LexiLogic@CALCS 2025: Predicting Preferences in Generated Code-Switched Text

Pranav Gupta[*], **Souvik Bhattacharyya**[*], **Niranjan Kumar M, Billodal Roy**
Lowe's
Correspondence: {pranav.gupta, souvik.bhattacharyya, niranjan.k.m, billodal.roy}@lowes.com

Abstract

Code-switched generation is an emerging application in NLP systems, as code-switched text and speech are common and natural forms of conversation in multilingual communities worldwide. While monolingual generation has matured significantly with advances in large language models, code-switched generation still remains challenging, especially for languages and domains with less representation in pre-training datasets. In this paper, we describe our submission to the shared task of predicting human preferences for code-switched text in English-Malayalam, English-Tamil, and English-Hindi. We discuss our various approaches and report on the accuracy scores for each approach.

1 Introduction

Code-switching, the act of alternating between two or more languages or language varieties within the same utterance or conversation, is an everyday phenomenon in multilingual communities throughout the world (Myers-Scotton, 1993). Traditional text corpora lack sufficient code-switched data, because code-switching is typically viewed as something informal and considerable care is taken to remove foreign words in monolingual corpora (Sitaram et al., 2020). However, with the emergence of new internet users across the world who engage in written and verbal code-switched communication along with code-switched user content on social media platforms, generating and understanding code-switched content has become more relevant than ever before. Contrary to normal belief, large language models (LLMs) are not yet fully capable of understanding and generating code-switched speech (Winata et al., 2021; Zhang et al., 2023).

Another important and often overlooked aspect is evaluation metrics for code-switched generations. While there have been efforts on evaluating the abilities of NLP systems on code-mixed text, (Khanuja et al., 2020) there have been much fewer studies on rating code-mixed text generations. Existing metrics might not be general enough or up to date with current societal and linguistic trends. Metrics to rate model-based generation of synthetic code-mixed data have mostly relied on methods suitable for monolingual text, such as chrF (Popović, 2015) and COMET (Rei et al., 2020). Robust evaluation metrics for code-switched generations can in turn help in post-training and optimizing LLMs for applications that require code-switched generation. In this paper, we explore approaches for predicting human preferences on pairs of code-switched generations (Kuwanto et al., 2024) and report accuracy metrics.[1]

2 Related Work

While there have been fewer efforts on predicting human preferences in code-switched text, we review two closely related themes: metrics for evaluating NLP systems on code-switched data, and metrics for predicting human preferences on model-generated text.

2.1 Metrics for evaluating code-switching

Two of the most popular recent benchmarks for evaluating model performance on code-switched text are GlueCOS (Khanuja et al., 2020) and LinCE (Aguilar et al., 2020). There has also been some effort in automated evaluation methods, such as Guzmán et al. (2017). With the rise of general-purpose LLMs, LLM-based evaluation metrics are also being increasingly explored for evaluating the capabilities of NLP systems to work with code-switched text. Correlation of such automated metrics with human judgment, however, is a major chal-

[*]These authors contributed equally to this work.

[1]The code repository for our models can be found at: https://github.com/souvikshanku/CALCS-2025/.

Proceedings of the 7th Workshop on Computational Approaches to Linguistic Code-Switching, pages 48–53
May 3, 2025 ©2025 Association for Computational Linguistics

lenge. Moreover, given the highly context-specific and complex nature of code-switching, linguistically motivated approaches such as intonation units (Pattichis et al., 2023) and equivalence constraint theory (Kuwanto et al., 2024) have also been important considerations in defining metrics for code-switched text.

2.2 Aligning automated evaluation metrics with human preferences

While traditional automated evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and chrF (Popović, 2015), along with newer metrics based on LLMs (Zheng et al., 2023) are widely used in NLP, aligning them with human metrics is challenging. Recent efforts such as COMET (Rei et al., 2020) and MetaMetrics (Anugraha et al., 2024) have focused on this issue.

