

How Well Do Language Models Handle Romanian Diacritics? A Comparison of Monolingual and Multilingual Architectures

Alexandra Toma, Maria Cristina Popescu

Faculty of Mathematics and Computer Science

University of Bucharest

alexandra.toma1@s.unibuc.ro

maria-cristina.popescu@s.unibuc.ro

Abstract

Romanian orthography relies on five diacritic characters (ă, î, â, ș, ț) that are frequently omitted in informal digital text, yet the impact of text cleaning decisions regarding diacritics on Romanian language models has not been systematically studied across architectures and tasks. We address this gap by examining how diacritics affect subword tokenization, contextual embeddings, and three downstream tasks - POS tagging, named entity recognition, and semantic textual similarity - comparing Romanian BERT with mBERT, DistilmBERT, and mT5. Each model is fine-tuned on diacritic-preserving and diacritic-stripped data and evaluated on both, isolating both overall performance and robustness to train-test mismatch.

This work provides the first systematic, multi-task account of how diacritic effects propagate from tokenization to downstream performance, showing this depends jointly on pretraining corpus and architecture, and offers practical guidance for Romanian NLP systems that must handle diacritic-inconsistent text.

1 Introduction

Romanian is a Romance language spoken by roughly 24 million people and written using five diacritic characters: ă, î, â, ș, and ț. In informal digital communication such as social media and SMS these characters are often omitted and replaced by their ASCII counterparts (Tufiş and Ceaşu, 2007). This is especially relevant for tweet-based and social-media NLP, where Romanian-specific pretraining has recently shown strong gains on Romanian Twitter data (Tăiatu et al., 2023).

Diacritics in Romanian carry real linguistic information. Minimal pairs such as *fata* and *față* (en: 'the girl' and 'face') show that removing diacritics may change meaning or alter grammatical form. Although diacritic restoration has been studied for years ((Ruseti et al., 2020), (Petrică et al., 2014)),

much less attention has been paid to how modern transformer models internally handle Romanian text when diacritics are present or absent. In particular, it remains unclear whether multilingual and Romanian-specific models differ in their tokenization behavior, embedding stability, and downstream robustness under diacritic variation.

This paper addresses that gap through three research questions:

- **Tokenization** - How does diacritic presence affect subword segmentation and does this differ between multilingual and monolingual models?
- **Embeddings** - How similar are the contextual representations produced for sentence pairs that differ only in diacritics?
- **Performance** - How does diacritic removal affect performance under matched and mismatched training and evaluation conditions?

To answer these questions, we evaluate Romanian BERT (Dumitrescu et al., 2020), mBERT (Devlin et al., 2019), DistilmBERT (Sanh et al., 2019), and mT5 (Xue et al., 2021). Our intrinsic analysis combines tokenization metrics and embedding similarity measured on strictly audited sentence pairs.

The study makes two main contributions. First, it compares how monolingual and multilingual models respond to diacritic variation at both the tokenization and representation levels. Second, it evaluates whether these differences affect Part of Speech Tagging (POS), Named Entity Recognition (NER) and Semantic Textual Similarity (STS) performance under matched and mismatched training and evaluation conditions. Together, these analyses provide a unified view of how Romanian diacritics influence transformer models from input segmentation to downstream performance.

2 Related Work

Romanian NLP has increasingly relied on pre-trained transformers, especially multilingual models such as mBERT and Romanian-specific models such as Romanian BERT. Multilingual models share their vocabulary and representational capacity across many languages, while monolingual models can dedicate more capacity to Romanian forms, including diacritic-bearing subwords. Romanian-specific pretraining has obtained strong results on tasks including sentiment analysis, dialect identification, and diacritics restoration (Masala et al., 2020). Recent multilingual work also shows that broader pretraining can improve cross-lingual transfer, but it does not specifically address Romanian orthographic variation (Conneau et al., 2020).

LiRo provides a broad Romanian benchmark spanning text classification, NER, POS tagging, dependency parsing, machine translation, question answering, and semantic textual similarity (Dumitrescu et al., 2021). The DIACRITICS-RO¹ dataset provides aligned fields intended for diacritic restoration. However, restoration data cannot automatically be treated as a clean diacritics-only perturbation dataset: source-target pairs may also differ in spelling, punctuation, whitespace, or historical orthography. This motivates the audit introduced in this work.

The downstream resources are the Romanian RRT treebank distributed through Universal Dependencies for POS (Nivre et al., 2020), RONEC for NER (Dumitrescu and Avram, 2020), and RO-STS for semantic textual similarity (Dumitrescu et al., 2021; Dumitrescu, 2021). Together, they allow us to measure whether the intrinsic effects of diacritic removal persist across token classification and sentence-level semantic prediction.

The literature also leaves open how orthographic noise propagates through transformer pipelines. Existing Romanian-specific work and benchmark studies generally compare models on task scores, but rarely connect tokenizer behavior, embedding stability and downstream robustness in one analysis (Dumitrescu et al., 2020; Masala et al., 2020). This is important because diacritic removal can alter subword segmentation, change contextual representations and affect entity boundaries in NER. Our study addresses this gap by comparing multilingual and monolingual transformers on paired

Romanian sentences with and without diacritics.

3 Tokenization and Representation Analysis

3.1 Dataset

We use a fixed 10,000-pair sample from the DIACRITICS-RO dataset released on Hugging Face. The dataset is designed for Romanian diacritic restoration and contains paired text examples: an input sentence without Romanian diacritics and a corresponding target sentence with restored diacritics. The full dataset contains approximately 341K examples. For this study, we selected 10,000 rows from the training split.

The local sample contains the input and target sentences but does not include the original source field. Table 1 summarizes the explicit audit. Of the 10,000 pairs, 9,118 satisfy the strict criterion and 882 are excluded because they contain whitespace, spelling, punctuation, or other textual differences. Applying the documented 3–250-word length filter removes a further 136 pairs, leaving 8,982 examples for the tokenization and representation analyses.

Audit outcome	Pairs
Valid diacritics-only	9,118
Whitespace difference	783
Spelling difference	70
Punctuation difference	28
Other difference	1

Table 1: Audit of the fixed 10,000-pair DIACRITICS-RO sample. The 9,118 valid rows sum with the four exclusion categories to 10,000.

