RILEC: Detection and Generation of L1 Russian Interference Errors in English Learner Texts

Darya Kharlamova and **Irina Proskurina**†
Higher School of Economics
Université Claude Bernard Lyon 1, Université Lumière Lyon 2, ERIC
dasha.kh18@gmail.com

Abstract

Many errors in student essays can be explained by influence from the native language (L1). L1 interference refers to errors influenced by a speaker's first language, such as using *stadion* instead of *stadium*, reflecting lexical transliteration from Russian. In this work, we address the task of detecting such errors in English essays written by Russian-speaking learners. We introduce RILEC, a large-scale dataset of over 18,000 sentences, combining expert-annotated data from REALEC with synthetic examples generated through rule-based and neural augmentation. We propose a framework for generating L1-motivated errors using generative language models optimized with PPO, prompt-based control, and rule-based patterns. Models fine-tuned on RILEC achieve strong performance, particularly on word-level interference types such as transliteration and tense semantics. We find that the proposed augmentation pipeline leads to a significant performance improvement, making it a potentially valuable tool for learners and teachers to more effectively identify and address such errors.

Keywords: L1 interference, data augmentation, grammatical error detection, learner corpora

1. Introduction

The influence of our native language (L1) on writing in a second language is manifold. It can lead to pronunciation-induced spelling errors, lexical choices based on the mother tongue, and tense or word order decisions that mirror L1 grammar (Odlin, 2003).

Some examples of native language interference in second-language acquisition include the absence or unconventional usage of certain grammatical categories, such as articles for Russian speakers and auxiliary verb constructions for German speakers, misused structures, such as confusion between *have* and *be* among French and Spanish speakers, and the incorrect use of *will* in conditionals, particularly by Russian and Spanish learners.

Features extracted from a learner's first language (L1) have been shown to enhance error detection and correction tools designed for language learners (Chang et al., 2008; Ng et al., 2014). For instance, native language information has been shown to influence the performance of grammatical error classifiers targeting errors commonly made by non-native English learners (Rozovskaya et al., 2017). It is also known to improve Large Language Model (LLM)-powered tools designed to support reading comprehension and writing skills (Antoniou-Kritikou et al., 2024; Heck and Meurers, 2023).

Although much research has been done on grammatical error detection (GED), simply identifying ungrammatical structures does not reveal the underlying causes of these errors. As a result, existing tools which detect and correct errors but do not explain their causes hinder the learning process more than help it (Brown, 2014). Information about the possible reasons behind the errors is also important for teachers to address specific difficulties, adjust study plans, create helpful learning materials, and score essays more efficiently. In addition, identifying L1-motivated errors can be beneficial for students at any proficiency level, as lower-level students might not recognize these errors independently, and even trained professionals sometimes struggle to detect them reliably.

In this paper, we investigate the problem of classifying potential causes of L1 interference. We use a linguistic interference tagging system shown in Table 1 to annotate our dataset, drawing on the re-

We will pay more attention to our health if we will have enough time. The straightforward approach is for people to pay attention on their nutrition and daily activity, and to refuse from using alcohol and tobacco. On one hand, most of parents want to shield the younger generation from another cureless diseases, including ads between TV shows or youtube's videos. On the other hand, we have firm and organizations producing unhealthy goods, and these firms want to expand their base of consumers through advertising.

Table 1: An excerpt from a real essay parsed by the model fine-tuned on RILEC: Synonyms, Copying Expression, Tense Semantics, Word Form Transmission, and Transliteration.

† Work done while at HSE University.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 10826–10837
11–16 May 2026.

search into interference done by Weinreich, 1979.

Our main contributions are the following: (1) We introduce **RILEC** (**R**ussian L1 **I**nterference **L**earner **E**nglish **C**orpus), the first large-scale dataset of L1-motivated errors, containing over 18,000 sentences, both real and synthetically augmented. [1] (2) We design a framework for L1 interference data augmentation, covering the generation of PPO-optimized GPT models, (§4.1), pattern-based rule generation (§4.2), and prompt-based controlled generation (§4.3). (3) Within this framework, we demonstrate and analyze capabilities and limitations of LLMs in erroneous sentences generation for data augmentation (§4.1,4.3). (4) We conduct a preliminary evaluation to assess the effect of each augmentation method on real data annotation (§5) and release the best-performing model for future uptake.

2. Related Work

L1-Interference Error Detection L1 interference is the influence of the speaker's native language on their L2 production, either via direct transfer, or structural and semantic shifts (Weinreich, 1979; Harris and Campbell, 1995). It is often shaped by linguistic similarity, leading to substitution or code-switching (Weinreich, 1979; Kučuk, 2023; Doğruöz et al., 2021). Recent works show that L1 features, such as identity, language family, and n-grams are useful in tasks like natural language inference (NLI) and grammatical error detection (Chang et al., 2008; Hermet and Désilets, 2009; Tetreault et al., 2013; Malmasi et al., 2017; Rozovskaya et al., 2017). For instance, Kochmar and Shutova (2016) demonstrate that L1 semantic information improves lexical error detection in verb-noun phrases. Zomer and Frankenberg-Garcia (2021) propose an L1-aware encoder-decoder model, outperforming GECToR and highlighting the need to integrate L1 features into developed grammatical error correction (GEC) systems.

