LLM-based Code-Switched Text Generation forError Correction

Tom Potter*University of Manchester [thomas.potter@postgrad.manchester.ac.uk](mailto:thomas
Zheng Yuan King's College London zheng.yuan@kcl.ac.uk

Abstract

With the rise of globalisation, code-switching (CSW) has become a ubiquitous part of multilingual conversation, posing new challenges for natural language processing (NLP), espe- cially in Grammatical Error Correction (GEC). This work explores the complexities of apply- ing GEC systems to CSW texts. Our objectives include evaluating the performance of state- of-the-art GEC systems on an authentic CSW dataset from English as a Second Language (ESL) learners, exploring synthetic data gener- ation as a solution to data scarcity, and develop- ing a model capable of correcting grammatical errors in monolingual and CSW texts. We gen- erated synthetic CSW GEC data, resulting in one of the first substantial datasets for this task, and showed that a model trained on this data is capable of significant improvements over exist- ing systems. This work targets ESL learners, aiming to provide educational technologies that aid in the development of their English gram- matical correctness without constraining their natural multilingualism.

1 Introduction

Code-switching (CSW), the practice of fluidly al- ternating between two or more languages in conver- sation, has become commonplace in recent years. This linguistic phenomenon, emerging as a natu- ral consequence of multilingualism, is now widely accepted in social and professional settings (Yow et al., 2018). Many works have highlighted the utility and cultural importance of CSW in general conversation (Beatty-Martínez et al., 2020; Falbo and LaCroix, 2021). Further research indicates that these advantages extend to language learning, with CSW offering many pedagogical benefits. These include increasing students' access to content and improving their confidence. Nguyen et al. (2022) discuss the mechanisms for this, where students

use a familiar language to grasp foreign, complex concepts. CSW can also serve as a scaffolding tool, helping to bridge gaps in a student's com- prehension of a language and enabling them to build upon existing knowledge. These benefits reduce the barriers between a student and their target language and help promote a learning en- vironment conducive with active exploration and deeper understanding. Therefore, it is essential that English as a Second Language (ESL) learn- ers are not penalised for expressing their cultural identity through CSW. Grammatical error correc- tion (GEC) is the task of automatically detecting and correcting errors in text. Research on GEC for CSW text remained largely unexplored. Chan et al. (2024) were the first to demonstrate that exposing a sequence-tagging GEC model to CSW text during the training process improves performance com- pared to a monolingual system. However, further work is essential to ensure language technology is inclusive and reflective of real-world linguistic practices. Figure 1 shows two examples of CSW from our target population with their grammatical corrections.[1]

---

*Work completed whilst at King's College London.

[1]The definition of CSW is a subject of ongoing debate. Throughout this work, we use the term CSW to refer specifi- cally to the type of language mixing exhibited by ESL learners.

Figure 1: Examples of GEC in ESL learner language.

Example 1: According to the test, [lacks in me → my shortcomings] are and .
Example 2: When we [call → say] , do we actually mean a glove compartment in English?

Despite significant advancements in GEC in re- cent years, a gap persists in addressing CSW texts, with monolingual GEC datasets labelling CSW as a type of error (Nguyen et al., 2022). There are several reasons for this, the most prominent being the scarcity of high-quality training data, a prob- lem that plagues monolingual GEC systems. The unique linguistic features of CSW, including its variable syntax, semantics and pragmatics, add additional complexity to this task. Monolingual seq2seq GEC models, e.g. T5 (Rothe et al., 2021), struggle with CSW text as they fail to represent the non-English inputs, resulting in their inability to output the CSW text. On the other hand, multi- lingual seq2seq models and edit-based GEC mod- els like GECToR (Omelianchuk et al., 2020) can handle CSW text but struggle with the ambiguity present at language switching points. This ambi- guity challenges the models' ability to accurately correct the text.

This paper aims to bridge this gap. Firstly, to ad- dress the data scarcity issue, we propose a method for generating high-quality synthetic CSW GEC data, using which we produce, to our knowledge, one of the first substantial datasets labelled for this task.[2] Secondly, we train a token classification- style GEC system, tailored to correct errors in texts produced by ESL learners. This demographic is significant for our study as they not only present consistent CSW patterns but also stand to benefit greatly from a GEC system capable of handling CSW text.