3.2 Methodology

3.2.1 Models

Table 2 presents the four models shared by the intrinsic and downstream analyses. Romanian BERT is the monolingual reference. mBERT and DistilmBERT share a multilingual WordPiece vocabulary, while the latter has half as many encoder layers. mT5 provides a multilingual SentencePiece encoder-decoder contrast.

¹<https://huggingface.co/datasets/klusai/diacritics-ro>

Model	Family	Layers	Vocab.
Romanian BERT	Mono.	12	50K
mBERT	Multi.	12	119K
DistilBERT	Multi.	6	119K
mT5-base	Multi.	12	250K

Table 2: Models used in the intrinsic and downstream analyses. “Layers” refers to encoder depth.

3.2.2 Pair Normalization and Audit

For every candidate pair, both strings are normalized to Unicode NFC. Legacy Romanian cedilla characters are standardized ($\text{\textcircled{S}} \rightarrow \text{\textcircled{S}}$ with comma below and $\text{\textcircled{T}} \rightarrow \text{\textcircled{T}}$ with comma below, including uppercase forms), and whitespace is normalized separately. Excluded pairs are categorized as punctuation, whitespace, spelling, or other differences. We additionally count legacy cedillas, combining characters, non-breaking spaces, and unusual punctuation before and after cleaning.

3.2.3 Tokenization and Representation Measures

For each retained pair, we tokenize both variants with the official tokenizer of every model. We compute fertility (Rust et al., 2021),

$$\text{fertility}(s) = \frac{|T(s)|}{|W(s)|},$$

token inflation (Lundin et al., 2026),

$$\text{inflation}(s) = \frac{|T(s_d)| - |T(s_{nd})|}{\max(|T(s_{nd})|, 1)},$$

the number of unknown tokens, the single-character token rate, and the Jaccard overlap between token-type sets. All per-sentence outputs retain the example identifier and source.

For the representation analysis, we extract an architecture-neutral anchor representation and an attention-mask-weighted mean representation. For BERT-like models, the anchor is the classification token; because mT5 has no CLS token, its anchor is the first non-special lexical token. For each pair we compute cosine similarity between diacritized and stripped representations. We also analyze focus-word distances on Romanian minimal pairs, similarity across relative encoder depth, the association with diacritic density, and cross-model agreement using Kendall’s τ .

3.3 Results

The tokenization results are reported in Table 3. Romanian BERT produces substantially fewer subwords than the multilingual models. Adding correct

diacritics reduces its mean token count by 1.16 tokens and its fertility from 1.53 to 1.47. In contrast, the multilingual tokenizers become less compact: the mean token count rises by 1.72 for mBERT and DistilBERT and by 3.02 for mT5.

Model	Tokens		Fertility		UNK/sent.		Δ token
	ND	D	ND	D	ND	D	D–ND
RoBERT	30.75	29.58	1.53	1.47	0.03	0.03	-1.16
mBERT	40.10	41.82	1.96	2.07	0.12	0.12	1.72
DistilBERT	40.10	41.82	1.96	2.07	0.12	0.12	1.72
mT5	39.57	42.59	1.92	2.08	0.00	0.00	3.02

Table 3: Audited tokenization results on 8,982 strict diacritics-only pairs after the documented length filter. ND denotes no diacritics and D denotes diacritics.

Figure 1 further illustrates the opposite effect of diacritics on monolingual and multilingual tokenization. Romanian BERT achieves lower fertility when diacritics are present, indicating that diacritized Romanian words are represented using fewer subwords. Fertility increases for all three multilingual tokenizers, with the largest increase observed for mT5.

mBERT and DistilBERT have identical tokenization statistics because DistilBERT reuses the mBERT tokenizer and vocabulary. Their subsequent representational behavior can nevertheless differ because their encoder architectures and depths are not identical. The UNK rate does not change between the two orthographic conditions for any model: Romanian BERT produces 0.03 unknown tokens per sentence, mBERT and DistilBERT produce 0.12, and mT5 produces none. Consequently, the observed tokenization differences are primarily caused by changes in subword segmentation.

At the corpus-vocabulary level, the Jaccard overlap between the sets of subword types used in the diacritized and stripped conditions is 64.50% for Romanian BERT, 83.63% for mT5, and 88.09% for both mBERT and DistilBERT. Thus, high aggregate vocabulary overlap does not imply identical sentence-level segmentation.

The representation analysis uses the same 8,982 audited pairs and model set. As shown in Table 4, DistilBERT is the most stable model under the orthographic perturbation, with an anchor similarity of 0.98 and a mean-pooled similarity of 0.95. Although mBERT also has high mean anchor similarity (0.96), its larger standard deviation and negative minimum indicate substantial variation across

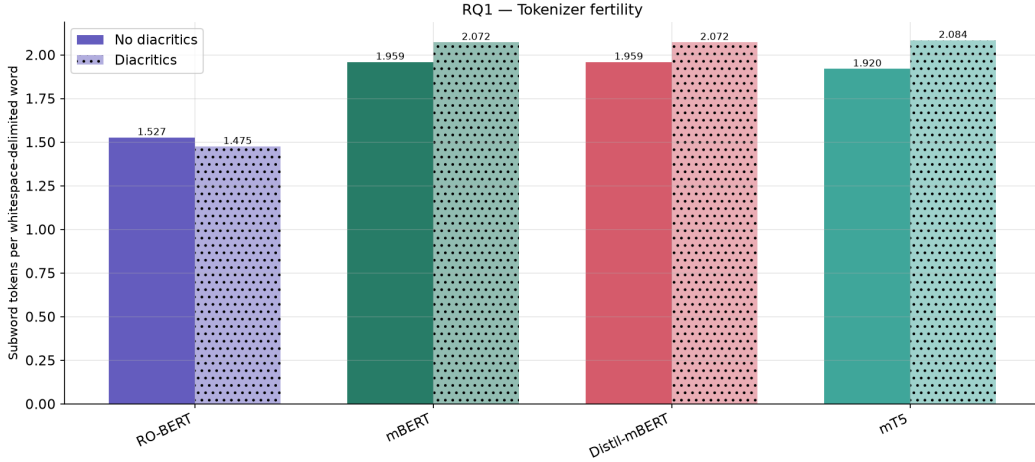


Figure 1: Tokenizer fertility with and without Romanian diacritics on the strictly audited pairs.

individual pairs. Romanian BERT is more consistent than mBERT but exhibits a larger average representational change than DistilmBERT.

