Distinguishing Grammatical Errors from Code-Switching in Multilingual Learner Texts

Helmi Baazaoui[1,3][a], Lilia Cheniti Belcadhi[2][b], David Laiymani[3][c] and Christophe Guyeux[3][d]
[1]*Faculty of Science of Monastir, Tunisia*
[2]*ISITC Hammam Sousse, Tunisia*
[3]*Université Marie et Louis Pasteur, France*
helmi.baazaoui@univ-fcomte.fr, liliachenitibelcadhi@gmail.com, david.laiymani@univ-fcomte.fr, guyeux@gmail.com

Keywords: Code-Switching, Grammatical Error Correction, Curriculum Learning, Multilingual NLP, Intelligent Tutoring Systems.

Abstract: Grammatical Error Correction (GEC) is essential in educational NLP but struggles with learner writing containing code-switching (CSW) mixing languages within a sentence. Traditional monolingual GEC models often mislabel valid multilingual spans as errors and lack pedagogical feedback. We propose a novel three-stage curriculum learning framework that distinguishes grammatical errors from code-switches at the token level. Starting with synthetic CSW-GEC data generated via linguistically informed span injection, we create two human-annotated English–French and English–Arabic benchmarks. Fine-tuning multilingual Transformers (XLM-R, mDeBERTa) with staged supervision improves fluency, error sensitivity, and robustness. Results show curriculum models outperform baselines, with mDeBERTa achieving top $F_{0.5}$ scores. Error analysis highlights challenges at GE–CSW boundaries, guiding future dataset expansion and finer error classification. This work supports Intelligent Tutoring Systems that correct without penalizing code-switching, enabling inclusive language learning.

1 INTRODUCTION

Grammatical Error Correction (GEC) plays a central role in many NLP applications, ranging from writing assistance to Intelligent Tutoring Systems (ITS), by identifying and rectifying learner mistakes. State-of-the-art monolingual GEC models, such as GECToR (Omelianchuk et al., 2020), formulate error correction as a sequence-tagging task and attain high accuracy with minimal latency.

However, these systems typically assume that input is written in a single language, rendering them ineffective for code-switched (CSW) learner data where grammatical errors may be masked by or conflated with valid language alternations. In language learning contexts, learners frequently resort to code-switching or translanguaging to express confusion, emotion, or lexical gaps (Cenoz and Gorter, 2021; Liu et al., 2024a). Without distinguishing genuine grammatical errors from strategic language switching, ITS lose the ability to accurately track learner mistakes or diagnose underlying issues such as syntactic weaknesses, vocabulary limitations, or affective needs (Nguyen et al., 2022; Ramadhan et al., 2024).

Consider the sentence:

I didn't understood le concept de l'énergie.

In this example, *understood* constitutes a grammatical error (incorrect verb form following "did"), while *le concept de l'énergie* represents a valid code-switch to French, likely due to a vocabulary gap. A traditional monolingual grammar correction system may incorrectly attempt to translate or remove the French span while failing to identify the actual grammatical error. This misrepresents both the learner's intent and their underlying difficulty.

In contrast, a system capable of distinguishing grammatical errors from intentional code-switching can identify both the learner's struggle with English verb morphology and their lack of vocabulary to express the concept in English. Such a distinction enables more accurate and personalized feedback, for example by offering grammar correction alongside

[a] https://orcid.org/0009-0006-5257-0082
[b] https://orcid.org/0000-0001-8142-6457
[c] https://orcid.org/0000-0003-2580-6660
[d] https://orcid.org/0000-0003-0195-4378

targeted vocabulary support, and provides ITS with a clearer profile of the learner's specific challenges.

Research on code-switching has largely focused on token-level language identification using conditional random fields and character-level features (Solorio et al., 2014; Molina et al., 2016), and more recently on neural sentence-level classifiers (Burchell et al., 2024). In educational contexts, (Nguyen et al., 2022) survey existing ITS and document how current systems fail to handle pedagogical code-switching, where language switches serve to express emotion, simplify explanations, or scaffold meaning (Cenoz and Gorter, 2021; Sonkar et al., 2023; Ramadhan et al., 2024). Although code-switching offers pedagogical benefits such as improved confidence, participation, and comprehension (Cenoz and Gorter, 2021; Ramadhan et al., 2024), educational NLP tools remain ill-equipped to process mixed-language input due to limited training data and evaluation benchmarks (Potter and Yuan, 2024; Chan et al., 2024; Pérez-Ortiz et al.,).

At the intersection of GEC and CSW detection, (Chan et al., 2024) propose injecting foreign-language spans into GEC corpora to simulate code-switching behavior, while (Potter and Yuan, 2024) leverage large language models to generate mixed-language erroneous sentences. (Liu et al., 2024b) extend these ideas using pedagogy-inspired strategies for structured code-switching. Despite their effectiveness, these approaches conflate grammatical error correction and language demarcation into a single tagging objective.

In this work, we propose a three-stage curriculum learning framework that explicitly distinguishes grammatical errors from code-switches at the token level, thereby enabling ITS to better identify learner challenges related to grammar, vocabulary, and expressive intent. Specifically, we: (1) generate synthetic CSW-GEC data through linguistically motivated foreign-span injection (Pratapa et al., 2018; Chan et al., 2024); (2) construct two human-annotated benchmarks for English–French and English–Arabic with high inter-annotator agreement ($\kappa = 0.88$); and (3) fine-tune multilingual Transformer models using staged training that progresses from fluent English to monolingual correction and finally to joint GE/CSW tagging via curriculum learning (Xu et al., 2020; Guan et al., 2025).