Data Augmentation for GEC and GED The performance of GEC and GED models depends heavily on the size and quality of training data (Lichtarge et al., 2019, 2020; Nagata et al., 2022). Synthetic augmented data is often used to scale small training corpora and to mitigate error type imbalance (Stahlberg and Kumar, 2021; Wang et al., 2024; Lichtarge et al., 2020). Augmentation techniques are largely shared across GEC and GED, with tag-then-rewrite strategies proving particularly effective (Omelianchuk et al., 2020). These techniques are commonly categorized into rule-based methods (noise injection and pattern matching) and model-based approaches (conditional generation and translation)(Kiyono et al., 2020; Fang et al., 2023; Ye et al., 2023; Stahlberg and Kumar, 2021). Wang et al. (2024) further introduce a contextual augmentation method that uses a model to regenerate context around error patterns extracted from existing parallel corpora. Ye et al. (2023) propose a mixed MixEdit approach, which regulates the similarity and diversity of generated data for augmentation quality improvement.

To the best of our knowledge, existing work has not addressed L1-specific data augmentation for the L1-GED task. To bridge this gap, we build on prior research in data augmentation for GED, employing both rule-based and LLM-based methods to generate L1-influenced data for error detection. We report a comparison of English learner and GEC corpora in Table 2.

3. Corpus of L1-Interference Errors

Our corpus builds on the Russian Error-Annotated Learner of English Corpus (REALEC; Vinogradova and Lyashevskaya (2022)), which contains essays written by native Russian speakers in IELTS-like writing tasks, including descriptions of graphical data and opinion pieces, each under 300 words. The corpus is manually annotated for errors, specifically for the presence of interference errors.[2] Each error is represented by a span supplied with a corresponding correction. The subset already annotated for the presence of interference errors (henceforth REALEC-L1) is used for baselines and augmentation.

3.1. L1-Interference Annotation Scheme

We use an annotation scheme for the L1-interference error tagging system derived from the framework of language interference analysis proposed by Weinreich (1979). We describe the five-tag system in detail below.

Copying Expression This category includes word-for-word translations of L1 expressions and collocations. For instance, in (1b), the Russian expression (*kazhdovo iz nas*) is used as a substitute for an English linking phrase and is translated directly.

(1) a. A big bath was prepared for everyone.
b. * A big bath was prepared for **every of us**. (Bolshaya vanna byla prigotovlena dla **kazhdovo iz nas**.)

[1] https://github.com/harlamovads/RILEC

[2] A detailed description of the annotation scheme and associated tags can be found in the official manual released with the corpus: https://realec.org.

Dataset	L1 Background	Sents.	Error Tags	CEFR Level	Task
FCE (Yannakoudakis et al., 2018)	NA, Mixed L1s	2,695	71	B1-B2	GEC/GED
KJ (Nagata et al., 2011)	JA	3,199	22	A1-A2?	GEC/GED
CoNLL-2014 (Ng et al., 2014)	NA, Mixed L1s	1,312	28	C1	GEC/GED
JFLEG (Napoles et al., 2017)	NA, Mixed L1s	747	-	A1-C2?	GEC
BEA-2019 (Bryant et al., 2019)	NA, Mixed L1s (NUS students; EN, ZH, MS, TA)	4,384	25	A1-Native	GEC/GED
ICNALE (Ishikawa, 2023)	10 Asian L1s (ZH, ID, JA, KO, TH, ZH-TW, HK-YUE, PK-UR, PH-TL, TA)	15,000 essays	-	A2-C1	GEC
ICLEv3 (Granger et al., 2020)	26 L1 (BG, ZH, CS, NL, FI, FR, DE, IT, JA, NO, PL, RU, ES, SV, TR, TN, ...)	9,529 essays	-	B2-C2	GEC/NLI
TOEFL11 (Blanchard et al., 2013)	11 L1s (AR, ZH, FR, DE, HI, IT, JA, KO, ES, TE, TR)	12,100 essays	-	A2-C1	GEC/NLI
REALEC (Vinogradova and Lyashevskaya, 2022)	RU	18,710 essays	∼50	B1-C1	GEC/GED
RILEC	**RU**	**18,830**	**5**	**B1-B2**	**L1 GED**

Table 2: Comparison of English learner and GEC corpora by L1 background, dataset size, tag inventory, CEFR range, and task. Only RILEC includes explicit L1-specific interference tags. GEC = Grammatical Error Correction. GED = Grammatical Error Detection. NLI = Native Language Identification. L1 GED = L1-interference Error Detection. A question mark indicates unknown or approximate information.

Synonyms This category includes errors where the author intends to use a Russian word with multiple meanings corresponding to different English lexemes but selects the incorrect English word out of these corresponding words. For example, in (2), the student mistakenly uses *overcome* instead of *cover*. Both of these English words correspond to the same Russian lexeme as provided in the translation *preodolet'*.

(2) * The distance can be **overcame** by the train. (Distantsiya mozhet byt' **preodolena** poezdom.)

(3) and (4) provide examples of contexts that use the same word *preodolet'* in Russian, but correspond to two different words: *overcome* and *cover* in English. We categorize this error as Synonyms, because the synonymous English words are confused due to the influence of a Russian word that encompasses both meanings.

(3) They covered the necessary distance in one day. (Oni preodoleli nuzhnoe rasstoyanie za odin den'.)