At the lexical level, mT5 is the most sensitive model. Its focus-word cosine distance reaches 0.48, compared with 0.26 for mBERT, 0.22 for DistilmBERT, and 0.18 for Romanian BERT. It also has the lowest mean anchor similarity and the largest variance. Because mT5 has no classification token, its anchor is defined as the first non-special lexical token. Its mean-pooled similarity is therefore the more directly comparable sentence-level measure.

Model	Anchor	Std.	Min.	Mean	Word dist.
RoBERT	0.92	0.03	0.64	0.91	0.18
mBERT	0.96	0.12	-0.12	0.90	0.26
DistilmBERT	0.98	0.06	0.13	0.95	0.22
mT5	0.85	0.25	-0.25	0.90	0.48

Table 4: Cosine similarity between diacritized and stripped representations. Higher sentence similarity indicates greater invariance; higher focus-word distance indicates greater lexical sensitivity.

Figure 2 shows how representation similarity changes across relative encoder depth. For Romanian BERT, mBERT, and mT5, the minimum similarity occurs in the final encoder layer, whereas DistilmBERT reaches its minimum at layer 3 of 6. The effect of diacritic removal is therefore not restricted to the input embeddings, but its propagation through the encoder depends on the model architecture.

Diacritic density is negatively associated with representation similarity for DistilmBERT ($r = -0.256$), mBERT ($r = -0.205$), and Romanian BERT ($r = -0.087$). Sentences containing a

higher proportion of diacritic-bearing characters therefore tend to undergo larger representational changes in these models.

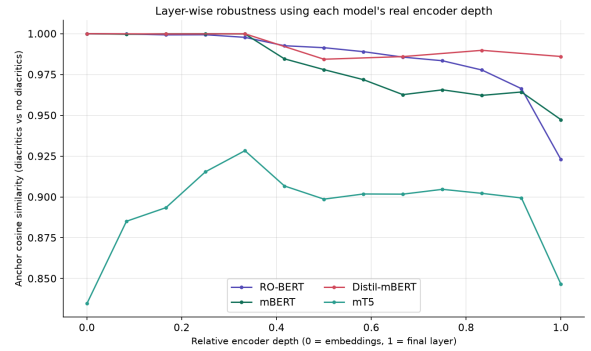


Figure 2: Layer-wise cosine similarity between diacritized and stripped representations, aligned by relative encoder depth.

Taken together, the tokenization and representation results show that orthographic robustness cannot be inferred from tokenizer efficiency alone. Romanian BERT produces the most compact segmentation of diacritized text, but DistilmBERT achieves the highest average representation similarity. Moreover, mBERT and DistilmBERT share identical tokenization statistics while exhibiting different layer-wise behavior. mT5 combines the largest diacritic-induced token increase with the greatest focus-word displacement. These differences suggest that the effect of Romanian diacritics depends on both tokenizer design and encoder architecture. The following performance analysis examines whether these intrinsic differences are also reflected in POS, NER and STS outcomes.

4 Performance variation analysis

4.1 Dataset

To determine whether the effects of Romanian diacritics observed in model representations also influence downstream performance, we evaluate three tasks at different levels of linguistic analysis: part-of-speech tagging (POS) at the token level, named entity recognition (NER) at the entity level, and semantic textual similarity (STS) at the sentence-pair level.

For POS tagging, we use the Romanian RRT Universal Dependencies treebank (Dumitrescu et al., 2020). It contains 8,043 training sentences, 752 validation sentences, and 729 test sentences.

For NER, we use RONEC, a Romanian corpus annotated with 16 named-entity categories (Dumitrescu and Avram, 2020). The experimental splits contain 9,000 training sentences, 1,330 validation sentences, and 2,000 test sentences.

For STS, we use RO-STIS², which contains Romanian sentence pairs annotated with similarity scores. The experimental splits contain 5,749 training pairs, 1,500 validation pairs, and 1,379 test pairs.

Task	Train	Dev	Test	Primary metric
POS	8,043	752	729	Token F1
NER	9,000	1,330	2,000	Entity F1
STS	5,749	1,500	1,379	Pearson/MAE

Table 5: Downstream datasets and evaluation metrics. The diacritized and stripped conditions use aligned splits of identical size.

4.2 Methodology

For each task, every model is fine-tuned on two aligned dataset variants: the original version containing Romanian diacritics (*diac*) and a corresponding version from which only Romanian diacritics are removed (*nodi*). We apply this transformation to RONEC, RoRefTrees, and RO-STIS while preserving the original dataset splits and annotations. This design produces eight task-specific fine-tuned variants - four models under two training conditions - for each downstream task.

Each fine-tuned model is evaluated on both the *diac* and *nodi* versions of the corresponding test set. This results in four training - evaluation conditions: *diac* $\rightarrow$ *diac*, *diac* $\rightarrow$ *nodi*, *nodi* $\rightarrow$ *diac*,

and *nodi* $\rightarrow$ *nodi*. The matched conditions measure performance when training and evaluation follow the same orthographic convention, whereas the mismatched conditions measure the models' robustness to changes in diacritic usage.

We evaluate Romanian BERT, mBERT, DistilmBERT, and mT5. The same data splits and orthographic transformations are used across models. Models are selected according to validation performance, and final metrics are computed on the corresponding held-out test split.

For POS tagging, we report token-level F1. For NER, we report entity-level strict F1. Errors are additionally classified as missed entities, spurious entities, incorrect entity types, and incorrect entity boundaries. For STS, we report Pearson and Spearman correlation together with MAE and RMSE. STS predictions are retained at the example level to analyze signed errors, absolute errors, and systematic prediction bias.