2 Data

2.1 Genuine CSW GEC Dataset

One of the only datasets labelled for GEC which does not remove CSW text, is the Lang- 8 dataset (Mizumoto et al., 2013), sourced from the Lang-8 language learning platform. This dataset, when filtered to contain entries where CSW is present, offers a foundation of authen- tic data, comprises 5,875 pairs of ungrammati- cal and corrected sentences across 6 CSW lan- guage pairs: English-Japanese (81.9Korean (13.0(3.4(0.5

The crowd-sourced nature of Lang-8 required manual validation to ensure accuracy. We tasked an annotator with the responsibility of verifying the original corrections in the dataset, as well as combing for missed errors, incorrect annotations and over-annotations.

2.2 Synthetic CSW GEC Data Generation

Given the small size of the available CSW GEC dataset, we introduced a 2-step approach to syn- thetic CSW GEC data generation. First, we gener- ated grammatically correct CSW sentences. This is followed by the introduction of errors.

2.2.1 Step 1: CSW Text Generation

Three different synthetic data generation tech- niques have been explored to generate CSW data.

---

[2]This dataset is available on GitHub.

Translation-based CSW Text Generation re- quired a monolingual corpus, a machine translation (MT) algorithm, and a sentence parser. To generate a CSW utterance, we used the Stanford Parser v4.5.4[3] (Manning et al., 2014) to build a syn- tactic parse tree. We then randomly selected and translated a subtree using the ArgosTranslate MT package[4] (Finlay, 2023; Klein et al., 2017). This method generates plausible CSW text. However, performance is dependent on the strength of the parsing and translation algorithms; and the style of language within the corpus. To approximate the style of our authentic CSW text, we used corrected monolingual sentences from the Lang-8 corpus.

Parallel Corpus-based CSW Text Generation avoids the need for a translation algorithm. Instead, we used the same Stanford Parser, this time with Spanish, French and German configurations; and the AWESOME word-alignment model[5] (Dou and Neubig, 2021), to identify parts of parallel cor- pora labelled for MT with similar syntactic struc- ture. For this method, we use the Europarl corpus (Koehn et al., 2012) due to the grammatical quality of its English component. Under the Equivalence Constraint Theory (Rizvi et al., 2021), these areas are where CSW is likely to take place. We, there- fore, randomly chose overlapping subtrees as can- didates for injection of non-English text. Although this method does not require MT, it is reliant on performant word-alignment and parsing systems; these are rare for many languages.

LLM Prompting-based CSW Text Generation The other methods of generating CSW text rely on injecting a second language into existing monolin- gual corpora. Hence, they are not able to recreate one of the main switching styles shown by ESL learners — CSW as a genuine pragmatic strategy. A common reason for this style of switching is when quoting another language. It is difficult to recreate this style using a monolingual foundation as sen- tences like The Japanese word for 'dog" is '" seldom appear in authentic monolingual corpora.

To generate diverse CSW texts without relying on existing corpora or inaccurate alignment algo- rithms, we leveraged the strong general knowl- edge of Large Language Models (LLMs). We demonstrated that OpenAI's GPT-3.5 (Brown et al., 2020) can create high-quality CSW sentences when shown examples of authentic utterances. Along with genuine CSW texts, we supplied a one-shot example of how to use the switching styles of an existing CSW text to generate a new sentence.[6]

Comparison of Synthetic CSW Text We used several CSW metrics to quantify the qualities of CSW texts: Code Mixing Index (CMI) (Gam- bäck and Das, 2016), Multilingual Index (M-Index) (Barnett et al., 2000), Probability of Switching (I- Index) (Guzmán et al., 2017), Burstiness (Goh and Barabási, 2008), and Complexity Factor (CF1-3) (Ghosh et al., 2017). Table 1 shows the value of each metric for our genuine CSW dataset, as well as for these 3 synthetic CSW datasets. We can see that the LLM prompting-based dataset was superior in its similarity to the authentic CSW data. Using this method, we generated a corpus of 73,293 utter- ances covering over 20 English language pairs, in- cluding the 6 language pairs in the original dataset.[7]

---

[3]This can be downloaded from the CoreNLP website.

[4]We used ArgosTranslate v1.8.0, available on GitHub.

[5]The authors shared their model on HuggingFace.

[6]The full prompt can be seen in Appendix A.

[7]The LLM does not always generate the language pairs we ask for. However, these sentences are still included in the dataset categorised under their actual language pair.

Table 1: Quantitative Description of the Genuine and Generated CSW Datasets Using Various CSW Metrics.