While this framework has not yet been deployed in a fully operational tutoring system, it establishes a foundational ITS component capable of diagnosing learner language development. Our current contribution focuses on validating a token-level disambiguation module that separates grammatical errors from code-switches, a critical prerequisite for future ITS integration (Pérez-Ortiz et al., ; Ramadhan et al., 2024; Liu et al., 2024b).

2 RELATED WORK

2.1 Monolingual Grammatical Error Correction (GEC)

Modern monolingual GEC systems often use sequence-tagging models rather than full generative decoders. For example, GECToR (Omelianchuk et al., 2020) is a Transformer-based tagger that predicts token-level edit operations to correct errors. This "tag, not rewrite" approach yields high accuracy and is much faster than traditional encoder-decoder models. A related challenge in monolingual GEC is data scarcity, so many works generate synthetic errorful data. Common methods include noise injection (randomly inserting, deleting or substituting characters/words) and other rule-based corruptions. For instance, early GEC methods randomly apply error edits (Rozovskaya and Roth, 2010; Felice and Yuan, 2014; Xu et al., 2019). Other approaches use back-translation: an MT model is trained to translate correct sentences back into an ungrammatical (errorful) form (Rei et al., 2017; Yuan et al., 2019; Stahlberg and Kumar, 2021). A related technique is round-trip translation (translating to another language and back), which also injects naturalistic errors. All these synthetic-data strategies have been shown to improve GEC performance when combined with real error-corrected corpora.

2.2 Code-Switching Detection & Generation

CSW Detection. Detecting code-switching in text typically relies on language identification (LID) at the token level. In many social-media and dialog settings, systems first tag each word with a language (using CRFs or neural taggers) and then infer switch points. For example, shared tasks on code-switched language ID (e.g. (Solorio et al., 2014); (Molina et al., 2016)) trained token-level classifiers using character n-grams, word features and CRFs or LSTMs. (Ramanarayanan and Pugh, 2018) and others showed that rich character- and context-based features yield high accuracy for tweet-style code-switch detection. More recently, (Burchell et al., 2024) reformulate CSW detection as a sentence-level multi-label LID problem: their models label each sentence with the set of lan-

guages it contains rather than tagging every word. In learner writing, a similar strategy is to use LID to filter or flag words not in the target language, then mark code-switched spans as needed, but this remains an under-explored area.

Synthetic CSW Generation. To create training data for CSW processing, researchers have developed methods to synthesize code-mixed text. Linguistically-driven approaches use syntactic constraints. For instance, (Pratapa et al., 2018; Pratapa et al., 2021) implement the Equivalence Constraint (Poplack, 1978) by mixing Hindi–English sentence pairs according to matching parse-tree orders. Similarly, models based on the Matrix Language Framework (Myers-Scotton, 2002) generate CS sentences by embedding phrases from one language into the syntactic "frame" of the other; systems by (Lee and Li, 2019) and (Gupta et al., 2020) follow this idea. Another linguistic rule, the Functional Head Constraint (Belazi et al., 1994), has been used by (Li and Fung, 2012) to avoid switches between tightly-linked head complement pairs. By contrast, machine-translation-based methods (Tarunesh et al., 2021) treat code-switching as a target style. (Xu and Yvon, 2021) train an NMT model that takes monolingual English as input and produces code-mixed output. More recently, large parallel corpora (Winata et al., 2019) of code-mixed text (e.g. English–Hindi) have enabled data-driven synthesis. For example, (Srivastava and Singh, 2021; Srivastava and Singh, 2022) and (Gautam et al., 2021) built English–Hindi/Spanish/Arabic code-switching corpora that can be used to train or benchmark generators. These MT-based generators can produce fluent CSW sentences by effectively "translating" English into a mixed form.

2.3 GEC for Code-Switched Text

Only very recent work has applied GEC to code-switched inputs. (Chan et al., 2024) note that existing GEC systems fail on CSW text because they are trained on English-only data and cannot handle foreign words. To address this, (Chan et al., 2024) create synthetic CSW-GEC training data: they translate selected spans of monolingual GEC corpora into another language (e.g. Chinese, Korean, Japanese) to simulate learner code-switching. (Potter and Yuan, 2024) pursue a similar goal by using large language models to generate code-mixed English–L1 sentences containing grammatical errors, producing one of the first sizable CSW-GEC datasets. Both studies show that fine-tuning GEC models on these synthetic CSW examples significantly improves error correction on real code-switched learner text. However, note that these approaches still treat CS words like any other input token: the models simply attempt to correct the mixed sentence and do not explicitly label which tokens are code-switched versus erroneous.

2.4 Joint GE vs. CSW Disambiguation

To our knowledge, no prior work explicitly tags tokens as "grammatical error" vs "code-switch" in learner text. Even in CSW-focused GEC studies, any distinction between code-switching and (non-linguistic) errors is generally ignored ((Chan et al., 2024), for example, explicitly state that finer typological distinctions are outside their scope). More broadly, staged or curriculum training strategies have been explored in NLP. For instance, (Xu et al., 2020) show that fine-tuning LMs with an easy-to-hard curriculum can improve performance on NLU tasks. (Guan et al., 2025) propose a preference-based multi-stage fine-tuning paradigm for LLMs, explicitly structuring learning into multiple continuous stages. However, none of these methods have been applied to jointly disambiguate grammatical errors versus code-switching. In particular, no existing system combines GEC with a separate code-switch detector in a unified three-stage pipeline (Monolingual grammatical fluency $\rightarrow$ error-detection $\rightarrow$ CSW-detection). Our proposed three-stage model is, therefore, novel in the literature.