(4) They overcame all the challenges and emerged victorious. (Oni preodoleli vse ispytanija i vyshli pobeditelami.)

Tense Semantics This type of error occurs when an English tense is used incorrectly due to the corresponding grammatical form in Russian. In (5b), we see that the present tense is used to describe graphical data relating to the past. While English requires the verb tense to align with the time frame of the event in such cases, in Russian, it is acceptable to use both past and present verb tenses for both historical present and graphical data description. This influences the learners and makes the confusion of past and present tenses in such contexts more frequent. Other errors of this type pertain to usage of *will* in conditionals, etc.

(5) a. In 1999 the share decreased.
b. * In 1999 the share **decreases**. (V 1985 jeta dola **snizhaetsa**'.)

Note that we do not address issues arising from English tenses not present in Russian (e.g., Present Perfect vs. Past Simple), as these involve the absence of features in L1, unlike the presence of Russian tense features causing problems here.

Transliteration This well-known type of error involves using Russian words written with the English alphabet in an English text. An example of such an error is shown in (6), where *cassa* is used instead of *cashier*.

(6) * And often a lot of money comes to **the cassa**. (Chasto mnogo deneg postupaet v **kassu**.)

Word Form Transmission This type of interference error involves transferring a grammatical category from Russian to English. In (7), the form *billions* is used erroneously most likely due to the equivalent Russian collocation which requires the plural form, with the plural marking carrying over into the English text.

(7) * The primary cost was $5 **billions**. (Osnovnaya stoimost' sostavljala 5 **milliardov** dollarov.)

The REALEC-L1 subset contains 6,086 sentences, each annotated as exhibiting at least one interference error in the main REALEC corpus. To assess annotation reliability, we conduct an additional round on 500 sentences from REALEC-L1, carried out by four Russian-speaking linguists with C1–C2 proficiency in English. All annotators are thoroughly familiarized with the guidelines. We first conduct a preliminary calibration on 100 sentences to ensure a consistent understanding of the criteria. The annotations are then conducted independently, without communication between annotators. Upon completion, we compute pairwise inter-annotator agreement (IAA) using Cohen's kappa (Cohen, 1960). The results, shown in Figure 1, range from 0.72 to 0.84, indicating a high level of inter-annotator consistency.

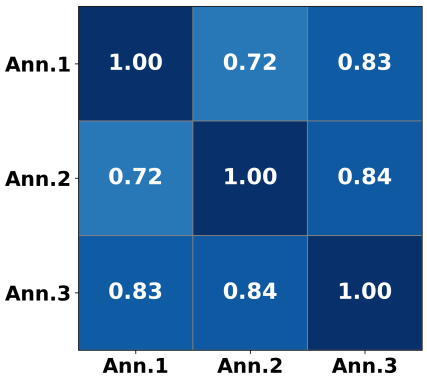

Figure 1: Pairwise inter-annotator agreement (Cohen's kappa) for the REALEC-L1 data annotation.

The full RILEC dataset includes this subset and is further expanded using data augmentation techniques. Detailed corpus statistics by error type are presented in §4.4 with generated examples in Table 4. We further describe how the synthetic data were generated and assess their contribution to classification performance. The numbers of sentences in the train and test splits from REALEC-L1 are reported in Table 3.

4. Data Generation

We use three approaches for data augmentation: (1) small-scale LLMs optimized with Proximal Policy Optimization (PPO) to generate sentences similar to annotated examples; (2) a rule-based algorithm that introduces controlled errors by replacing correct words with incorrect ones; (3) LLM prompting to generate erroneous sentences by modifying and replicating annotated interference errors. We detail each method in the following subsections.

Dataset	Train	Test
REALEC-L1	3,882	2,204
GPT-BASED (PPO)	4,583	1,965
LLM-BASED	556	239
RULE-BASED (GECTOR)	4,131	1,771
RILEC	12,652	6,179

Table 3: L1-error dataset splits by train and test size. The GPT-, LLM-, and rule-based datasets represent synthetic augmented subsets.

4.1. PPO-based Generation

In this section, we describe error generation with language models optimized using PPO (Schulman et al., 2017).

Model Set Up for PPO We experiment with two autoregressive models: GPT2[3] and DistilGPT2[4]. We first perform model fine tuning to adapt the models to in-domain generation using all sentences from REALEC. We observe that GPT2 is less affected by this step, as its next-word prediction loss decreases less than that of DistilGPT2, which performs better. We therefore continue with the latter model.

Reward Models To enable controlled generation of the five error types, we train a separate binary classifier for each error type, as described in §3. Each classifier is based on the RoBERTa-base model[5] fine tuned on the train subset of REALEC-L1. For each class, the dataset is divided into positive examples (containing the target error type) and negative examples (not containing it), with 80% used for training and 20% for evaluation. We fine tune each classifier for five epochs with a batch size of 16, a weight decay of 0.01, and a learning rate of 2×10^{-5}.