4.3 Results

4.3.1 Part Of Speech Tagging

Figures 3 and 4 visually present the variation of performance across training and evaluation conditions, both for UPOS and XPOS tagging. Detailed numeric results can be found in the Appendix (Figures 7 and 8). Matched training-evaluation conditions consistently outperform mismatched ones for both UPOS and XPOS, with RoBERT achieving the best and most stable scores (UPOS: 0.980/0.978; XPOS: 0.967/0.962 for *diac/nodi*). The *diac* $\rightarrow$ *nodi* mismatch causes the largest degradation across models, especially for XPOS: mBERT drops 11.5 points (0.961 $\rightarrow$ 0.846) and DistilmBERT 9.9 points (0.954 $\rightarrow$ 0.856), versus only 2.1 points for RoBERT. The reverse mismatch (*nodi* $\rightarrow$ *diac*) is consistently milder for every model.

This gradient aligns with pretraining data. RoBERT was pretrained on clean, diacritic-preserving Romanian text, giving stable subword representations for diacritic-marked forms. mBERT and DistilmBERT were pretrained on Wikipedia texts with diacritics largely stripped, leaving their embedding space underspecified for diacritic-bearing subwords. mT5 falls in between: it was pretrained on mC4, a Common Crawl-derived corpus that preserves diacritics in principle but reflects noisy, real-world Romanian web text, itself a mix of correctly diacritized and stripped forms. This partial, inconsistent exposure likely

²https://huggingface.co/datasets/dumitrescustefan/ro_sts

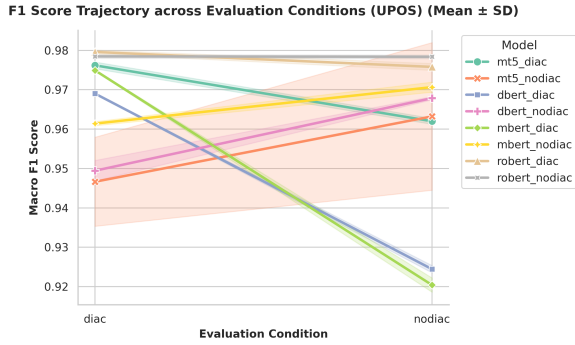


Figure 3: Macro F1 trajectories (UPOS) from *diac* (left) to *nodiak* (right) evaluation, for models fine-tuned on either orthographic condition. Translucent shaded bands represent standard deviation ($\pm$ SD) across runs.

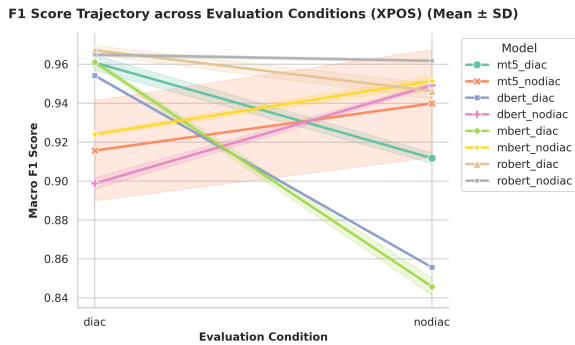


Figure 4: Macro F1 trajectories (XPOS) from *diac* (left) to *nodiak* (right) evaluation, for models fine-tuned on either orthographic condition. Translucent shaded bands represent standard deviation ($\pm$ SD) across runs.

explains why mT5’s degradation under *diac* $\rightarrow$ *nodiak* mismatch is intermediate — better than the fully diacritic-stripped multilingual encoders, but still clearly worse than RoBERT’s uniformly clean pretraining signal.

XPOS is more affected than UPOS across all models, since fine-grained morphosyntactic tags (case, mood, definiteness) often depend on diacritic contrasts, whereas coarse UPOS categories remain largely recoverable from context alone.

These results indicate that diacritic robustness is driven less by fine-tuning condition than by whether diacritics were present during pretraining — clean and uniform exposure to diacritics (RoBERT) yields the most stable transfer, systematically stripped (mBERT, DistilmBERT) yields the least, and noisy/mixed exposure (mT5, via mC4) yields intermediate robustness.

4.3.2 Named Entity Recognition

Figure 5 (supported by the numeric results shown in Figures 9 from Appendix) reveals that, like in the

case of POS tagging, matched training–evaluation conditions generally outperform mismatched ones. *Diac*-trained mBERT achieves the highest overall F1 (0.8713, *diac* $\rightarrow$ *diac*), while RoBERT shows the most stable performance across both training regimes (0.8685 *diac* $\rightarrow$ *diac*, 0.8692 *nodiak* $\rightarrow$ *nodiak*). DistilmBERT is consistently the weakest performer across all four conditions (0.77–0.83), indicating that its reduced model capacity limits entity recognition quality regardless of orthographic condition.

The *diac* $\rightarrow$ *nodiak* mismatch produces a clear degradation for models fine-tuned on diacritic-preserving data: mBERT drops 3.9 points (0.871 $\rightarrow$ 0.832), DistilmBERT 2.4 points (0.792 $\rightarrow$ 0.768), mT5 1.9 points (0.835 $\rightarrow$ 0.816), and RoBERT only 0.8 points (0.869 $\rightarrow$ 0.860). In contrast, *nodiak*-trained models show flat or slightly improving performance when evaluated on *nodiak* text relative to *diac* text — mBERT_*nodiak* and DistilmBERT_*nodiak* both increase by roughly 0.7–3.6 points, and RoBERT_*nodiak* is essentially invariant (0.8692 in both conditions).

Studying the rates of error types for each category (Figures 10 and 11 in Appendix) unveils the fact that across all models and conditions, missed entities are the dominant error type for nearly every class (typically 40–75% of errors), while wrong_boundary and wrong_type errors make up the remainder in varying proportions depending on the entity class.

For models trained on *diac* data and evaluated on *nodiak* text, the missed-entity share increases consistently across most classes, at the expense of wrong_boundary errors. For mbert_train_diac, DATETIME rises from 59.4% to 69.3% missed (with wrong_boundary falling from 35.5% to 26.1%), and similar shifts appear for ORDINAL (75.5 $\rightarrow$ 79.4), NUMERIC (68.7 $\rightarrow$ 71.0), and PERSON (60.8 $\rightarrow$ 65.4). This pattern indicates that diacritic removal primarily degrades recall - entities that would otherwise be identified (even if imperfectly bounded) go undetected entirely.