3 CSW GEC DATASET GENERATION

This section describes our method for generating a Code-Switched Grammatical Error Correction (CSW-GEC) dataset from an existing monolingual GEC corpus. The goal is to create training data in which code-switching and grammatical errors coexist, while keeping them clearly distinguishable. Inspired by recent work (Chan et al., 2024), we design a linguistically and semantically guided pipeline that injects code-switched phrases only into non-error regions, thereby preserving grammatical error signals for effective training.

3.1 Source Data and Error Alignment

We use the Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012) learner corpus as the base dataset, containing sentence pairs of grammatically incorrect input (processed_input) and corrected out-

put (processed_output). To identify grammatical error spans, we align token sequences using the difflib.SequenceMatcher, following approaches used in edit-tagging GEC models (e.g., GECToR; (Omelianchuk et al., 2020)). Tokens are labeled as:

- GE (Grammatical Error): tokens differing from the corrected version.
- CORRECT: tokens preserved in both the source and correction.

This annotation schema ensures we can later inject code-switched text without interfering with GE spans.

3.2 Code-Switching Injection Strategies

We apply code-switching only to GE-free spans, and use eight injection methods to maximize linguistic diversity. Each sentence is processed using one randomly selected strategy from the following:

3.2.1 Token-Level Injection

Token-level injection comprises three strategies: **ratio-token** randomly replaces verbs, adjectives, and adverbs until a target CSW ratio is reached (Mizumoto et al., 2011); **cont-token** translates a continuous span of tokens from a random start point (Muysken, 2000; Nguyen, 2020); and **noun-token** substitutes individual nouns with their code-switched equivalents to mirror natural borrowing (Myslín and Levy, 2015; Nguyen, 2018; Muysken, 2000).

3.2.2 Phrase-Level Injection

Phrase-level injection uses three methods: **rand-phrase** translates a random 2–4 token phrase, **ratio-phrase** selects a phrase matching a predefined sentence proportion, and **overlap-phrase** chooses the longest phrase with minimal overlap with grammatical error spans (Chan et al., 2024).

3.2.3 Content-Aware Injection

Content-aware injection targets semantically or emotionally salient text: **semantic_based_injection** employs Sentence-BERT embeddings to identify the most central phrase for switching (Reimers and Gurevych, 2019), while **emotion_based_injection** uses VADER sentiment scores to inject CSW into emotionally charged spans (Hutto and Gilbert, 2014; Williams et al., 2021).

3.3 Contextual Phrase Translation

In this approach, each selected phrase is first wrapped in a markup tag and submitted to the Google Translate API (via `googletrans`). To enhance fluency, a surrounding context window of approximately 30 characters on each side is translated, from which the target phrase is then extracted. Should this "carving" process fail, a fallback to word-level translation is employed, and all results are stored in a translation cache to ensure consistency and reduce API calls. By preserving local syntactic context, this method produces more natural code-switched sentences than isolated phrase translation (Song et al., 2019).

3.4 Human Re-Annotated Dataset

We manually created two test sets of 300 sentences each for English–French (EN–FR) and English–Arabic (EN–AR). Each sentence was either manually crafted by professional bilingual linguists or semi-automatically generated through prompting-based techniques using OpenAI's GPT-4o, inspired by recent work on LLM-based code-switched text generation (Potter and Yuan, 2024). The resulting sentences simulate real learner data and include both grammatical error (GE) and code-switching (CSW) phenomena. Tokens were annotated at the word level as GE, CSW, or neither, following guidelines informed by prior token-level annotation work (Aguilar et al., 2020; Khanuja et al., 2020). All sentences were independently labeled by two annotators, and any disagreements were resolved through adjudication by a senior computational linguist. Inter-annotator agreement was measured on a 10% overlap subset, yielding Cohen's $\kappa = 0.88$ (Cohen, 1960) and macro-averaged $F_1 = 0.91$, demonstrating high reliability. These EN–FR and EN–AR sets serve as gold-standard benchmarks for evaluating the precision of GE vs. CSW disambiguation systems.

3.5 Quantitative Evaluation via CSW Complexity Metrics

To quantify how well our synthetic data mirrors real-world code-switching, we compute the following measures:

Code-Mixing Index (CMI) Measures language imbalance in a sentence, as proposed by (Gambäck and Das, 2016), Multilingual Index (M-Index) Captures entropy of language distribution in the text, following (Barnett et al., 2000), I-Index (Probability of Switching) Reflects the proportion of switch points, as defined by (Guzmán et al., 2017), Burstiness Adapted from (Goh and Barabási, 2008), measuring whether CS occurs in bursts or evenly, Complexity Factors (CF1–3) From (Ghosh et al., 2017), covering language diversity, switch count, and mix-

Table 1: Code-Switching Metrics for Each Injection Strategy.

Strategy	CMI	M-Index	I-Index	Burstiness	CF1	CF2	CF3	BCE
ratio-token	14.88	0.45	0.24	-0.43	7.69	25.99	24.04	0.13
cont-token	21.40	0.63	0.31	-0.47	11.12	31.28	29.06	0.16
noun-token	9.80	0.32	0.16	-0.37	6.62	19.53	18.14	0.20
rand-phrase	21.03	0.54	0.17	-0.39	7.64	23.52	21.83	0.08
ratio-phrase	32.12	0.82	0.24	-0.36	10.77	31.66	29.36	0.11
overlap-phrase	32.05	0.79	0.21	-0.37	8.15	26.75	24.76	0.09
semantic_based	2.26	0.06	0.01	-0.02	2.66	9.16	8.59	0.01
emotion_based	5.21	0.15	0.05	-0.17	3.83	12.22	11.43	0.05
Combined (random)	22.71	0.61	0.22	-0.42	8.51	26.12	24.23	0.12
Human Annotated	24.54	0.64	0.23	-0.41	8.30	26.06	24.20	0.11

ing degree. Burst Clustering Entropy (BCE) Captures the concentration of code-switching by computing the inverse normalized entropy of monolingual span lengths, following (Shannon, 1948). It is defined as:

$$\mathrm{BCE} = 1 - \frac{-\sum_{i=1}^{k} p_i \log_2 p_i}{\log_2 k},$$

where p_i is the proportion of the i^{th} monolingual span in the sentence, and k is the total number of such spans. Higher BCE values indicate more clustered (bursty) switching.

