of speech (counts with % in parentheses). Percentages are relative to the total number of tokens with given PoS in the annotation.

y^{cnv}_{PoS}	$\hat{y}^{cnv}_{PoS}$	Old	New	Full	Total
PROPN	NOUN	47 (1.1%)	23 (2.1%)	64 (1.2%)	134
PART	ADV	5 (0.2%)	5 (0.3%)	110 (2.2%)	120
AUX	VERB	32 (3.1%)	9 (2.0%)	56 (3.7%)	97

For the (S, DET, PRON) error type, the solution lies in more careful handling of attributive pronouns within the conversion script, as discussed in Section 5.2.2.

Correcting all errors would require three steps: fixing the conversion script, reporting the source-side errors to SynTagRus maintainers (if they have not yet been addressed), and reconverting the corpus using the updated resources.

However, implementing this scenario is not straightforward. First, the portions of SynTagRus used for conversion in 2016 and 2021 have since undergone substantial changes in the original data. Missing parts of texts and sentences have been added, and numerous manual corrections to the annotations have been made.

Second, the presence of source-side errors that do not appear in SynTagRus-UD indicates that some errors were corrected in the converted data rather than in the source. This complicates reconversion, as applying it to the modern version of SynTagRus (or even the 2016 and 2021 snapshots) would require tracking and preserving the manual changes made to the current version of SynTagRus-UD, which poses a challenge.

Ideally, all corrections to the converted corpus should be implemented within the conversion script itself, allowing the development of SynTagRus and SynTagRus-UD to remain independent yet aligned. We acknowledge, however, that this is not always feasible.

6. Conclusion

We have introduced a simple but effective method for isolating conversion errors in annotated corpora. Our key insight is to apply the same annotation error detection model to both the source and converted corpora and compare their outputs. Because the model's prediction errors are largely consistent across both runs, discrepancies between the two AED outputs provide evidence of conversion errors. This turns the source corpus into a resource uniquely available in the conversion setting that prior work has not fully exploited.

We demonstrated the method on the conversion of SynTagRus to Universal Dependencies, focusing on morphological annotation. Our analysis revealed:

- Genuine conversion errors in the annotation of auxiliary verbs, pronouns, ordinal numerals, and multi-word named entities that survived manual validation and would otherwise require exhaustive inspection to locate.
- Data inconsistencies between variants, such as the treatment of adverbs vs. particles, which our stratified analysis successfully distinguished from conversion errors.
- Model errors introduced by the conversion process itself, where new annotation distinctions created ambiguity that the tagger could not resolve.

Crucially, we showed that standard Retag applied to the converted corpus alone would have missed many of these errors since noise in the annotation errors alters their frequency distribution. This confirms that our comparative approach provides information unavailable through conventional AED.

At the same time, our proof-of-concept has several limitations that point to directions for future research. First, we analyzed parts of speech and individual morphological features separately rather than full morphological tags; extending the method to joint prediction would provide a more holistic view. Second, we did not investigate source-side errors, though these could be valuable for upstream corpus maintenance. Third, we used Retag as our AED method, but our comparative framework is agnostic to the underlying error detection technique; exploring other AED methods within this framework could reveal whether certain detectors are better suited to the conversion setting. Finally, exhaustive error detection would require the comprehensive analysis of less frequent errors and cross-validation or repeated subsampling to achieve full corpus coverage; we leave this to future implementation.

Our method generalizes to any token-classification task and could extend to dependency parsing with appropriate relation alignment. For corpus developers, it offers a practical, interpretable method to systematically improve conversion quality without manual inspection of entire datasets.

7. Bibliographical References

Akshay Aggarwal and Daniel Zeman. 2020. Estimating POS annotation consistency of different treebanks in a language. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 93–110, Düsseldorf, Germany. Association for Computational Linguistics.

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. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6504–6514, Torino, Italia. ELRA and ICCL.

Igor M. Boguslavsky, Alexandra V. Chaga, Pavel V. Djachenko, Tatyana I. Frolova, Evgenia S. Inshakova, Leonid L. Iomdin, Alexander V. Lazurski, Leonid G. Mityushin, Andrey A. Movsesyan, Ivan P. Rygaev, Victor G. Sizov, and Svetlana P. Timoshenko. 2024. The current state of the SynTagRus corpus. *Proceedings of the V. V. Vinogradov Russian Language Institute*, 42(4):141–169.

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

Flavio Massimiliano Cecchini, Marco Passarotti, Paola Marongiu, and Daniel Zeman. 2018. Challenges in converting the index Thomisticus treebank into Universal Dependencies. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 27–36, Brussels, Belgium. Association for Computational Linguistics.

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

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian ud treebanks. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018)*, pages 53–66, Linköping, Sweden. Linköping University Electronic Press.

Kira Droganova and Daniel Zeman. 2016. Conversion of SynTagRus (the Russian dependency treebank) to Universal Dependencies. Technical report, Praha, Czechia.

Anders Johannsen, Héctor Martínez Alonso, and Barbara Plank. 2015. Universal dependencies for danish. In *Proceedings of the Fourteenth International Workshop on Treebanks and Linguistic Theories (TLT14), 11–12 December 2015 Warsaw, Poland*, pages 157–167.

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.

Arya D. McCarthy, Miikka Silfverberg, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2018. Marrying Universal Dependencies and Universal Morphology. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 91–101, Brussels, Belgium. Association for Computational Linguistics.

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 163–172, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

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.

Daniils Timchenko and Andrey Movsesyan. 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]. In *Sbornik trudov 48-j meždisciplinarnoj školy-konferencii IPPI RAN*, pages 329–340, Moscow.

Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Balkız Öztürk Başaran, Tunga Güngör, and Arzucan Özgür. 2019. Improving the annotations in the Turkish Universal Dependency treebank. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 108–115, Paris, France. Association for Computational Linguistics.

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.