Certain categories - MONEY, QUANTITY, FACILITY, and LOC - show elevated wrong_type shares (30–60%) across all models and both evaluation conditions, indicating that these errors stem from intrinsic semantic overlap between adjacent categories (e.g., MONEY/QUANTITY/NUMERIC, FACILITY/LOC/ORG). This class-level confusability is therefore a property of the tagset itself and persists

independently of orthographic condition.

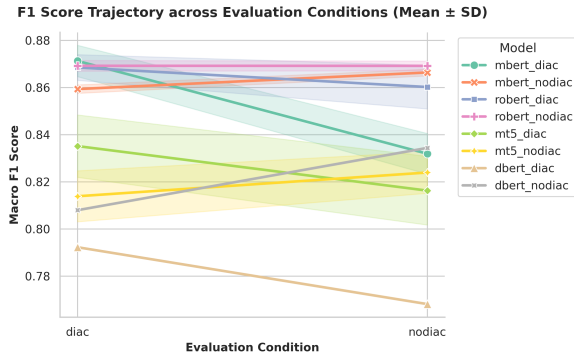


Figure 5: Macro F1 trajectories from *diac* (left) to *nodiac* (right) evaluation, for models fine-tuned on either orthographic condition. Translucent shaded bands represent standard deviation ($\pm$ SD) across runs.

4.3.3 Semantic Textual Similarity

Table 6 presents the STS results for all completed training and evaluation conditions. We report Pearson correlation, mean absolute error (MAE), and prediction bias. Better performance is indicated by higher Pearson correlation, lower MAE, and bias closer to zero.

Romanian BERT achieves the best overall results. Its Pearson correlation ranges from 0.791 to 0.803, and it has the lowest MAE in all completed conditions. It is also relatively stable when the evaluation condition changes.

mBERT and DistilMBERT are more affected by a mismatch between training and evaluation text. When trained with diacritics and evaluated without them, mBERT’s MAE increases from 0.801 to 0.883, while DistilMBERT’s MAE increases from 0.789 to 0.851. When trained without diacritics, both models perform better on the corresponding stripped test set.

All configurations have positive bias, meaning that the models tend to predict similarity scores that are too high. This effect is strongest for mT5, which has bias values of 0.678 and 0.692.

Figure 6 summarizes the MAE results and makes the matched and mismatched conditions easier to compare. The heatmap confirms that Romanian BERT has the lowest error across the completed configurations. It also shows that models trained with diacritics consistently incur higher error when evaluated on stripped text. For models trained without diacritics, the direction is reversed: evaluation on stripped text generally yields the lower MAE.

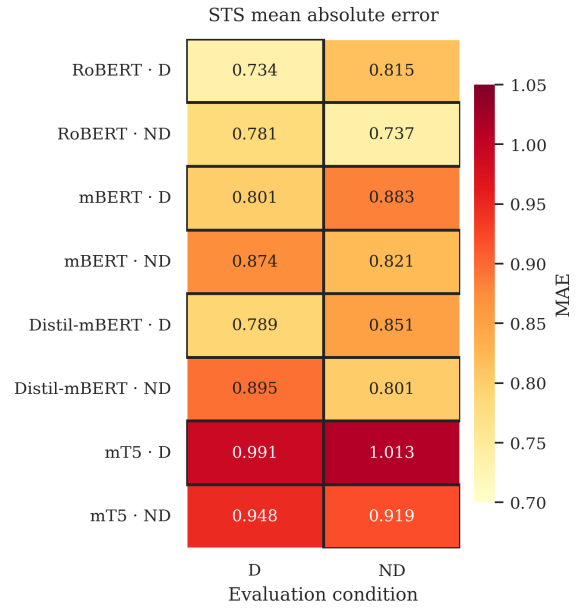


Figure 6: Mean absolute error on the STS task across training and evaluation conditions. Rows identify the model and training condition, while columns represent evaluation on diacritized (D) and stripped (ND) text. Lower values indicate better performance.

5 Conclusion

This study examined how Romanian diacritic removal affects transformer-based NLP systems across three dimensions - subword tokenization, contextual embeddings, and three downstream tasks (POS tagging, NER, STS) - comparing a monolingual model (RoBERT) against three multilingual architectures (mBERT, DistilMBERT, mT5).

Across all three downstream tasks, a consistent pattern emerged: matched training–evaluation conditions outperform mismatched ones, and diacritic-trained models degrade more sharply when evaluated on stripped text than diacritic-stripped models do in the reverse direction. This asymmetry was most severe for XPOS tagging and NER, where fine-grained morphosyntactic and entity-boundary decisions rely heavily on diacritic-marked surface forms, and was also confirmed for STS, where diacritic-trained mBERT and DistilMBERT showed substantially higher MAE under mismatch, while *nodiac*-trained variants transferred better in both directions. RoBERT was consistently the most robust model across all tasks, showing minimal performance change regardless of training or evaluation condition — a pattern we linked to its pretraining on clean, diacritic-preserving Romanian text, in

Model	Train	Eval diac			Eval nodiac		
		<i>r</i>	MAE	Bias	<i>r</i>	MAE	Bias
Romanian BERT	diac	0.799	0.734	0.232	0.791	0.815	0.427
	nodiac	0.803	0.781	0.423	0.803	0.737	0.287
mBERT	diac	0.743	0.801	0.195	0.707	0.883	0.289
	nodiac	0.735	0.874	0.420	0.736	0.821	0.226
DistilmBERT	diac	0.749	0.789	0.102	0.718	0.851	0.201
	nodiac	0.708	0.895	0.404	0.737	0.801	0.061
mT5	diac	0.719	0.991	0.678	0.715	1.013	0.692
	nodiac	0.733	0.948	0.619	0.737	0.919	0.535

Table 6: STS performance and prediction bias under matched and mismatched training and evaluation conditions. Higher Pearson correlation (r) and lower MAE indicate better performance. Bias is the mean signed residual $\hat{y} - y$; positive values indicate systematic overestimation of semantic similarity.

contrast to mBERT and DistilmBERT’s diacritic-stripped Wikipedia pretraining and mT5’s noisy, mixed-diacritic mC4 corpus.