Table 1 shows the value of each metric for all Strategy. We can see that Among all injection methods, the **Combined (random)** strategy consistently achieves the closest alignment with the **Human Annotated** across all metrics, reflecting similar switching density, structural complexity, and distributional properties. Using this method (Combined) we generated a corpus of 40,000 EN–FR and 40,000 EN–AR. While our dataset shows strong alignment with human-annotated benchmarks on structural metrics, synthetic CSW cannot fully capture the complexity of real-world learner language. In particular, it lacks sociolinguistic phenomena such as slang, informal register, regional variants, and culturally grounded usage patterns(Kandiawan, 2023; Puspita and Ardianto, 2024). In future work, we plan to supplement this dataset with authentic learner data from classroom forums, essays, and speech transcripts. This enrichment will allow us to study naturally occurring CSW behavior and improve model robustness to diverse multilingual learner profiles.

4 MODEL & TRAINING PIPELINE

To effectively distinguish grammatical errors (GE) from valid code-switching (CSW) in multilingual learner data, we adopt a three-stage curriculum learning strategy. We implement and compare two competitive multilingual Transformer architectures XLM-RoBERTa-base and mDeBERTa-v3-base under identical settings to identify the optimal backbone for this task.

4.1 Core Architectures

Our architecture supports plug-and-play of either encoder:

- XLM-RoBERTa-base (Conneau et al., 2019) is a multilingual general-purpose model, pretrained on 2.5 TB of CommonCrawl data across 100 languages. It has demonstrated strong cross-lingual performance in many evaluation benchmarks
- mDeBERTa-v3-base (He et al., 2021) advances multilingual pretraining using disentangled attention, relative positional encoding, and ELECTRA-style (Clark et al., 2020) objectives. It achieves superior zero-shot transfer—e.g., +3.6 F1 on XNLI(Conneau et al., 2018) compared to XLM-R

Both models share the same SentencePiece tokenizer and a lightweight token classification head. We fine-tune them under identical curriculum stages to ensure fair comparison.

4.2 Three-Stage Training Formulation

To progressively train the model to distinguish between grammatical errors (GE) and valid code-switching (CSW), we adopt a three-stage curriculum learning framework (Xu et al., 2020; Baheti et al., 2018). Each stage is formulated as a token-level classification task, with a shared multilingual encoder fine-tuned through increasingly complex supervision. Let $X = (x_1, \ldots, x_n)$ denote the input token sequence of length n, and let $Y^{(k)} = (y_1^{(k)}, \ldots, y_n^{(k)})$ represent the label sequence at training stage $k \in \{1, 2, 3\}$.

Table 2: token-level precision, recall, and $F_{0.5}$ on synthetic and human-annotated test sets for EN–FR, EN–AR, and combined EN–FR_EN–AR.

Model	Training	EN–FR			EN–AR			EN–FR_EN–AR		
		Prec.	**Rec.**	**$F_{0.5}$**	**Prec.**	**Rec.**	**$F_{0.5}$**	**Prec.**	**Rec.**	**$F_{0.5}$**
Synthetic Test Set										
XLM-R	Single-Stage	0.85	0.78	0.82	0.84	0.75	0.80	0.79	0.75	0.77
XLM-R	Curriculum	0.83	0.76	0.80	0.85	0.79	0.83	0.82	0.77	0.80
mDeBERTa	Single-Stage	0.86	0.80	0.84	0.86	0.81	0.85	0.84	0.80	0.82
mDeBERTa	Curriculum	**0.88**	**0.82**	**0.86**	**0.89**	**0.84**	**0.87**	**0.86**	**0.81**	**0.83**
Human-Annotated Test Set										
XLM-R	Single-Stage	0.82	0.74	0.78	0.81	0.73	0.77	0.77	0.73	0.77
XLM-R	Curriculum	0.84	0.76	0.80	0.83	0.75	0.81	0.80	0.78	0.79
mDeBERTa	Single-Stage	0.87	0.79	0.85	0.87	0.80	0.86	0.84	0.79	0.80
mDeBERTa	Curriculum	**0.89**	**0.83**	**0.88**	**0.90**	**0.85**	**0.89**	**0.87**	**0.83**	**0.84**

We denote the shared encoder(e.g., XLM-RoBERTa or mDeBERTa) as f_θ, and the stage-specific classification head as $h_\phi^{(k)}$. The model is trained to minimize the token-level cross-entropy loss at each stage:

$$\theta^{(k)}, \phi^{(k)} = \arg\min_{\theta,\phi} \mathcal{L}^{(k)} \left(h_\phi^{(k)}(f_\theta(X)), Y^{(k)} \right) \quad (1)$$

At each stage, the encoder is initialized from the previous stage's checkpoint (i.e., $\theta^{(k)} \leftarrow \theta^{(k-1)}$), while the classification head $\phi^{(k)}$ is trained from scratch.