Metric	Genuine CSW	LLM CSW	Translation CSW	Corpus CSW heightCMI
15.52	16.14	27.81	11.42 M-Index	0.007
0.004	0.015	0.006 I-Index	0.21	0.21
0.30	0.20 Burstiness	-0.07	-0.04	0.03
-0.11 CF1	6.38	5.82	17.13	2.55 CF2
19.77	19.03	31.11	16.04 CF3	18.34
17.61	30.05	14.20 height		

2.2.2 Step 2: Synthetic Error Generation

Several works have shown the effectiveness of rule-based error injection for GEC data genera- tion. Many use the PIE-synthetic dataset (Awasthi et al., 2019), a perturbed version of the 1BW cor- pus (Chelba et al., 2013). For each sentence, the authors introduce between 0 and 4 errors of ran- dom type. We extended this work by introducing a new subset of error types that are not only more common in ESL learners, but also are areas where the SOTA performance collapses when faced with CSW text: noun, pronoun, word order, determiner, and punctuation errors.[8]

To increase the diversity of errors, we adopted a second style of error injection, Backtransla- tion (Stahlberg and Kumar, 2021). By swapping the source and target sentences of a monolingual dataset, we trained a GECToR-based system to in- duce errors in our synthetic CSW sentences.

Using these methods, we created two datasets: Syn-CSW PIE and Syn-CSW Rev-GECToR. After removing pairs containing no injected errors, we are left with 70,180 and 18,159 sentences each.

3 CSW GEC Systems

For our GEC system targeting CSW texts, we chose a GECToR model (Omelianchuk et al., 2020), with a RoBERTa-base foundation, due to its proven effi- cacy with limited training data and stronger perfor- mance on CSW texts compared to seq2seq models. We added a new CSW class to the error detection head, adding the ability to detect CSW tokens.

Following Tarnavskyi et al. (2022), we used a 3-stage training schedule. In the first, we used the same distilled 1BW corpus, and added all our synthetic CSW GEC data. For the second, we used several GEC datasets: NUCLE (Dahlmeier et al., 2023), FCE (Yannakoudakis et al., 2011), WI Locness (Bryant et al., 2019), Lang-8 and our 2 synthetic CSW datasets. As our genuine CSW dataset is a subset of the Lang-8 corpus, we checked and removed any duplicates. Following previous works, we finished training using the WI Locness dataset due to its superior quality. In this final stage, we added a sampled subset of our syn- thetic CSW sentences and 90CSW data, ensuring exposure to synthetic and gen- uine CSW text. At each stage, we reserved 5the data for validation.[9] Finally, we tuned inference parameters using a grid-search to optimise the $F_{0.5}$ on the final validation set. By beginning with pre- training on large amounts of lower-quality data in the early stages, this multi-stage learning process allows the model to first build a robust GEC foun- dation before refining it with high quality data in the latter stages. This approach allows the model to learn incrementally, reducing the risk of the model being overwhelmed by the complexity of the task from the outset.

---

[8]Error type analysis is presented in Appendix B.
[9]An exact breakdown of contributions by each dataset is given in Appendix C.

Table 2: ERRANT-based Precision, Recall and $F_{0.5}$ Scores of Baselines and Our Model Throughout Training

2*Model			BEA-2019 Test				
		P	R	$F_{0.5}$	P		
Existing GEC systems	heightGECToR	77.88	53.07	71.22	71.14		
62.03		47.19	58.34	11.70	24.98	13.09	
67.23		53.88	64.05	66.15	26.04	50.57	
51.73		67.20	65.41	29.93	52.87	Stage 3	
68.92		84.66	22.92	55.02 Inference Tweaks	69.01		
76.02		38.67	63.71 height				

4 Results and Analysis

4.1 Baseline Comparisons

We compared our model against two well- established systems: a RoBERTa-base GECToR model (Omelianchuk et al., 2020), with near SOTA performance on the BEA-2019 test set (Bryant et al., 2019), and a seq2seq T5 model (Rothe et al., 2021). To assess these models, we evaluated their performance on the BEA-2019 test set and the re- maining 10RANT (Bryant et al., 2017) GEC evaluation results, as outlined in Table 2, demonstrate a clear degra- dation in performance when these two systems are applied to CSW texts. The ERRANT toolkit de- tects and classifies edits between source and target sentence pairs into predefined error categories. It enables the comparison of a proposed set of edits with a reference set, providing a way of calculating metrics, such as precision and recall, across these categories.
Note: The `\multirow` command is not available without an additional package. The header will compile as plain text by removing `\multirow` if needed.