Diacritics have opposite effects on monolingual versus multilingual segmentation: they make RoBERT’s tokenization more compact (lower fertility), but inflate multilingual token counts, most severely for mT5. However, tokenizer efficiency alone does not predict representational or downstream robustness: DistilmBERT, despite sharing mBERT’s tokenizer and vocabulary exactly, achieves the highest representational stability under diacritic perturbation (anchor similarity 0.98), while RoBERT - the most efficient tokenizer for diacritized text - shows a larger average representational shift than DistilmBERT. This dissociation indicates that encoder depth and architecture also shape how diacritic perturbations propagate through the network - a conclusion reinforced by the layer-wise analysis, where DistilmBERT’s minimum similarity occurs mid-network (layer 3 of 6) and not necessarily in the final layer, as seen in all other models.

mT5 stands out as uniquely sensitive at the lexical level, combining the largest diacritic-induced token inflation with the greatest focus-word representational displacement (cosine distance 0.48) and the strongest positive bias in STS predictions - consistent with its exposure to inconsistently diacritized web text during pretraining, which appears to leave its representations less anchored to a single canonical (diacritized or stripped) form.

Taken together, these results show that robustness to Romanian diacritic removal is not a single, unified property of a model, but emerges from the interaction of pretraining corpus composition, tokenizer design, and encoder depth. Two practi-

cal levers emerge, applicable at different stages of model development. Where architecture choice is open, pretraining on diacritic-preserving Romanian text, as with RoBERT, is the more effective route to orthographic robustness overall: this robustness holds regardless of the fine-tuning condition used, since RoBERT’s *diac*- and *nodiac*-trained variants perform almost identically across both evaluation conditions. Where a multilingual backbone must be used instead - for instance, in multilingual or cross-lingual applications - fine-tuning on stripped or mixed-orthography data is the more effective mitigation available within that constraint. It does not close the gap with a diacritic-aware monolingual model, but it substantially reduces the *diac* $\rightarrow$ *nodiac* performance penalty that is otherwise largest for these architectures across all three tasks studied.

Limitations

This study is limited to encoder-based, masked language models (Romanian BERT, mBERT, DistilmBERT) and one encoder-decoder architecture (mT5). Decoder-only architectures were not included, as extending this analysis to generative, autoregressive models would likely require a different experimental setup - for instance, evaluation on tasks such as question answering or zero-shot classification, and a distinct approach to prompting and output parsing under diacritic variation. We leave this extension to future work.

All downstream experiments in this study use full fine-tuning, where every model parameter is updated during training. As a result, the downstream task results alone cannot cleanly separate how much of a model’s diacritic sensitivity stems from its pretrained representations versus from the

fine-tuning process itself adapting to the orthographic condition seen in training - a distinction our tokenization and representation analyses, conducted on frozen pretrained models, address only partially, since they do not involve any task-specific adaptation. Future work should compare full fine-tuning against a frozen-encoder setting, where only a lightweight classification head is trained on top of fixed pretrained representations, to more directly isolate whether diacritic robustness is primarily a property of the pretrained embeddings or is instead shaped by task-specific adaptation during fine-tuning.

Finally, our *diac* and *nodiac* conditions represent two clean, internally consistent extremes, whereas real-world informal Romanian text typically mixes diacritized and non-diacritized words within the same document, sentence, or even word, and may also contain incorrect or partial diacritic substitutions. Future work should evaluate models under this mixed-orthography setting - for example, by randomly or systematically alternating diacritic presence across tokens in the same input - to better approximate the noisy conditions models encounter in deployment.

Use of AI Assistance

We used ChatGPT and Claude for language editing and proofreading to improve the readability of the manuscript. We also used ChatGPT and Gemini to assist in writing Python scripts for data preprocessing and plotting figures. The authors reviewed and edited all outputs, and take full responsibility for the contents of this paper.

Acknowledgments

This research is supported by InstRead: Research Instruments for Text Complexity, Simplification, and Readability Assessment CNCS - UEFISCDI project number PN-IV-P2-2.1-TE-2023-2007.

References

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.

Stefan Dumitrescu, Andrei-Marius Avram, and Sampo Pyysalo. 2020. [The birth of Romanian BERT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4324–4328, Online. Association for Computational Linguistics.

Stefan Daniel Dumitrescu. 2021. [RO-STTS: Romanian semantic textual similarity dataset](#). Hugging Face Datasets.

Stefan Daniel Dumitrescu and Andrei-Marius Avram. 2020. [Introducing RONEC - the Romanian named entity corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4436–4443, Marseille, France. European Language Resources Association.

Stefan Daniel Dumitrescu, Petru Rebeja, Beata Lorincz, Mihaela Gaman, Andrei Avram, Mihai Ilie, Andrei Pruteanu, Adriana Stan, Lorena Rosia, Cristina Iacobescu, Luciana Morogan, George Dima, Gabriel Marchidan, Traian Rebedea, Madalina Chitez, Dani Yogatama, Sebastian Ruder, Radu Tudor Ionescu, Razvan Pascanu, and Viorica Patraucean. 2021. [Liro: Benchmark and leaderboard for romanian language tasks](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Jessica M. Lundin, Ada Zhang, Nihal Karim, Hamza Louzan, Guohao Wei, David Ifeoluwa Adelani, and Cody Carroll. 2026. [The token tax: Systematic bias in multilingual tokenization](#). In *Proceedings of the 7th Workshop on African Natural Language Processing (AfricaNLP 2026)*, pages 103–112, Rabat, Morocco. Association for Computational Linguistics.

Mihai Masala, Stefan Ruseti, and Mihai Dascalu. 2020. [RoBERT - a Romanian BERT model](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6626–6637, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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 Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Eugen Petrică et al. 2014. [A robust diacritics restoration system using unreliable raw text corpora](#). In *Proceedings of SLTU*.

Stefan Ruseti, Teodor-Mihai Cotet, and Mihai Dascalu. 2020. [Romanian diacritics restoration using recurrent neural networks](#). *Preprint*, arXiv:2009.02743.

Phillip Rust, Ivan Vulić, Jonas Frohberg, Florian Freudenthaler, and Hinrich Schütze. 2021. [How good is your tokenizer? analysing the performance of multilingual models beyond XLM-R](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3260–3273, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. [DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter](#). *ArXiv*, abs/1910.01108.