Stage 1: Grammaticality Pretraining. The first stage introduces the model to fluent, monolingual English. Each input X is a corrected sentence from the Lang-8 corpus (`originalCorrected`), and the labels $Y^{(1)}$ are uniformly set to "O", indicating the absence of grammatical errors or code-switching. This step aligns the encoder with grammatical structure and provides task-domain adaptation.

Stage 2: Grammatical Error (GE) Detection. The second stage trains the model to detect grammatical errors in learner English. The inputs X come from uncorrected Lang-8 sentences (`original`), and the label sequence $Y^{(2)}$ follows the IOB2 format with tags from {B-GE, I-GE, O}. Initialized from Stage 1, the encoder is fine-tuned to identify GE spans, improving its sensitivity to ungrammatical patterns.

Stage 3: Joint GE and CSW Detection. The final stage requires the model to simultaneously detect grammatical errors and code-switched spans. The input X consists of synthetic CSW-GEC sentences containing French or Arabic segments. Labels $Y^{(3)}$ use an extended tag set {B-GE, I-GE, B-CSW, I-CSW, O}. The encoder, initialized from Stage 2, is fine-tuned with a five-class classification head to distinguish between ungrammatical and code-switched sequences at the token level.

4.3 Training Setup and Optimization

All models are trained using the AdamW optimizer with a weight decay of 0.01 and a batch size of 16. A linear learning rate schedule is applied, with the first 10% of steps dedicated to warmup followed by linear decay. Tokenization is handled using a shared SentencePiece model. Early stopping is applied in all stages based on the macro-F1 score computed on a 5% validation split. In the three-stage curriculum, Stage 1 trains a single-label model with only the "O" class for 2 epochs at a learning rate of 3e-5. Stage 2 expands the label set to three classes (O, B-GE, I-GE) and is trained for 5 epochs using a learning rate of 2e-5. In Stage 3, the model uses the full five-label IOB tagging scheme and is trained for 10 epochs at a lower learning rate of 3e-6. For comparison, a simple fine-tuning baseline initializes the full five-label head from the start and trains the model end-to-end for 15 epochs at a learning rate of 5e-5. All training setups use the same regularization strategy, batch size, warmup schedule, and early stopping criterion. Evaluation metrics include macro-averaged precision, recall, and F0.5.

5 RESULTS AND ANALYSIS

5.1 Main Results: Architecture Comparison

Table 2 presents a comparative evaluation of two multilingual Transformer backbones XLM-RoBERTa (XLM-R) and mDeBERTa under both single-stage

(A) Correct Detection (EN–FR)
Sentence: "Even though she was tired, *elle a continué à travailler* without a break."
Gold: No GE; "elle a continué à travailler" → *CSW* ("she continued to work")

Pred.: French span correctly identified as *CSW*; no GE flagged
Comment: Model detects CSW without over-flagging

(B) Missed GE (EN–AR)
Sentence: "He don't "يريد أن يُكمِل المشروع" before Friday."

Gold: "don't" → *GE* (should be "doesn't"); "يريد أن يُكمِل المشروع" → *CSW* ("wants to complete the project")
Pred.: "يريد أن يُكمِل المشروع" → *CSW*

Comment: Model misses English verb error

(C) CSW Misclassified as GE (EN–FR)
Sentence: "They wanted to try the new *restaurant gastronomique* near campus."
Gold: "restaurant gastronomique" → *CSW*

Pred.: "restaurant gastronomique" → *GE*
Comment: Valid French noun phrase wrongly flagged as GE

(D) Missed GE and CSW (EN–FR)
Sentence: "The woman who *travaille* there always arrive late."
Gold: "travaille" → *CSW* (French verb: "works"); "arrive" → *GE* (subject–verb agreement; should be "arrives")
Pred.: No detections.
Comment: Model misses both the agreement error and the French code-switch

Figure 1: Complex examples illustrating model strengths and weaknesses: (A) correct CSW GE detection, (B) Missed GE, (C) CSW misclassified as GE, and (D) missed detection of both GE and CSW.

fine-tuning and our proposed three-stage curriculum learning framework. Experiments were conducted on human-annotated token-level test sets for English–French (EN–FR), English–Arabic (EN–AR), and their combination. Each token was labeled as a grammatical error (GE), a code-switched token (CSW), or neither, and performance was measured using macro-averaged $F_{0.5}$ scores to prioritize precision. Across all evaluation settings, curriculum-based training consistently outperformed the single-stage baseline. Specifically, XLM-R improved from 0.78 to 0.80 on EN–FR, from 0.77 to 0.81 on EN–AR, and from 0.77 to 0.79 on the combined test set. mDeBERTa exhibited even greater improvements, increasing from 0.85 to 0.88 on EN–FR, from 0.86 to 0.89 on EN–AR, and from 0.80 to 0.84 on the combined evaluation. These results underscore two key findings: first, that curriculum learning enhances both grammatical error and code-switching detection capabilities; and second, that mDeBERTa, when trained under this framework, offers the most robust and generalizable performance across diverse multilingual inputs. Consequently, we adopt curriculum-trained mDeBERTa as our primary model for subsequent analysis.

5.2 Ablation: Impact of Curriculum Staging

To assess the effectiveness of curriculum learning, we perform an ablation study comparing full three-stage training against simplified alternatives. The results clearly indicate that curriculum staging significantly improves both grammatical error (GE) detection and code-switch (CSW) disambiguation.