Reinforcement Optimization with PPO Next, we apply PPO to optimize the fine tuned DistilGPT2 for controlled error generation. Training is performed separately for each of the five error types, resulting in five optimized models. During training, the model generates sentences that are evaluated by the corresponding error specific classifier. The reward function assigns a positive score if the generated continuation contains the target error and

[3] hf.co/gpt2
[4] hf.co/distilgpt2
[5] hf.co/xlm-roberta

a negative score otherwise. PPO is performed for five epochs with an 80/20 data split on the entire REALEC dataset containing L1 annotated sentences, using a batch size of 32 and a learning rate of 2×10^{-5}. We find that DistilGPT2 classifier based reward scores increase during training, indicating successful alignment with the intended error generation. We further analyze data generated by the five models trained for each L1 tag. Sample continuations generated from short input segments are presented in Table 4.

Evaluation and Expert Feedback We manually analyze examples generated by the PPO optimized language models for each error type. To assess generation quality, we qualitatively annotate 50 randomly selected outputs per error type, marking the presence of the respective L1 influenced error. Based on these annotations, we find that the generated samples correctly represent the target errors in all cases except for two categories. For *Tense Semantics* and *Transliteration*, the trained models fail to reliably generate errors. The former model does not produce suitable context for the erroneous verb form, while the latter generates random character sequences resembling transliteration errors (see Example 8).

(8) Also, in the Middle East wich beaute a dramatost of hi teck and the femenest ode to alhtejut's exalt is so difunct.

As a result, we exclude PPO-based generations for these two categories from the final dataset. For these categories, we only apply a rule-based generation strategy (see §4.1). To ensure natural sentence openings, we extract the most frequent sentence-initial words from the full REALEC corpus and use them as weighted prompts during generation. We select prompts for generation from these words using random sampling weighted by their frequency in REALEC. This ensures that the initial word distribution closely matches the original dataset, avoiding over-representation of specific words that could bias later training. In total, we obtain 6,547 examples by generating continuations for the three error tags based on query conditions, after filtering out sentences shorter than five tokens.

Rule-based Error Injection We use a rule-based method to inject errors related to *Tense Semantics* and *Transliteration*. First, we use the fine tuned GPT2 (see §4.1) to obtain domain specific but mostly grammatically correct sentences (using the same probability based prompting strategy).

For the *Tense Semantics* tag, we filter sentences containing a year and modify the verb in the corresponding clause to the Present Simple form using the SpaCy package (Honnibal et al., 2020). As the generated years are typically associated with past or future events, this verb change results in an erroneous interpretation. We generate this type of error because it reflects one of the most frequent error patterns found in the non synthetic REALEC-L1 data, where past and future events are incorrectly described using the present tense when interpreting graphical data (Vinogradova and Lyashevskaya, 2022).

For the *Transliteration* tag, we replace randomly selected nouns with their transliterated versions obtained using the Google Translate API (accessed in July 2024).[6]

Examples of both error types are presented in Table 4. In total, we generate 1,748 augmented sentences for *Tense Semantics* and 895 for *Transliteration* using the described rule-based strategy.

4.2. Error Generation with GECToR

We adopt a rule-based approach following the one used for data augmentation for GECToR (Omelianchuk et al., 2020) training. Error generation used by GECToR authors focuses on introducing errors of three types: missing words, redundant words, and words that need to be replaced. We focus solely on implementing the replacement errors, as they are the most relevant for interference errors.

Data Preparation First, we collect possible corrections for the erroneous spans. We use the full REALEC-L1 with native span corrections. To extend possible corrections, we utilize the XLM-RoBERTa model.[7] We mask the error spans in the base sentences and treat the suggested filled masks as corrections for the respective sentences. After that, we pair the corrections and the error spans to create a dictionary of possible errors: each correction is mapped to a list of tokens that could erroneously replace it. For each error type, we create its own corrections dictionary.

Error Injection We adjust GECToR's official implementation so that, for every sentence, a single word is chosen and replaced in accordance with the dictionaries. After the error is introduced, the code marks it and specifies its type based on the collection that was sourced to introduce the error. We run the code on grammatically correct generations from the GPT2-based generator after fine-tuning on REALEC-L1, as described in §4.1, since it was found to produce thematically suitable and mostly grammatically correct sentences. To classify errors into five types, we use a baseline classifier

[6] cloud.google.com/translate
[7] hf.co/xlm-roberta

Error Type	Generation Examples
CopExp	Overall all these places are **in our age**. (PPO) This is extremely unlikely for everyone and every scientist and medical worker in **influence on** world. (GECToR) Males in **66-75 ages** made a slight jump compared to 56-65 years old males. (LLM)
Synonyms	If such a person should enter this prison, they should be **given** what they had already done. (PPO) In the one hand, people should make their living, and they should **take** the necessary education. (PPO) This number is 1.5 times higher than the previous ones during the **departed** period. (GECToR)
WFT	Also, the global total increase has fallen, and nearly 2 billion people, making it number **1 billion billions.** (PPO) This will be **girls'** longest run we will see in nearly every post on this web. (GECToR)
TenSem	It can be seen that in Africa the rate of unemployment **increases** at the same rate as in 2000, and then in Africa it **remains** at a steady rate. (Rule-Based) There **is** no changes in this part of the population in the period in which 2000 was presented. (GECToR) The proportion of old individuals aged 66 and over **decrease** in periods 1950-1995 from 6% to 4% and start growing after that. (LLM)
Transliteration	The percent of **investitsment** came in Russia, where in 2015 it was 5 billion Euros and in 2018 it is 8 billion Euros. (Rule-Based) She is receiving lots of **funat** emails. (LLM) This can be deduced from the fact that the highest rate of unemployment among South Asia **fabricks** Latin America in 2014 and 2015 was observed. (GECToR)

Table 4: Examples of ungrammatical model generations from RILEC, grouped by L1 error type. Data augmentation methods are indicated in brackets: PPO (Proximal Policy Optimization with DistilGPT2), GECToR (dictionary-based), Rule-based (tense replacement or transliteration), and LLM (prompt-based generation). CopExp = Copying Expression; WFT = Word Form Transmission; TenSem = Tense Semantics.

fine-tuned on REALEC-L1, using the same experimental settings as in §5. As a result, we obtain 5,900 augmented examples through rule-based error injection. Examples of the generated data are provided in Table 4.