Iulian-Marius Tăiatu, Andrei-Marius Avram, Dumitru-Clementin Cercel, and Florin Pop. 2023. [RoBERT-Tweet: A BERT language model for romanian tweets](#). In *Proceedings of the 2023 International Conference on Database and Expert Systems Applications*.

Dan Tufiş and Alexandru Ceaşu. 2007. Diacritics restoration in romanian texts. *A Common Natural Language Processing Paradigm for Balkan Languages*, pages 49–55.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

6 Training configuration

For each configuration, we fine-tuned each model three times with different seeds to account for the stability of results. The training parameters were based on those provided in the *Romanian-Transformers Model Evaluation* Google Colab file³ (used in the *Romanian Transformers* collaborative project led by Dumitrescu Stefan), which was also a starting point for the fine-tuning process. The learning rate value was set to $2e - 5$, with a factor of $\frac{2}{3}$ and a patience of 5. The batch size was 8 and early stopping was applied with a patience of 5 epochs. The fine-tuning process was run on Tesla T4 and A100 GPUs.

³https://colab.research.google.com/drive/1fa9Yd_7xTSCxwmqoX8ptrZsr0SfpB6_0?usp=sharing

7 Results and error analysis

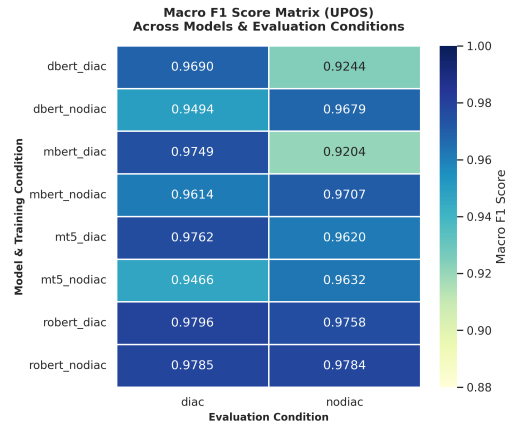


Figure 7: Macro F1 score matrices for UPOS tagging across models, training conditions (*diac/nodiac*), and evaluation conditions (*diac/nodiac*).

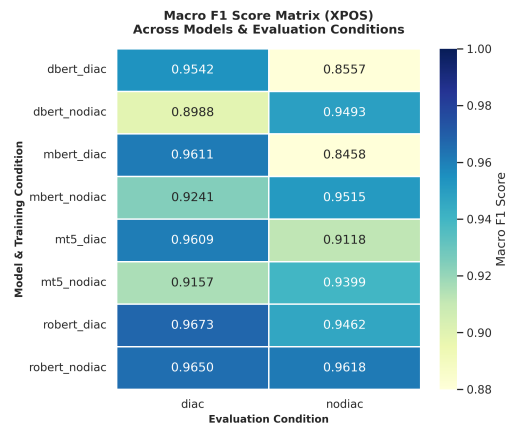


Figure 8: Macro F1 score matrices for XPOS tagging across models, training conditions (*diac/nodiac*), and evaluation conditions (*diac/nodiac*).

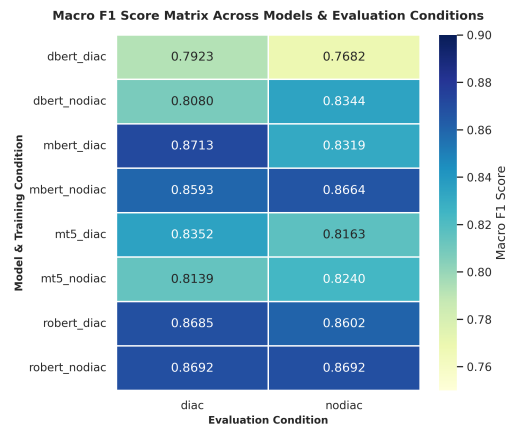


Figure 9: Macro F_1 score performance matrix across fine-tuning conditions (*diac* vs. *nodiac*) and evaluation setups for all evaluated model architectures.

Faceted Error Type Breakdown Heatmap (% per Class)