4.2 Detailed Model Performance

The progression of our model throughout training provided insights into its evolving capabilities and effectiveness of our synthetic data. We monitored several metrics, including the ERRANT precision, recall and $F_{0.5}$ score, for the BEA-2019 test set and the remaining unused 10dataset. These metrics, as displayed in Table 2, indicate a steady improvement in the ability to handle CSW texts. Notably, the performance on the CSW dataset shows a significant leap in the final stages, where the contribution of our synthetic dataset is largest. This improvement in CSW text handling did slightly compromise the model's per- formance on monolingual GEC tasks, as seen on the BEA-2019 test set. This suggests a trade-off in- herent in specialising the model for CSW contexts. However, our model remains competitive amongst SOTA monolingual GEC systems of its size.

Three illustrative examples of our model's cor- rections, taken from the CSW test set, can be seen in Figure 2. The first example demonstrates a case where the model has correctly identified all of the changes required, including the incorrect capitalisa- tion of a word, a missing word, and some missing punctuation. The second example shows a 'near miss"; here, the model has correctly identified the majority of the changes required but dropped the 'I" whilst rearranging the start of the sentence. Fi- nally, the third example presents a scenario where the model has fallen slightly short, failing to recog- nise the need for 'were" instead of 'was" in this hypothetical context.
Gold Correction 1: We have many [New → new] words for [∅ → the] unemployed[∅ → :] '"[∅ → ,] ' "[∅ → ,] ""

Figure 2: Three examples of model's proposed corrections from the CSW test set.

Proposed Correction 1: We have many [New → new] words for [∅ → the] unemployed[∅ → :] '"[∅ → ,] '"[∅ → ,] ""

Gold Correction 2: [I and my girlfriend → My girlfriend and I] looked [∅ → at a] picture called "" (Immaculate Conception).

Proposed Correction 2: [I and my girlfriend → My girlfriend and] looked [∅ → at a] picture called " " (Immaculate Conception).

Gold Correction 3: If he [was a → were] Japanese, I suppose I [replied → would reply] like this: " 800m".

Proposed Correction 3: If he was [a → ∅] Japanese, I suppose I [replied → would reply] like this: " 800m".

4.3 Inference Tweaking and Error Thresholds

The inference tweaking phase was crucial in tun- ing the balance between precision and recall. The changes made here, particularly lowering the mini- mum error thresholds before the model makes an edit, indicated a clear attempt to force the model to make more corrections.[10] While this slightly lowered precision on monolingual errors, it signifi- cantly enhanced the performance on CSW text.

To determine that the improved performance of our proposed model was not entirely due to the different inference configuration, we conducted a similar grid search for the existing GECToR model. However, instead of using the Stage 3 validation dataset, as we did with our model, we used the CSW test set directly. The highest $F_{0.5}$ achieved by the baseline model was 56.46, providing evidence that our proposed model beats all inference con- figurations of the previous GECToR system when applied to CSW texts.

4.4 Synthetic Data Impact

The synthetic CSW text and error injection meth- ods were central to this project. The resemblance of our synthetic text to real ESL learner data, as shown by the similarity metrics in Table 1, is a testament to the effectiveness of our chosen gen- eration method. The improvements in $F_{0.5}$ scores provide further evidence of this.

Our extended PIE-synthetic dataset aimed to in- troduce four error types common in ESL students: noun, pronoun, punctuation and word errors. When compared to the monolingual GECToR, our model is stronger in all of these areas.[11] This provides strong evidence that the targeted approach to error injection was successful in boosting the model's ability in these areas.

5 Conclusion

The primary aim of this paper was to build a GEC system capable of effectively correcting English errors in CSW text, whilst maintaining competi- tive performance on monolingual data. To address the scarcity of CSW data, we explored methods of generating synthetic CSW text. We used several CSW metrics to establish that the LLM prompting- based approach was the most capable

---

[10]Implementation details are presented in Appendix D.
[11]Error type analysis of our model is given in Appendix E.

of generating text resembling the content in our genuine dataset. From there, we used two error injection methods to create the first substantial datasets labelled for CSW GEC. This significantly expanded the train- ing data available. Importantly, it also opened up opportunities for future research in CSW GEC and CSW NLP more generally. We demonstrated the efficacy of our synthetic data generation techniques by training the first GEC model aimed at correcting errors in CSW texts. Our model showed a clear improvement in performance on CSW data, sur- passing the SOTA in this area.