4.3. Prompted Generation of Interference Errors

To generate new examples for each interference tag, we prompt decoder-only LLMs with class details for each interference type with examples. After generating data with the best performing model, we perform error-annotation using LLMs.

Data Generation We begin by selecting the best model for generation through annotation among LLMs designed for open-ended conversations. The three models we select for experiments are accessible via API[8] include: Claude 2,[9] GPT-3.5,[10] and Mistral (Jiang et al., 2023).[11] We prompt the models to generate new examples using the following input, concatenated with 10 randomly selected sentences from the corpus manually annotated for L1 interference class errors:

> Here are some sentences with L1-motivated mistakes. Find the mistakes in these sentences and generate new contexts with different meanings, while retaining the mistakes from the original sentences.

For generation, we set the temperature to 1.0 and use the default repetition penalty for each model. We paraphrase the prompt four times and repeat generation for each version. Overall, we obtain 40 erroneous sentences for each of the models.

Next, we ask an expert specializing in error annotation to review the generated model outputs. The expert marks a generation as successful if it contains an error of the same type as in the source sentence and as unsuccessful if the present error differs from the one in the provided sentence. From the annotation results, we find that the Claude 2 model outperforms the rest, with 38 out of 40 successful generations. Mistral successfully generates examples in half of the cases, whereas GPT-3.5 does not follow the instructions and fails to generate erroneous sentences at all.

We select Claude 2 for data generation and generate 800 examples, each time using 10 newly selected samples from the source corpus. The generation process takes a few hours, with necessary pauses due to per-hour generation limits. We then proceed with the annotation of the generated examples.

[8] Accessed in July 2024.
[9] anthropic.com/introducing-claude
[10] https://chat.openai.com
[11] https://mistral.ai

Evaluation Given the generated examples, we repeat the procedure, but this time for annotating the examples. We select the model optimal for annotation with assistance of the expert. First, we prompt the models with annotation instructions to identify and label errors of a given type. For annotating examples, we follow the instructions from the source REALEC-L1 dataset, originally designed for expert annotators and construct the following prompt:

> Here are sentences that contain mistakes. Some mistakes are caused by interference with the Russian language. Find and highlight such mistakes. Classify the mistakes according to the *Instructions*. The following sentences are provided as examples.

We annotate 40 randomly selected sentences from the source data using this prompt for three models and ask the linguistic expert to review the annotated outputs. The expert marks their agreement with the error span and the type of L1 interference specified by the model, both of which we consider together as an accurate annotation. We find that the Mistral model outperforms the rest, with 40 out of 40 correct annotations. Meanwhile, Claude and GPT-3.5 correctly annotate fewer than 10 cases. We therefore select Mistral model for data annotation. Using this model, we annotate 800 examples generated by Claude-2 previously selected for data generation.

As a result, we obtain 794 annotated examples out of 800 previously generated erroneous sentences. Six sentences were excluded at this stage due to very closely resembling other LLM-generated sentences.

4.4. Corpus Overview

Figure 2 shows the distribution of errors by type and generation method in RILEC. With the described data augmentation, we obtain approximately 4,000 erroneous texts for each L1 interference type, resulting in a total of 18,830 sentences. This dataset is subsequently used to train an L1 error classification model. In addition, we evaluate the diversity of the synthetic data relative to the original data using Self-BLEU, MAUVE, and 3-gram novelty, as reported in Appendix B.

5. L1-interference Error Classification

In this section, we report results of models trained with different augmented datasets and analyze the performance of the resulting L1 interference error detection models.

5.1. Experimental Settings

We approach L1-interference error classification as a multi-span classification task and implement

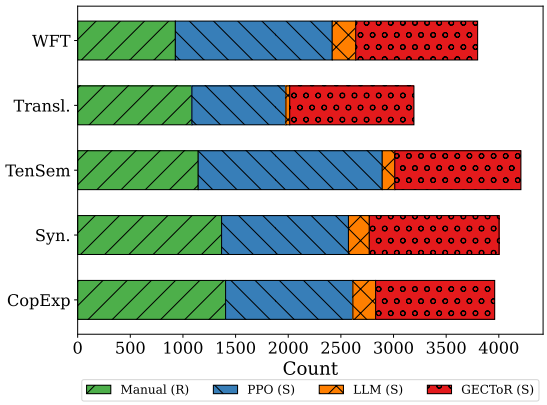

Figure 2: Error distribution in RILEC, including synthetic (S) and real (R) data. CopExp = Copying Expression; WFT = Word Form Transmission; TenSem = Tense Semantics.

the solution using the SpaCy span classification pipeline.[12] We perform experiments using the RoBERTa-base model[13]. Each model is fine tuned with a batch size of 128, a span classification threshold of 0.5, and trained for ten epochs.