3 Dataset

We use the labeled component of the CSPref dataset (Kuwanto et al., 2024), and split it into a train set and a test set. While there are 62613 rows in the dataset, there are only 403 unique (original_l1, original_l2) pairs. In order to avoid leakage between our train and test splits, we split based on unique (original_l1, original_l2) pairs and randomly choose 30 of the unique (original_l1, original_l2) pairs for the test set. This resulted in 50373 and 12240 rows in the train and test splits respectively. All the corresponding rows were then assigned to either the train or test set based on the corresponding split of (original_l1, original_l2). The final evaluations happen on a separate holdout test set.[2] Relevant columns in the initial labeled dataset were as follows:

- original_l1: original sentence in language 1
- original_l2: original sentence in language 2
- sent_1: code-switched generation 1
- sent_2: code-switched generation 2
- chosen: whether sent_1 or sent_2 is a better generation. This could have 3 values- "sent_1", "sent_2", and "tie."
- lang: language pair used for code-switching (English-Hindi, English-Malayalam, English-Tamil)

The goal of the task is to use the other columns to predict the label, i.e., the values in the "chosen" column. In our models we chose not to use the "lang" column as a feature, due to the possibility of using our models to evaluate on data from unseen language pairs.

The details of the initial dataset before our train-test split are given in Table 1.

4 Model Experiments

4.1 Finetuning GPT-2

GPT-2 has been used as a reward model for aligning large language models (LLMs) with human preferences in the past, making it a promising opportunity for us to conduct experiments on this model for the code-switching task.

Following (Stiennon et al., 2022), (Ouyang et al., 2022), we utilize the base GPT-2 model as a reward model by removing the unembedding layer and attaching a randomly initialized linear head that outputs a scalar value, which can be interpreted as the score GPT-2 assigns to the input. For each datapoint, we construct pairs of reference sentences and code-switched texts, obtaining two rewards, r_1 and r_2. During training, we aim to maximize the reward for the better code-switched completion. This is achieved by concatenating the two rewards and then applying the softmax function. As a result, we use the cross-entropy loss as our loss function to minimize during the optimization process. In the dataset, we effectively have three "classes": whether one of the two given sentences was preferred by the human raters, or if there was a tie between them. To adapt to this three-class classification problem, during training, in the case of a tie, we randomly assign one of the sentences as the preferred sentence. This approach is fundamentally inspired by the Bradley-Terry model (Bradley and Terry, 1952).

$$\begin{aligned}\text{loss}(r_\theta) = &- E_{(x,y_0,y_1,i)\sim D} \\ &\left[\log\left(\sigma\left(r_\theta(x, y_i) - r_\theta(x, y_{1-i})\right)\right)\right],\end{aligned}$$

where r is the reward model parameterized by θ, x is the reference input, (y_0, y_1) are the two code-switched completions, and i denotes the preferred completion selected by the human rater.

While evaluating our trained model, we obtain the model outputs, i.e., the probability values after applying softmax, and then determine if it's a tie by checking whether the absolute difference between the two values is below a specified threshold. This threshold is selected to maximize the macro F1 score on the held-out validation set. We observe

[2] Our submission model achieved a public leaderboard score of 1.00 and a private leaderboard score of 0.46, and can be found on the Kaggle shared task leaderboard.

Label \ Lang pair	Eng-Hin	Eng-Mal	Eng-Tam	Overall
sent_1	8866	7995	5955	22816
sent_2	8951	8136	5973	23060
tie	3486	4524	8727	16737
Total	21303	20655	20655	62613