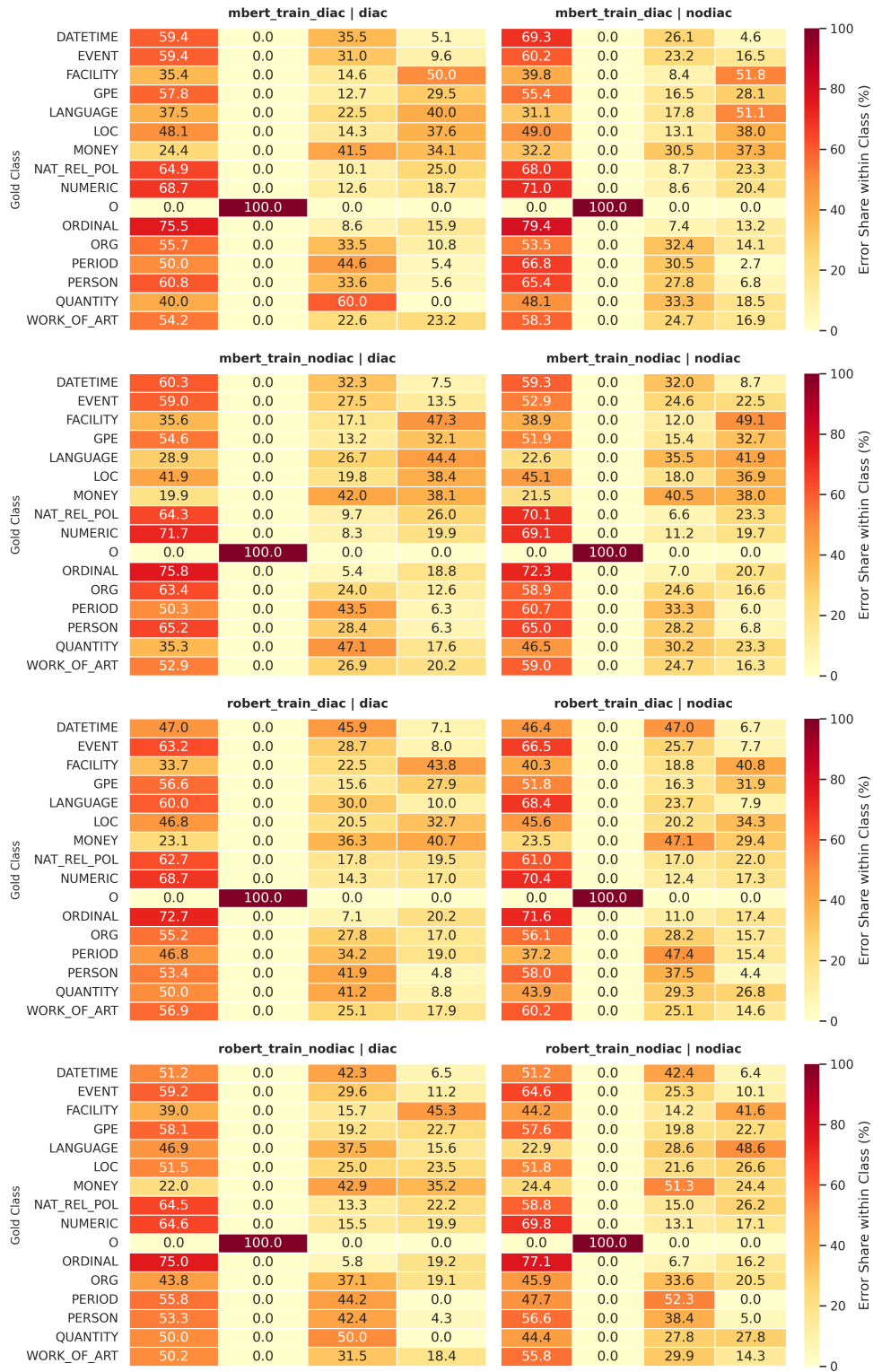


Figure 10: Error breakdown heatmap showing the relative distribution of error types (from left to right: *missed*, *spurious*, *wrong_boundary*, *wrong_type*) per entity class across fine-tuning and evaluation setups for two models. Each facet header specifies the model variant, training condition, and test setup (model_train_cond | eval_cond).

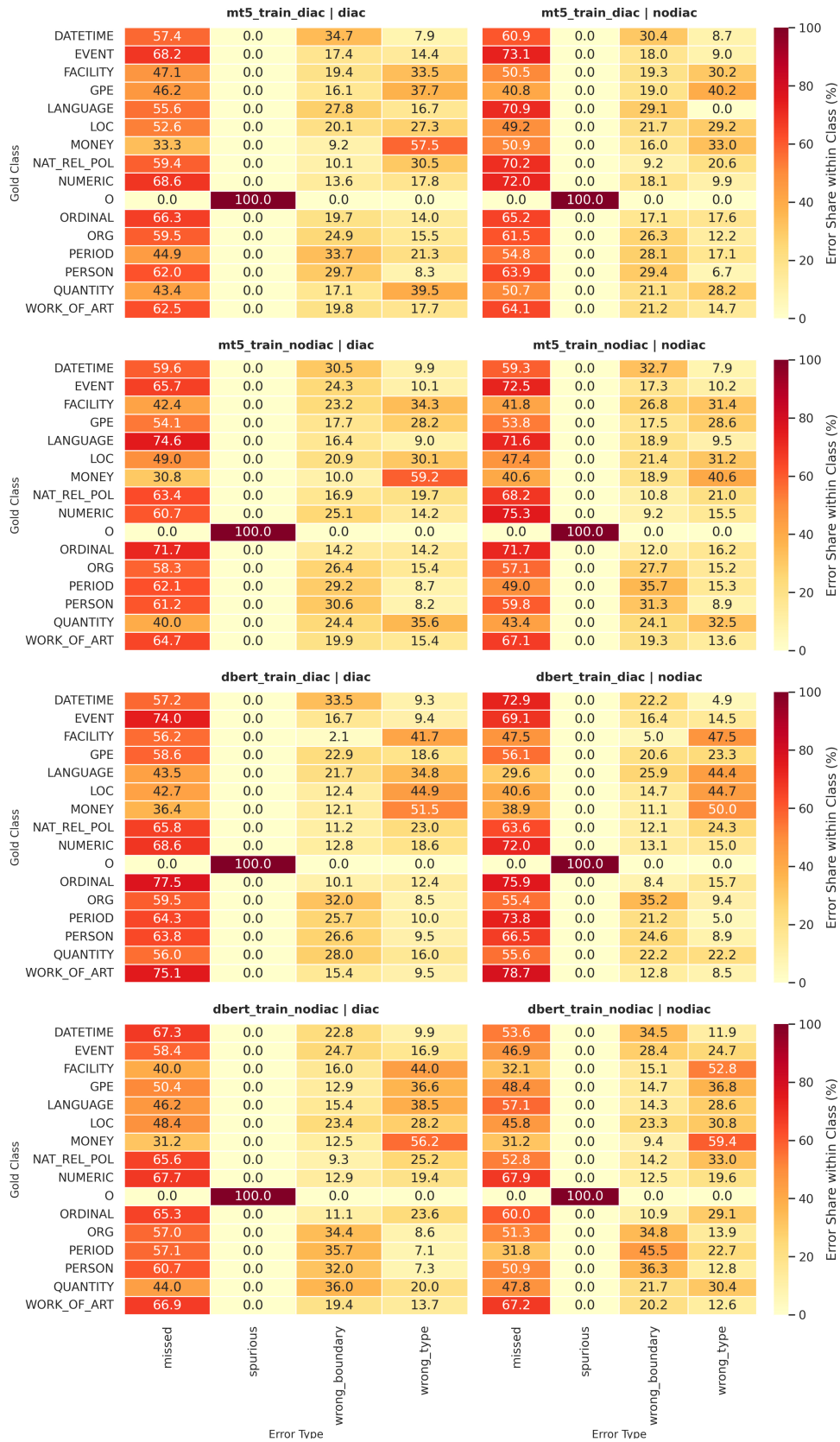


Figure 11: Error breakdown heatmap showing the relative distribution of error types (from left to right: *missed*, *spurious*, *wrong_boundary*, *wrong_type*) per entity class across fine-tuning and evaluation setups for two models. Each facet header specifies the model variant, training condition, and test setup (model_train_cond | eval_cond).