In particular, the full curriculum comprising **Stage 1: Grammaticality Pretraining**, **Stage 2: GE Detection**, and **Stage 3: Joint GE+CSW Detection** offers structured progression that helps the model gradually acquire syntactic fluency, error sensitivity, and finally, multilingual robustness. This staged learning enables the model to accurately distinguish between GEs and CSW spans, which often share superficial lexical or structural similarities. Skipping Stage 1, which aligns the model with grammatical English, reduces accuracy in detecting subtle grammatical errors. Likewise, omitting Stage 2 impairs the model's ability to differentiate GE from CSW, leading to overgeneralization in Stage 3. In contrast, the full curriculum encourages more precise token-level classifications, as evidenced by the improved $F_{0.5}$ scores across both synthetic and human-annotated test sets.

These findings confirm that curriculum staging is not merely additive but essential: it scaffolds linguistic competencies in a pedagogically aligned manner, directly enhancing performance on multilingual learner data.

5.3 Error Analysis

We performed a qualitative analysis of 300 model predictions from the EN–FR and EN–AR test sets to better understand the model's behavior in challenging cases. Figure 1 presents four representative examples that illustrate both typical strengths and recurring failure modes of the curriculum-trained model. Most residual errors occur near the boundary between grammatical errors (GE) and code-switching (CSW),

where linguistic structures become more complex and ambiguous. Common failure patterns include: (i) mistakenly flagging valid CSW tokens as grammatical errors, (ii) overlooking subtle GEs that appear close to code-switched spans, and (iii) failing to detect either phenomenon when they occur in close proximity. These observations suggest the need for improved boundary-aware contextual encoding and the inclusion of training examples where GE and CSW spans deliberately overlap. Such refinements could help the model better distinguish token-level phenomena in highly entangled multilingual input. While this work focuses on disambiguation performance, it does not yet assess the model's effectiveness in real learning environments. Educational metrics such as feedback adoption, observed learner improvement, and user satisfaction remain essential for validating pedagogical impact (Yuce et al., 2019). To address this, we plan to integrate our system into an Intelligent Tutoring System (ITS) and evaluate its impact through A/B testing (Baker et al., 2022), learner surveys, and usage analytics.

6 CONCLUSION

This study introduced a curriculum-based framework to disentangle grammatical errors (GE) from code-switching (CSW) in multilingual learner writing. By generating linguistically informed synthetic CSW data, constructing two high-quality human-annotated benchmarks (EN–FR and EN–AR), and applying a three-stage fine-tuning strategy to multilingual Transformer models, we achieved precise token-level disambiguation between GE and CSW. Experimental results demonstrated that curriculum-trained mDeBERTa consistently outperformed both single-stage and XLM-R baselines, achieving superior $F_{0.5}$ scores across synthetic and real-world evaluations. These findings confirm the pedagogical potential of our framework for enhancing feedback in Intelligent Tutoring Systems (ITS) and promoting inclusive language learning that recognizes valid multilingual expression. Despite these promising results, several challenges remain before full deployment in educational contexts.

7 LIMITATIONS

While the proposed framework makes significant progress in distinguishing grammatical errors from code-switching, it presents several limitations. First, our experiments rely largely on synthetic code-switched data generated through controlled linguistic injection. Although structurally aligned with human annotations, this data cannot fully capture the sociolinguistic richness of authentic learner communication, including informal expressions, idiomatic usage, and cultural variation. Second, the current focus on two language pairs (EN–FR and EN–AR) restricts generalization to broader multilingual contexts. Extending evaluation to typologically diverse pairs such as English–Spanish or English–Chinese would provide a more comprehensive validation. Third, while our human-annotated benchmarks achieved high reliability ($\kappa = 0.88$), their limited size constrains the statistical significance and diversity of observed error patterns. Fourth, the curriculum learning process assumes a linear progression of difficulty; more adaptive or dynamic sequencing strategies could further enhance model transferability and learning stability. Finally, this study concentrates on token-level disambiguation rather than user-facing interaction. Future research should embed the model within operational ITS environments to assess pedagogical metrics such as feedback uptake, learner improvement, and engagement over time.

ACKNOWLEDGEMENTS

The author conducted the conception, implementation, experimentation, and analysis of this work. The author gratefully acknowledges the supervisors for their guidance, methodological advice, and constructive feedback throughout all stages of the research, including manuscript review and correction.

REFERENCES

Aguilar, G., Kar, S., and Solorio, T. (2020). Lince: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the 12th Workshop on Computational Approaches to Code Switching (CALCS)*.

Baheti, A., Sitaram, S., Choudhury, M., and Bali, K. (2018). Curriculum design for code-switching: Experiments with language identification and language modeling with deep neural networks. In *ICON 2017*.

Baker, R. S., Nasiar, N., Gong, W., and Porter, C. (2022). The impacts of learning analytics and a/b testing research: a case study in differential scientometrics. *International Journal of STEM Education*, 9(1):16.

Barnett, R., Codó, E., Eppler, E., Forcadell, M., Gardner-Chloros, P., van Hout, R., Moyer, M., Torras, M. C., Turell, M. T., Sebba, M., Starren, M., and Wensing, S. (2000). *The LIDES Coding Manual: A document for preparing and analyzing language interaction data*

Version 1.1–July, 1999. University of Pennsylvania, Philadelphia, PA.

Belazi, H. M., Rubin, E. J., and Toribio, A. J. (1994). Code switching and X-bar theory: The functional head constraint. *Linguistic Inquiry*, 25(2):221–237.

Burchell, L., Birch, A., Thompson, R., and Heafield, K. (2024). Code-switched language identification is harder than you think. In Graham, Y. and Purver, M., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Long Papers)*, pages 646–658, St. Julian's, Malta. Association for Computational Linguistics.

Cenoz, J. and Gorter, D. (2021). *Pedagogical Translanguaging and Its Application to Language Classes*. Elements in Language Teaching. Cambridge University Press.