Table 1: Dataset details of the CSPref dataset

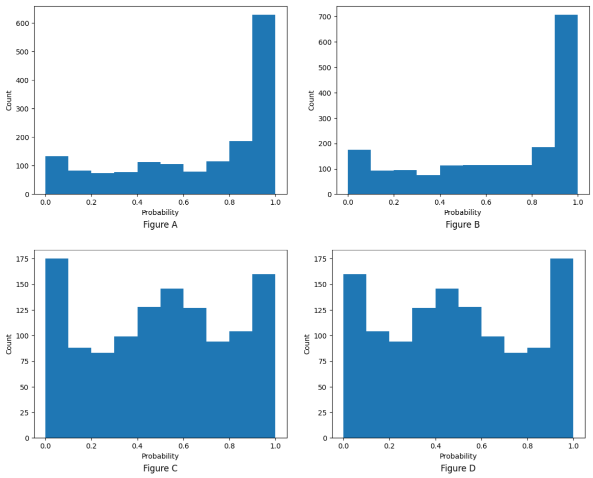

Figure 1: **Fig A**: Probability of sent_1 being preferred when actually sent_1 is chosen. **Fig B**: Probability of sent_2 being preferred when actually sent_2 is chosen. **Fig C**: Probability of sent_1 being preferred when there is a tie. **Fig D**: Probability of sent_2 being preferred when there is a tie.

that when the model is confident about the quality of an input, its value is at either end, but when there is a tie, the score tends to fluctuate unpredictably as can be seen in Figure 1.

The provided dataset contained three language pairs. To validate if cross-lingual transfer occurs during the learning process for rating code-switched texts, we trained and evaluated our model three times. Initially, we trained it only on English-Hindi pairs, then on English-Hindi and English-Tamil pairs, and finally on all three language pairs.

We provide our training hyperparameters and the obtained results in the following section.

Parameter	Value
Learning rate	3e-5
Learning rate decay	0.9
Batch size	14
Grad. Acc. Steps	2
Training epochs	5

Table 2: Training hyperparameters for GPT2-based RM

Table 3 summarizes the accuracy metrics obtained from our experiments with GPT-2. When we trained our model exclusively on code-switched texts of English-Hindi pairs, we achieved moderate performance in English-Hindi and slightly lower performance in English-Tamil and English-Malayalam pairs. However, when we extended our training set by including more language pairs, we observed an overall increase in performance.

4.2 Logistic regression on top of multilingual embeddings

In this approach, we trained a 3-class logistic regression model on top of multilingual embeddings of the concatenation of original_l1, original_l2, sent_1, and sent_2, using the one-versus-rest approach. The prediction is defined as:

$$\arg\max_i \sigma(w_i.x(concat[s_1, s_2, s_3, s_4])),$$

where $i \in \{sent\_1, sent\_2, tie\}$, w_i denotes the weight of the i-versus-rest classifier, x(.) denotes the embedding transformation, and s_1, s_2, s_3, s_4 denote the strings corresponding to original_l1, original_l2, sent_1, and sent_2. For the embedding model, we chose Cohere `embed-multilingual-v3.0`, given its ease of use, strong performance on the MTEB benchmark (Muennighoff et al., 2023), and coverage of over 100 languages. This model has an accuracy of 0.69 and 0.52 on the train and test sets respectively.

4.3 Fasttext classification

Fasttext (Bojanowski et al., 2017) is an efficient tool which provides strong baseline performance in text classification, without relying on large pre-trained language models. We train a 3-class classification model on concatenated original_l1, original_l2, sent_1, and sent_2 with default parameters, i.e., learning rate of 0.1, 100-dimensional word vectors, a context window of size 5, 5 epochs, and a negative sampling size of 5. The training and test accuracies for the Fasttext classification model are shown in Table 4.

Trained On			Test Set Accuracy		
Eng-Hin	Eng-Tam	Eng-Mal	Eng-Hin	Eng-Tam	Eng-Mal
✓	-	-	**0.47**	0.41	0.46
✓	✓	-	0.41	0.56	0.45
✓	✓	✓	0.42	**0.60**	**0.56**

Table 3: Accuracy obtained after finetuning GPT-2

Data split \ Lang pair	Eng-Hin	Eng-Mal	Eng-Tam	Overall
Train	0.69	0.67	0.76	0.71
Test	0.37	0.40	0.38	0.38

Table 4: Accuracy obtained for the train and test splits of the CSPref dataset

4.4 GPT-4o

Given the higher correlation with human judgment scores when using GPT-4o (Kuwanto et al., 2024) when compared with other metrics to judge the quality of code-mixed generations in the CSPref dataset, we chose to use GPT-4o to decide between "sent_1," "sent_2," and "tie." Our instruction message to GPT-4o gave it an approximate prior of an equal distribution of "sent_1," "sent_2," and "tie," and additionally explained the process of choosing a certain label. In order to speed up the inference process, we batched dataset rows before sending them to GPT-4o for preference prediction. We experimented with various batch sizes and found a batch size of 20 to be a good compromise between speed and accuracy.