6   Limitations

This research, while comprehensive, encounters several limitations that highlight areas for potential improvement. One primary limitation lies in the overrepresentation of Japanese in the genuine CSW dataset. This raises questions about the model's ap- plicability to a broader range of language pairs.

This is an unfortunate consequence of using a dataset sourced from Lang-8, a Japanese language learning network. Although our method demon- strated that it could generate texts from a wider range of language pairs, it is possible that all CSW data used shows a bias towards Japanese styles of CSW. Such a bias in our system could inadvertently lead to reduced accessibility and effectiveness for ESL learners who CSW with languages other than Japanese. If this system were to be used as an aid in ESL education, steps should be taken to ensure that it does not contribute to existing inequalities present in language learning platforms. English learning tools should be accessible regardless of the student's native language and future work should focus on developing more inclusive datasets to help mitigate these risks.

Another possible limitation relates to the style of model chosen. The sequence tagging method was selected due to its lower data requirements, but this decision may have constrained the capabilities of the model. Many of the errors typical of ESL students require complex restructuring of the sentence — a notably difficult task for edit-based GEC systems. Although the data needs are more substantial, it is likely that NMT GEC systems may fare better as they are not constrained by a limited vocabulary of edits.

To assess the likeness of our generated CSW text, we introduced several common CSW metrics. Although useful, these metrics are not very sophis- ticated, and often struggle to accurately capture the nuances of CSW patterns across different sub- populations. These language patterns can have a substantial impact on the optimal approaches to problems across CSW NLP, and hence, the field would benefit from further research in this area. Ideally, we would have conducted a human study to evaluate the quality of our synthetic data. How- ever, given the constraints of the project, it was not possible, and we acknowledge this as a limitation of our work.

Finally, we reported results for a RoBERTa- base GECToR system. Although we also tested other base models, including BERT, DeBERTa and ELECTRA, we did not look at larger models or en- semble systems. Future extensions could explore this area, building upon the observation that larger models or simple voting ensembles can yield better results than the smaller base models (Tarnavskyi et al., 2022).

In summary, whilst the current work makes sig- nificant contributions to the field of GEC for CSW text, these limitations indicate crucial areas for fur- ther research and development.

References

[1] Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence trans- duction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natu- ral Language Processing (EMNLP-IJCNLP)*, pages 4259–4269, Hong Kong, China. Association for Com- putational Linguistics.

[2] Ruthanna Barnett, Eva Codó, Eva Eppler, Montse For- cadell, Penelope Gardner-Chloros, Roeland van Hout, Melissa Moyer, Maria Carme Torras, Maria Teresa Turell, Mark Sebba, Marianne Starren, and Sietse Wensing. 2000. The lides coding manual: A doc- ument for preparing and analyzing language inter- action data version 1.1—july, 1999. *International Journal of Bilingualism*, 4(2):131–132.

[3] Anne L Beatty-Martínez, Christian A Navarro-Torres, and Paola E Dussias. 2020. Codeswitching: A bilin- gual toolkit for opportunistic speech planning. *Fron- tiers in Psychology*, 11:1699.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

[5] Christopher Bryant, Mariano Felice, Øistein E. Ander- sen, and Ted Briscoe. 2019. The BEA- 2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Lin- guistics.

[6] Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceed- ings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

[7] Kelvin Wey Han Chan, Christopher Bryant, Li Nguyen, Andrew Caines, and Zheng Yuan. 2024. Grammati- cal error correction for code-switched sentences by learners of English. In *Proceedings of the 2024 Joint International Conference on Computational Linguis- tics, Language Resources and Evaluation (LREC- COLING 2024)*, pages 7926–7938, Torino, Italia. ELRA and ICCL.

[8] Ciprian Chelba, Tomás Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *CoRR*, abs/1312.3005.

[9] Daniel Dahlmeier, Hwee Tou Ng, Siew Mei Wu, et al. 2023. Nus corpus of learner english (nucle). National University of Singapore, NLP Group. Available: https://www.comp.nus.edu.sg/nlp/corpora.html.

[10] Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Lin- guistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

[11] Arianna Falbo and Travis LaCroix. 2021. Est-ce que vous compute? code-switching, cultural identity, and AI. *CoRR*, abs/2112.08256.