Chan, K. W. H., Bryant, C., Nguyen, L., Caines, A., and Yuan, Z. (2024). Grammatical error correction for code-switched sentences by learners of English. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N., editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7926–7938, Torino, Italia. European Language Resources Association (ELRA).

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations (ICLR)*.

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

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. (2018). Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Felice, M. and Yuan, Z. (2014). Generating artificial errors for grammatical error correction. In Wintner, S., Elliott, D., Garoufi, K., Kiela, D., and Vuli
ć, I., editors, *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126, Gothenburg, Sweden. Association for Computational Linguistics.

Gambäck, B. and Das, A. (2016). Comparing the level of code-switching in corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

Gautam, D., Kodali, P., Gupta, K., Goel, A., Shrivastava, M., and Kumaraguru, P. (2021). CoMeT: Towards code-mixed translation using parallel monolingual sentences. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 47–55, Online. Association for Computational Linguistics.

Ghosh, S., Ghosh, S., and Das, D. (2017). Complexity metric for code-mixed social media text. In *Proceedings of the 2017 International Conference on Computer Science and Engineering (CSE)*, pages 693–698, Kolkata, India. IEEE.

Goh, K.-I. and Barabási, A.-L. (2008). Burstiness and memory in complex systems. *EPL (Europhysics Letters)*, 81(4):48002.

Guan, C., Huang, C., Li, H., Li, Y., Cheng, N., Liu, Z., Xu, J., Chen, Y., and Liu, J. (2025). Multi-stage LLM fine-tuning with a continual learning setting. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3154–3169, Mexico City, Mexico. Association for Computational Linguistics.

Gupta, D., Ekbal, A., and Bhattacharyya, P. (2020). A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Proceedings of the 14th LREC Workshop on NLP for Code-Mixing (NLP4CMC 2020)*, pages 1–10, Marseille, France (online). European Language Resources Association (ELRA).

Guzmán, G., Ricard, J., Serigos, J., Bullock, B. E., and Toribio, A. J. (2017). Metrics for modeling code-switching across corpora. In *Proceedings of the 3rd Workshop on Computational Approaches to Linguistic Code-Switching*, pages 66–77, Copenhagen, Denmark. Association for Computational Linguistics.

He, P., Gao, J., and Chen, W. (2021). Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *arXiv preprint arXiv:2111.09543*.

Hutto, C. and Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM)*, pages 216–225. The AAAI Press.

Kandiawan, A. B. (2023). Code-switching and slang used by gen z indonesians on social media. *English Language Teaching and Research Journal*, 7(1):48–56.

Khanuja, S., Dandapat, S., Srinivasan, A., Sitaram, S., and Choudhury, M. (2020). Gluecos: An evaluation benchmark for code-switched nlp. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Lee, G. and Li, H. (2019). Modeling code-switch languages using bilingual parallel corpus. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2156–2165, Hong Kong, China. Association for Computational Linguistics.

Li, Y. and Fung, P. (2012). A Mandarin-English code-switching corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2515–2519, Istanbul,

Turkey. European Language Resources Association (ELRA).

Liu, Z., Yin, S. X., and Chen, N. (2024a). Optimizing code-switching in conversational tutoring systems: A pedagogical framework and evaluation. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 500–515. Association for Computational Linguistics.

Liu, Z., Yin, S. X., Lin, G., and Chen, N. F. (2024b). Personality-aware student simulation for conversational intelligent tutoring systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 626–642. Association for Computational Linguistics.

Mizumoto, T., Komachi, M., Nagata, M., and Matsumoto, Y. (2011). Mining revision log of language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Molina, G., AlGhamdi, F., Ghoneim, M., Hawwari, A., Rey-Villamizar, N., Diab, M., and Solorio, T. (2016). Overview for the second shared task on language identification in code-switched data. In Diab, M., Fung, P., Ghoneim, M., Hirschberg, J., and Solorio, T., editors, *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Muysken, P. (2000). *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press, Cambridge, UK.

Myers-Scotton, C. (2002). *Contact Linguistics: Bilingual Encounters and Grammatical Outcomes*. Oxford University Press, Oxford.

Myslín, M. and Levy, R. (2015). Code-switching and predictability of meaning in discourse. *Language*, 91(4):771–808.

Nguyen, H. N. (2020). *Cross-generational linguistic variation in the Canberra Vietnamese heritage language community: A corpus-centred investigation*. PhD thesis, University of Cambridge.

Nguyen, L. (2018). Borrowing or code-switching? traces of community norms in vietnamese-english speech. In *Proceedings of the 3rd Workshop on Computational Approaches to Linguistic Code-Switching*, pages 1–11, Santa Fe, New Mexico. Association for Computational Linguistics.

Nguyen, L., Yuan, Z., and Seed, G. (2022). Building educational technologies for code-switching: Current practices, difficulties and future directions. 7(3):220.

Omelianchuk, K., Atrasevych, V., Chernodub, A., and Skurzhanskyi, O. (2020). GECToR – grammatical error correction: Tag, not rewrite. In Burstein, J., Kochmar, E., Leacock, C., Madnani, N., Tetreault, J., and Yannakoudakis, H., editors, *Proceedings of the 15th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–159, Seattle, WA, USA (online). Association for Computational Linguistics.

Poplack, S. (1978). *Sometimes I'll start a sentence in Spanish y Termino en Español: Toward a Typology of Code-Switching*. Centro de Estudios Puertorriqueños, CUNY, New York.

Potter, T. and Yuan, Z. (2024). LLM-based code-switched text generation for grammatical error correction. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N., editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024)*, pages 16957–16965, Miami, Florida, USA. Association for Computational Linguistics.