Data For each augmentation method, we shuffle and split the data, using 80% for training and 20% for testing. The dataset split statistics are reported in Table 3. To estimate the contribution of augmented data to REALEC-L1 classification, we evaluate all models on the REALEC-L1 test set. We also report the F1-score on the in-domain data used for fine tuning. As a baseline, we use the model trained on the REALEC-L1 dataset described in §3.

5.2. Results

Table 5 reports the error detection performance for each L1-interference category. The classifier trained on RILEC, which includes all augmented data, substantially outperforms the baseline and models fine-tuned on individual augmentation subsets. The highest F1-scores are observed for *Transliteration* and *Word Form Transmission*, both exceeding 90%, with an overall average of approximately 74%, compared to 55.66% for the baseline.

Effect of Augmented Data Models fine-tuned on data generated through model-based augmentation methods outperform those trained on rule-based data, despite the larger size of the latter. The PPO-optimized DistilGPT model achieves the second-highest average F1-score (71.81%),

[12] https://v2.spacy.io/usage
[13] hf.co/FacebookAI/roberta-base

Dataset	$F1_{train}$	$F1_{test}$ Tag-specific					
		CopExp	Syn	TenSem	Transl.	WFT	Avg.
Base	55.66	15.79	46.60	75.44	83.33	57.14	55.66
GPT-Based	**83.76**	21.62	**79.25**	**82.14**	92.31	83.72	71.81
Rule-Based	78.53	31.82	61.40	77.06	84.62	75.00	65.98
LLM-Based	44.78	**41.03**	72.16	78.85	92.31	71.43	71.16
RILEC	83.00	33.33	69.39	80.00	**96.55**	**90.48**	**73.95**

Table 5: L1-interference error classification results for RoBERTa models fine-tuned on augmented subsets: augmented with DistilGPT optimized with PPO (GPT-based; §4.1), with prompting LLMs (LLM-based; §4.3), and obtained via rule-based generation (§4.2). We report average F1-scores on the training dataset and tag-specific scores. CopExp = Copying Expression; WFT = Word Form Transmission; TenSem = Tense Semantics. The best score is in bold, and the second-best score is underlined.

followed closely by the LLM-prompted model, despite using significantly fewer training examples. The LLM-based model also exhibits better generalization, as indicated by lower training and higher test performance.

Class-Specific Performance The model fine-tuned on the full RILEC dataset achieves high F1-scores (above 80%) on the *Tense Semantics*, *Transliteration*, and *Word Form Transmission* classes. In contrast, performance is lower on *Copying Expression* (33.33%) and *Synonyms* (69.39%), which require more complex handling of L1-specific lexical and collocational interference. For these categories, the best-performing models are those fine-tuned on LLM-prompted data and PPO-optimized DistilGPT, suggesting they are more effective at modeling such interference patterns.

Manual Analysis We conduct a manual evaluation with a linguist on two datasets: 100 test sentences ($Test_{100}$) and 70 randomly sampled sentences from the REALEC learner corpus (REALEC$_{70}$). This allows us to compare model performance on benchmark and real learner data. For $Test_{100}$, we report accuracy based on both gold annotations and expert-approved alternatives, including *Distinct TPs*—correct predictions unique to each model. For REALEC$_{70}$, we report true positives and TP–FP (net correct) counts. Table 6 shows that all models trained on augmented data outperform the baseline, with the highest scores (73–74%) achieved by models trained on RILEC and PPO-optimized GPT outputs.

The linguist notes that the rule-based model performs particularly well on tags such as *Synonyms* (9) and *Copying Expression* (10), which show the lowest overall scores in Table 6.

(9) ...the population of Sweden **outlived a considerable growth** (**perezhila znachitel'nyi rost**)

Model	Test$_{100}$		REALEC$_{70}$	
	Acc.	Dist.	TP	TP–FP
Base	0.66	0	6	4
GPT-based	0.74	14	15	7
Rule-based	0.74	10	17	13
LLM-based	0.68	8	8	5
RILEC	0.73	10	19	10

Table 6: Manual evaluation. Test$_{100}$: Acc. = accuracy, Dist. = distinct TPs. REALEC$_{70}$: TP = true positives, TP–FP = net correct predictions.

(10) ... we can not only **safe** our time, but also it allows us to **achieve to** our destination. (**sokhranit'**, **dostignut'**)

The model fine-tuned on optimized DistilGPT data achieves the highest number of distinct correct predictions (14), capturing valid errors missed by other models.

On REALEC$_{70}$, model performance follows a similar trend: RILEC-trained model achieves the highest true positive count (19), followed by the pattern-generated model (17), both outperforming the baseline (6). The same models also lead in terms of the TP–FP balance. The model fine-tuned on prompting data exceeds the baseline by a smaller margin, possibly due to sensitivity to domain mismatch.

Overall, our results demonstrate that the model fine-tuned on RILEC, which includes all augmented data, outperforms the baseline by a large margin in classification scores averaged across all L1 interference error classes. It achieves particularly high F1-scores on Word Form Transmission, Transliteration, and Tense Semantics errors. Manual analysis further confirms that the RoBERTa model fine-tuned on RILEC successfully detects errors in new learner texts, validating the effectiveness of the designed span annotation model in practice.