Model	Test Set Accuracy
Finetuned GPT-2	0.53
Cohere Embeddings + Logistic Regression	0.52
FastText	0.38
GPT-4o	**0.66**

Table 5: Train and test set accuracies of all the models

4.5 Results

The summary of our model accuracy scores is given in Table 5. We observed that GPT-4o does the best among all the models we tried for this task. With a larger training set of human preferences with a more diverse collection of language pairs, it might be easier to finetune larger models to capture human preferences better. During our exploratory data analysis and verification with native Hindi speakers, we also found that some of the sentences lacked coherence, which could be due to the fact that they were generated from smaller LLMs such as Llama. Note that we do not use the language pair as a feature or train different models for different language pairs.

5 Conclusion

In this paper, we experimented with various models to predict human preferences among candidate code-switched generations in English-Hindi, English-Malayalam, and English-Tamil. We observed that GPT-4o does the best among the various models we tried. Future work might explore the use of bigger models and datasets, and also a deeper comparative analysis between the variations across languages. For LLM-based approaches, we could also explore prompt optimization using tools such as DSPy (Khattab et al., 2024) and parameter-efficient finetuning methods such as LoRA (Hu et al., 2021) and its derivatives. Another interesting direction is to explore the effectiveness of these models to act as reward functions for aligning LLMs to generate more natural code-mixed text.

6 Limitations

While predicting human preferences is a crucial step in generating natural and accurate code-mixed text, we need to consider the ethical implications of such models, especially in case they are used in real world applications in multilingual communities such as e-commerce, governance, health care, and education. Underrepresented or misrepresented aspects in a preference dataset can propagate biases. Communities that code-switch in a unique, uncommon way might feel disenfranchised if these models cannot capture human preferences accurately. Moreover, we need to consider whether correlations between metrics and human judgment are a sufficient benchmark for comparing various models.

References

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

David Anugraha, Garry Kuwanto, Lucky Susanto, Derry Tanti Wijaya, and Genta Winata. 2024. MetaMetrics-MT: Tuning meta-metrics for machine translation via human preference calibration. In *Proceedings of the Ninth Conference on Machine Translation*, pages 459–469, Miami, Florida, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Preprint*, arXiv:1607.04606.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324.

Gualberto Guzmán, Joseph Ricard, Jacqueline Serigos, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2017. Metrics for modeling code-switching across corpora. In *Interspeech 2017*, pages 67–71.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2024. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*.

Garry Kuwanto, Chaitanya Agarwal, Genta Indra Winata, and Derry Tanti Wijaya. 2024. Linguistics theory meets llm: Code-switched text generation via equivalence constrained large language models. *Preprint*, arXiv:2410.22660.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Proceedings of the Workshop on Text Summarization Branches Out*, pages 74–81.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Carol Myers-Scotton. 1993. *Duelling Languages*. Clarendon Press, Oxford, England.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Rebecca Pattichis, Dora LaCasse, Sonya Trawick, and Rena Cacoullos. 2023. Code-switching metrics using intonation units. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16840–16849, Singapore. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2020. A survey of code-switched speech and language processing. *Preprint*, arXiv:1904.00784.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. *Preprint*, arXiv:2009.01325.

Genta Indra Winata, Samuel Cahyawijaya, Zihan Liu, Zhaojiang Lin, Andrea Madotto, and Pascale Fung. 2021. Are multilingual models effective in code-switching? In *Proceedings of the Fifth Workshop* *on Computational Approaches to Linguistic Code-Switching*, pages 142–153, Online. Association for Computational Linguistics.

Ruochen Zhang, Samuel Cahyawijaya, Jan Christian Blaise Cruz, Genta Winata, and Alham Fikri Aji. 2023. Multilingual large language models are not (yet) code-switchers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12567–12582, Singapore. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.