Pratapa, A., Anastasopoulos, A., Rijhwani, S., Chaudhary, A., Mortensen, D. R., Neubig, G., and Rudra, S. (2021). Evaluating the morphosyntactic well-formedness of generated texts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7154–7170, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pratapa, A., Bhat, G., Choudhury, M., Sitaram, S., Dandapat, S., and Bali, K. (2018). Language modeling for code-mixing: The role of linguistic theory based synthetic data. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1549–1559, Melbourne, Australia. Association for Computational Linguistics.

Puspita, V. G. and Ardianto, A. (2024). Code-switching and slang: An analysis of language dynamics in the everyday lives of generation z. *Linguistics Initiative*, 4(1).

Pérez-Ortiz, J. A., Esplà-Gomis, M., Sánchez-Cartagena, V. M., Sánchez-Martínez, F., Chernysh, R., Mora-Rodríguez, G., and Berezhnoy, L. A conversational intelligent tutoring system for improving english proficiency of non-native speakers via debriefing of online meeting transcriptions. pages 187–198.

Ramadhan, A., Warnars, H. L. H. S., and Razak, F. H. A. (2024). Combining intelligent tutoring systems and gamification: a systematic literature review. 29(6):6753–6789.

Ramanarayanan, V. and Pugh, R. (2018). Automatic token and turn level language identification for code-switched text dialog: An analysis across language pairs and corpora. In *Proceedings of the 19th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 80–88, Sapporo, Japan. Association for Computational Linguistics.

Rei, M., Felice, M., Yuan, Z., and Briscoe, T. (2017). Artificial error generation with machine translation and syntactic patterns. In Tetreault, J., Burstein, J., Leacock, C., and Yannakoudakis, H., editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rozovskaya, A. and Roth, D. (2010). Training paradigms for correcting errors in grammar and usage. In Kaplan, R., Burstein, J., Harper, M., and Penn, G., editors, *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 154–162, Los Angeles, California. Association for Computational Linguistics.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423.

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A., and Fung, P. (2014). Overview for the first shared task on language identification in code-switched data. In Diab, M., Hirschberg, J., Fung, P., and Solorio, T., editors, *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Song, K., Zhang, Y., Yu, H., Luo, W., Wang, K., and Zhang, M. (2019). Code-switching for enhancing nmt with pre-specified translation.

Sonkar, S., Liu, N., Mallick, D., and Baraniuk, R. (2023). CLASS: A design framework for building intelligent tutoring systems based on learning science principles. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1941–1961. Association for Computational Linguistics.

Srivastava, V. and Singh, M. (2021). Challenges and limitations with the metrics measuring the complexity of code-mixed text. In Solorio, T., Chen, S., Black, A. W., Diab, M. T., Sitaram, S., Soto, V., Yilmaz, E., and Srinivasan, A., editors, *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14, Online. Association for Computational Linguistics.

Srivastava, V. and Singh, M. (2022). Overview and results of MixMT shared-task at WMT 2022. In Koehn, P., Barrault, L., Bojar, O., Bougares, F., Chatterjee, R., Costa-juss
'a, M. R., Esplà-Gomis, M., Federmann, C., Ferrando, S., Fonollosa, J. A. R., Forcada, M. L., Fraser, A., Guzm
án, F., Haddow, B., Hiraoka, K., Kurohashi, S., Ljubešić, N., Mehandjiev, N., Mirkin, D., Nakazawa, T., Ndungo, N., Negri, M., Niraula, N., Nouno, N., Ormazabal, A., Paun, S., Pereira, D., Post, M., Ratajczyk, M., Savkov, A., Specia, L., and Turchi, M., editors, *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 806–811, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Stahlberg, F. and Kumar, S. (2021). Synthetic data generation for grammatical error correction with tagged corruption models. In Burstein, J., Horbach, A., Kochmar, E., Laarmann-Quante, R., Leacock, C., Madnani, N., Pilàn, I., Tetreault, J., and Yannakoudakis, H., editors, *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

Tajiri, T., Komachi, M., and Matsumoto, Y. (2012). Tense and Aspect Error Correction for ESL Learners Using Global Context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Tarunesh, I., Kumar, S., and Jyothi, P. (2021). From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3154–3169, Online. Association for Computational Linguistics.

Williams, A., Srinivasan, M., Liu, C., Lee, P., and Zhou, Q. (2021). Why do bilinguals code-switch when emotional? insights from immigrant parent-child interactions. *Language Learning and Development*, 17(4):387–406.

Winata, G. I., Madotto, A., Wu, C.-S., and Fung, P. (2019). Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, Hong Kong, China. Association for Computational Linguistics.

Xu, B., Zhang, L., Mao, Z., Wang, Q., Xie, H., and Zhang, Y. (2020). Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Xu, J. and Yvon, F. (2021). Can you traducir this? machine translation for code-switched input. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 44–52, Online. Association for Computational Linguistics.

Xu, P., Saghir, H., Kang, J. S., Long, T., Bose, A. J., Cao, Y., and Cheung, J. C. K. (2019). A cross-domain transferable neural coherence model. In Korhonen, A., Traum, D., and M
'arquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 678–687, Florence, Italy. Association for Computational Linguistics.

Yuan, Z., Felice, M., Rei, M., Kochmar, E., and Briscoe, T. (2019). Neural grammatical error correction with noise-aware training. In Korhonen, A., Traum, D., and M
'arquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 688–698, Florence, Italy. Association for Computational Linguistics.

Yuce, A., Abubakar, A. M., and Ilkan, M. (2019). Intelligent tutoring systems and learning performance: Applying task-technology fit and is success model. *Education and Information Technologies*, 24(4):2655–2675.