6. Conclusion

In this work, we introduce RILEC, a large-scale corpus for detecting L1 interference errors in English essays written by Russian-speaking learners. The corpus combines manually annotated data from REALEC with synthetically generated data created using rule-based and language model-based augmentation methods.

To assess the impact of data augmentation, we trained models on different corpus subsets and observe consistent performance gains, especially when using PPO optimization. Manual evaluation confirmed that augmented data better capture the semantics, style, and grammar of real learner language. Models trained on RILEC performed best on word-level interference types such as transliteration, tense semantics, and grammatical form mismatches, while showing lower accuracy on collocational and semantic categories like synonym substitutions and copying expressions.

In future work, we plan to extend RILEC with essays written by learners from diverse L1 backgrounds, using resources such as ICNALE and native language identification datasets (Hermet and Désilets, 2009; Ishikawa, 2018). We also plan to extend the evaluation to generative models. Preliminary experiments with GPT-5 and Claude indicate that these models exhibit low precision in L1 error detection on the RILEC data.

Limitations

This study focuses on the detection of L1-motivated errors using a fixed five-tag annotation scheme. While this allows for systematic evaluation, it limits the granularity of linguistic phenomena captured. Future work could extend this by analyzing part-of-speech distributions and syntactic dependency relations, which may help uncover deeper patterns of L1 interference and improve classification performance.

We also find that GPT2, even when fine-tuned and trained with PPO, fails to reliably generate certain error types, particularly Tense Semantics and Transliteration errors. This suggests that GPT2 resists producing ungrammatical or unnatural constructions in these categories. To address this, we implement rule-based augmentation strategies. Although effective, this introduces an external dependency and limits possible generation quality and diversity. Future work could explore alternative model compression techniques, such as quantization or pruning, to better adapt small models like GPT2 for controlled error generation.

Finally, our dataset is restricted to IELTS-style English essays written by Russian-speaking learners. While the proposed framework is applicable to other L1 groups and genres, we do not evaluate cross-linguistic generalizability in this work.

Ethical Considerations

This study uses anonymized, publicly available data from the REALEC corpus, in accordance with its educational and research license. No personal or sensitive information is included. Our released models support language learning by identifying L1-specific error patterns, not for grading or assessment. We emphasize that automated error detection should be used with human oversight. Furthermore, model-based data augmentation methods could be misused, for instance, to generate fake learner data, simulate interference patterns, or fabricate responses on learning platforms. While such risks are limited, we highlight responsible use and restrict our models to educational and research purposes. Future work may expand the dataset to cover more L1 backgrounds to improve generalizability and support learners from diverse linguistic backgrounds.

7. Bibliographical References

Ioanna Antoniou-Kritikou, Voula Giouli, George Tsoulouhas, and Constandina Economou. 2024. Using llms in a language teaching and learning application. *ERCIM News*, 2024(136).

H Douglas Brown. 2014. *Principles of language learning and teaching: A course in second language acquisition*. Pearson.

Yu-Chia Chang, Jason S Chang, Hao-Jan Chen, and Hsien-Chin Liou. 2008. An automatic collocation writing assistant for taiwanese efl learners: A case of corpus-based nlp technology. *Computer Assisted Language Learning*, 21(3):283–299.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. 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 1654–1666, Online. Association for Computational Linguistics.

Tao Fang, Xuebo Liu, Derek F. Wong, Runzhe Zhan, Liang Ding, Lidia S. Chao, Dacheng Tao, and Min

Zhang. 2023. TransGEC: Improving grammatical error correction with translationese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3614–3633, Toronto, Canada. Association for Computational Linguistics.

Alice C Harris and Lyle Campbell. 1995. *Historical syntax in cross-linguistic perspective*, volume 74. Cambridge University Press.

Tanja Heck and Detmar Meurers. 2023. On the relevance and learner dependence of co-text complexity for exercise difficulty. In *Swedish Language Technology Conference and NLP4CALL*, pages 71–84.

Matthieu Hermet and Alain Désilets. 2009. Using first and second language models to correct preposition errors in second language authoring. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–72, Boulder, Colorado. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Shin'ichiro Ishikawa. 2018. Icnale: the international corpus network of asian learners of english. *Icnale: the international corpus network of asian learners of english*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Shun Kiyono, Jun Suzuki, Tomoya Mizumoto, and Kentaro Inui. 2020. Massive exploration of pseudo data for grammatical error correction. *IEEE/ACM transactions on audio, speech, and language processing*, 28:2134–2145.

Ekaterina Kochmar and Ekaterina Shutova. 2016. Cross-lingual lexico-semantic transfer in language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 974–983, Berlin, Germany. Association for Computational Linguistics.

Mojca Stritar Kučuk. 2023. Error annotation in slovene learner corpus kost-why l1 students can (not) do the job. In *CLARC 2023–Language and Language Data International Scientific Conference, Rijeka, Croatia, 28-30 September*.

Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data weighted training strategies for grammatical error correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. 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 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A report on the 2017 native language identification shared task. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–75, Copenhagen, Denmark. Association for Computational Linguistics.

Ryo Nagata, Manabu Kimura, and Kazuaki Hanawa. 2022. Exploring the capacity of a large-scale masked language model to recognize grammatical errors. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4107–4118, Dublin, Ireland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the eighteenth conference on computational natural language learning: shared task*, pages 1–14.