[12] P.J. Finlay. 2023. Argos translate. Open-source offline translation library written in Python.

[13] Björn Gambäck and Amitava Das. 2016. Comparing the level of code-switching in corpora. In *Proceed- ings of the Tenth International Conference on Lan- guage Resources and Evaluation (LREC'16)*, pages 1850–1855, Portorož, Slovenia. European Language Resources Association (ELRA).

[14] Souvick Ghosh, Satanu Ghosh, and Dipankar Das. 2017. Complexity metric for code-mixed social media text. *Computación y Sistemas*, 21(4):693–701.

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

[16] Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E. Bullock, and Almeida Jacque- line Toribio. 2017. Metrics for Modeling Code-Switching Across Corpora. In *Proc. Interspeech 2017*, pages 67–71.

[17] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senel- lart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the As- sociation for Computational Linguistics (ACL 2017), System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

[18] Philipp Koehn et al. 2012. European parliament pro- ceedings parallel corpus.

[19] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language pro- cessing toolkit. In *Proceed- ings of 52nd Annual Meet- ing of the Association for Computational Linguis- tics: System Demonstrations*, pages 55–60, Balti- more, Maryland. Association for Computational Lin- guistics.

[20] Tomoya Mizumoto, Toshikazu Tajiri, Takuya Fujino, Seiji Kasahara, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2013. NAIST Lang-8 Learner Corpora. Language Learner Corpora compiled from Lang-8 SNS. Available for research and educational purposes.

[21] Li Nguyen, Zheng Yuan, and Graham Seed. 2022. Building educational technologies for code-switching: Current practices, difficulties and future directions. *Languages*, 7:220.

[22] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR — grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

[23] Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code- mixed text. In *Proceedings of the*

Figure 3: An Example LLM Prompt Used to Generate CSW Text

16th Conference of the European Chapter of the Association for Compu- tational Linguistics: System Demonstrations, pages 205–211, Online. Association for Computational Lin- guistics.

[24] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebas- tian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the As- sociation for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

[25] 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 Build- ing Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

[26] Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In *Accepted for publication at 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, Ireland.

[27] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

[28] W. Quin Yow, Jessica S. H. Tan, and Suzanne Flynn. 2018. Code-switching as a marker of linguistic com- petence in bilingual children. *Bilingualism: Lan- guage and Cognition*, 21(5):1075–1090.

A   Example LLM Prompt

Figure 3 shows an example LLM prompt used to generate synthetic CSW sentences from genuine examples. As we are using a private subset of the Lang-8 dataset, we are not permitted to share any of the CSW texts.

Settings: [no prose]

For each of the following code-switched sentences, generate a new sentence that uses the same two languages and a similar style of code-switching. The topic should be different. Ensure you use the correct grammar in the English portion of the sentence. Make sure that each sentence contains 2 languages. Only return the sentences and their number. You must follow all of the instructions.

For example, given the source sentence and label:

    1. This food is called "" .

An acceptable answer would be:

    1. This animal is called a "" .

Do not include any other information in the generated sentences. The 10 real examples are as follows:

    1. [CSW SENTENCE]
2. [CSW SENTENCE]

Table 3: $F_{0.5}$ Scores, TP, FP, FN, and Differences in $F_{0.5}$ Scores (BEA - CSW) for Different Categories in the BEA-2019 Test Split and our Genuine CSW Dataset.

Category	BEA-2019 Test				Genuine CSW		
	$F_{0.5}$	TP	FP	FN	$F_{0.5}$	TP	
80.45	432	80	205	46.27	472	351 NOUN	
29	16	94	4.34	21	147 ORTH	75.96	
30	198	36.45	181	264 OTHER	39.51	113	
557	3.55	39	241 PREP	75.44	263	58	
39.14	241	251 PRON	66.38	62	19	81	
21	32 PUNCT	80.93	786	165	266	0.35	
286 VERB	52.59	61	27	167	18.08	49	7
81.62	151	30	50	37.20	61	90 VERB:SVA	
128	14	26	57.58	114	79 VERB:TENSE	65.55	
62	133	33.80	116	172 WO	58.08	23	
63	6.33	2	11 height				

FN (Genuine CSW): DET 1336; NOUN 1725; ORTH 522; OTHER 4333; PREP 870; PRON 274; PUNCT 284; VERB 802; VERB:FORM 155; VERB:SVA 104; VERB:TENSE 448; WO 104.