Terence Odlin. 2003. Cross-linguistic influence. *The handbook of second language acquisition*, pages 436–486.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Alla Rozovskaya, Dan Roth, and Mark Sammons. 2017. Adapting to learner errors with minimal supervision. *Computational Linguistics*, 43(4):723–760.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A report on the first native language identification shared task. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta, Georgia. Association for Computational Linguistics.

Yixuan Wang, Baoxin Wang, Yijun Liu, Qingfu Zhu, Dayong Wu, and Wanxiang Che. 2024. Improving grammatical error correction via contextual data augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10898–10910, Bangkok, Thailand. Association for Computational Linguistics.

Uriel Weinreich. 1979. Languages in contact: Findings and problems.

Jingheng Ye, Yinghui Li, Yangning Li, and Hai-Tao Zheng. 2023. MixEdit: Revisiting data augmentation and beyond for grammatical error correction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10161–10175, Singapore. Association for Computational Linguistics.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *International conference on machine learning*, pages 4006–4015. PMLR.

Gustavo Zomer and Ana Frankenberg-Garcia. 2021. Beyond grammatical error correction: Improving L1-influenced research writing in English using pre-trained encoder-decoder models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2534–2540, Punta Cana, Dominican Republic. Association for Computational Linguistics.

8. Language Resource References

Blanchard, Daniel and Tetreault, Joel and Higgins, Derrick and Cahill, Aoife and Chodorow, Martin. 2013. *TOEFL11: A corpus of non-native English*. Wiley Online Library.

Bryant, Christopher and Felice, Mariano and Andersen, Øistein E. and Briscoe, Ted. 2019. *The BEA-2019 Shared Task on Grammatical Error Correction*. Association for Computational Linguistics.

Granger, Sylviane and Dupont, Maïté and Meunier, Fanny and Naets, Hubert and Paquot, Magali. 2020. *The International Corpus of Learner English. Version 3*. Presses universitaires de Louvain.

Ishikawa, Shin'ichiro. 2023. *The ICNALE guide: An introduction to a learner corpus study on Asian learners' L2 English*. Routledge.

Nagata, Ryo and Whittaker, Edward and Sheinman, Vera. 2011. *Creating a manually error-tagged and shallow-parsed learner corpus*.

Napoles, Courtney and Sakaguchi, Keisuke and Tetreault, Joel. 2017. *JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction*. Association for Computational Linguistics.

Ng, Hwee Tou and Wu, Siew Mei and Briscoe, Ted and Hadiwinoto, Christian and Susanto, Raymond Hendy and Bryant, Christopher. 2014. *The CoNLL-2014 Shared Task on Grammatical Error Correction*. Association for Computational Linguistics.

Vinogradova, Olga and Lyashevskaya, Olga. 2022. *Review of practices of collecting and annotating texts in the learner corpus REALEC*. Springer.

Yannakoudakis, Helen and Andersen, Øistein E and Geranpayeh, Ardeshir and Briscoe, Ted and Nicholls, Diane. 2018. *Developing an automated writing placement system for ESL learners*. Taylor & Francis.

A. Experimental Settings

All training and fine-tuning experiments were conducted on a single NVIDIA A100 GPU with 80 GB of GPU memory.

Model Fine-tuning We fine-tune GPT2 and DistilGPT2 using an 80/20 train–test split for 3 epochs with a batch size of 32, a learning rate of 2×10^{-5}, and weight decay of 0.01.

PPO Optimization and Reward Models PPO optimization and reward model training are performed for 5 epochs with a learning rate of 1.41×10^{-5} and a batch size of 16. Sampling uses a temperature of 0.7 with top-$k = 0$ and top-$p = 1.0$.

B. Lexical Diversity

To ensure that the augmentation pipelines enrich the source REALEC-L1 data distribution, we evaluate the diversity of data generated by the rule-based, LLM-based, and GPT-based pipelines. We evaluate Self-BLEU (Zhang et al., 2017), MAUVE (Pillutla et al., 2021), and 3-gram novelty on the downsampled sentences relative to the REALEC-L1 test set of 2,204 sentences. Table 7 reports the evaluation results for the generated data. We find that the rule-based pipeline achieves the highest 3-gram novelty (0.92), indicating strong lexical novelty relative to the source distribution. The LLM-based pipeline yields the highest MAUVE score (0.97), suggesting the closest distributional alignment with REALEC-L1, while also exhibiting the lowest Self-BLEU (12.24), reflecting greater internal diversity. The GPT-based pipeline produces moderate scores across all metrics. Overall, these results demonstrate that the proposed augmentation pipelines successfully generate diverse, non-redundant error patterns that enrich the original data distribution.

Pipeline	Self-BLEU ↓	MAUVE ↑	3-gram Novelty ↑
GPT-based	15.25	0.47	0.70
LLM-based	12.24	0.97	0.20
Rule-based	12.75	0.36	0.92
All Augmented	14.03	0.48	0.92
REALEC-L1	16.37	1.00	0.00

Table 7: Diversity and distributional similarity of augmented datasets relative to REALEC-L1. Lower Self-BLEU indicates greater diversity, while higher MAUVE and 3-gram Novelty indicate closer distributional alignment and higher n-gram novelty.