3. [CSW SENTENCE]
4. [CSW SENTENCE]
5. [CSW SENTENCE]
6. [CSW SENTENCE]
7. [CSW SENTENCE]
8. [CSW SENTENCE]
9. [CSW SENTENCE]
10.[CSW SENTENCE]

B Error Type Analysis of SOTA

Table 3 shows a breakdown of the performance of a single RoBERTa Large-based GECToR system trained purely on monolingual GEC data when ap- plied to two datasets, our genuine CSW dataset and the BEA-2019 (Bryant et al., 2019) test set. These datasets are approximately the same size. The model used represents a current near-SOTA single model sequence tagging-based GEC system measured using $F_{0.5}$ on the BEA-2019 test set. For brevity, we have removed categories with a low number of examples in either dataset or where per- formance is not significantly different.

C Training Data Schedule

In this section, we explicitly detail the data used at each stage of the training process.

C.1 Stage 1

For the initial pre-training stage, we used the distilled dataset proposed by the SOTA (Tar- navskyi et al., 2022). This dataset was constructed by extracting corrections from the monolingual 1BW corpus (Chelba et al., 2013) using the high- est performing GECToR ensemble. Through this dataset, we shuffled our PIE-synthetic CSW dataset. We deemed this dataset to be of lower quality than its

Table 4: Sentence Count and Contribution of Stage 2 Datasets

Dataset	Sentences heightLang-8
985,683 (80.54W	I Locness
68,608 (5.61NUCLE	54,258 (4.43FCE
26,929 (2.20Syn-CSW PIE	70,181 (5.73Syn-CSW Rev-GECToR
18,160 (1.48heightTotal	1,223,819 height

Table 5: Sentence Count and Contribution of Stage 3 Datasets

Dataset	Sentences heightW
I Locness	68,608 (67.23Syn-CSW Rev-GECToR
18,160 (17.80Syn-CSW PIE	10,000 (9.80CSW Genuine
5,279 (5.17heightTotal	102,047 height

Rev-GECToR counterpart. Consequently, it was used earlier in the training process. This provided roughly 1,200,000 examples for the initial training phase of which we split between train and validation sets according to a ratio of 19:1. Our synthetic CSW sentences comprised approximately 5.65of this dataset. We aimed to keep this percent- age small in this phase of the training process to allow the model to first learn to correct errors in monolingual texts. In later stages, we boosted the contribution of the CSW data.

C.2 Stage 2

For the second stage, we shuffled several GEC datasets. These are NUCLE (Dahlmeier et al., 2023), FCE (Yannakoudakis et al., 2011), WI Locness (Bryant et al., 2019), Lang-8 (Mizumoto et al., 2013) and our 2 newly created CSW datasets. As our genuine CSW dataset is a subset of the pri- vate Lang-8 corpus, we checked and removed any duplicates. Table 4 shows the overall contributions of each corpus towards the stage 2 dataset. Similar to the previous stage, the data was split into train and validation sets.

C.3 Stage 3

For the final stage, we combined the high quality WI Locness dataset with a sampled subset of the genuine CSW data and a sampled subset of the synthetic CSW texts. Again, the stage 3 dataset is split into train and validation sets. The remaining unused subset of the genuine CSW dataset was retained for testing purposes. Table 5 details the contributions to this stage from each dataset.

D Inference Hyperparameters

After training our model, we used the validation dataset from stage 3 to tune 2 inference parameters. These are:

- **additional**$_confidence$ $---This value is added to the probability of the$ KEEP $token. If this value is high, recall is lil$

- **min**$_error_probability$ $---For a change to be made to a sentence, the probability of at least one token in the sentence l$

Table 6: Precision (P), Recall (R) and $F_{0.5}$ Score of Our Proposed Model for Targeted Error Types in the Genuine CSW Test Dataset

Category	P	R	$F_{0.5}$ heightNOUN
0.2857	0.0833	0.1923 PRON	0.7647
0.4643	0.6771 PUNCT	0.7143	0.1139
0.3460 WO	0.7778	0.2000	0.4930 height

E   Error Type Analysis of Proposed Model

By exploring the ERRANT error classifications of our proposed model when applied to the CSW test dataset, we can further explore the effectiveness of our synthetic data in addressing the problematic areas identified in Appendix B. A breakdown of the precision, recall and $F_{0.5}$ score for each of the previously identified categories is shown in Table 6.