HinglishEval Generation Challenge on Quality Estimation of Synthetic Code-Mixed Text: Overview and Results

Vivek Srivastava
TCS Research
Pune, Maharashtra, India
srivastava.vivek2@tcs.com

Mayank Singh
IIT Gandhinagar
Gandhinagar, Gujarat, India
singh.mayank@iitgn.ac.in

Abstract

We hosted a shared task to investigate the factors influencing the quality of the code-mixed text generation systems. The teams experimented with two systems that generate synthetic code-mixed Hinglish sentences. They also experimented with human ratings that evaluate the generation quality of the two systems. The first-of-its-kind, proposed subtasks, *(i) quality rating prediction* and *(ii) annotators' disagreement prediction* of the synthetic Hinglish dataset made the shared task quite popular among the multilingual research community. A total of 46 participants comprising 23 teams from 18 institutions registered for this shared task. The detailed description of the task and the leaderboard is available at https://codalab.lisn.upsaclay.fr/competitions/1688.

1 Introduction

Code-mixing is the phenomenon of mixing words and phrases from multiple languages in a single utterance of a text or speech. Figure 1 shows the example code-mixed Hinglish sentences generated from the corresponding parallel Hindi and English sentences. Code-mixed languages are prevalent in multilingual communities such as Spain, India, and China. With the inflation of social-media platforms in these communities, the availability of code-mixed data is seeking a boom. It has lead to several interesting research avenues for problems in computational linguistics such as language identification (Singh et al., 2018; Shekhar et al., 2020), machine translation (Dhar et al., 2018; Srivastava and Singh, 2020), language modeling (Pratapa et al., 2018), etc.

Over the years, we have observed various computational linguistic conferences and workshops organizing the shared tasks involving code-mixed languages. Diverse set of problems have been hosted such as sentiment analysis (Chakravarthi et al., 2021; Patwa et al., 2020), offensive language identification (Chakravarthi et al., 2021), word-level language identification (Solorio et al., 2014; Molina et al., 2016), information retrieval (Banerjee et al., 2016), etc.

Example I

HINGLISH: ye ek code mixed sentence ka example hai
ENGLISH : this is an example code-mixed sentence

Example II

HINGLISH : kal me movie dekhne ja raha hu. How are the reviews?
ENGLISH: I am going to watch the movie tomorrow. How are the reviews?

Figure 1: Example parallel Hinglish and English sentences. The code-mixed Hinglish sentences contain words from Hindi and English languages.

Despite these overwhelming attempts, the natural language generation (NLG) and evaluation of the code-mixed data remain understudied. The noisy and informal nature of the code-mixed text adds to the complexity of solving and evaluating the various NLG tasks such as summarization and machine translation. These inherent challenges (Srivastava and Singh, 2020) with the code-mixed data makes the widely popular evaluation metrics like BLEU and WER obsolete. Various metrics (e.g., CMI (Das and Gambäck, 2014; Gambäck and Das, 2016), M-index (Barnett et al., 2000), I-index (Guzmán et al., 2017), Burstiness (Goh and Barabási, 2008), Memory (Goh and Barabási, 2008), etc.) have been proposed to measure the complexity of code-mixed data, but they fail to capture the linguistic diversity which leads to poorly estimating the quality of code-mixed text (Srivastava and Singh, 2021a).

In this shared task[1] (see Section 2) for the de-

[1] https://sites.google.com/view/hinglisheval

tailed description), we look forward to the new strategies that cater to the broad requirement of the quality evaluation of the generated code-mixed text. These methods will entail various linguistic features encompassing syntax and semantics and the perspectives of human cognition such as writing style, emotion, sentiment, language, and preference. We also put forward a subtask to understand the factors influencing the human disagreement on the quality rating of the generated code-mixed text. This could help design a more robust quality evaluation system for the code-mixed data.

A total of 46 participants comprising 23 teams from 18 institutions registered for this shared task. Out of which four teams have submitted their final reports. Section 3 presents the overview of the participants and submission methodology. Section 4 compares the four submissions and presents discussions around the similarity and differences of the approaches. We conclude and present future directions in Section 5.

2 The HinglishEval Shared Task

In this shared task, we propose two subtasks to evaluate the quality of the code-mixed Hinglish text. First, we propose to predict the quality of Hinglish text on a scale of 1–10. We aim to identify the factors influencing the text's quality, which will help build high-quality code-mixed text generation systems. We synthetically generate the Hinglish sentences using two different approaches (see Section 2.1) leveraging popular English-Hindi parallel corpus. Besides, we also have at least two human-generated Hinglish sentences corresponding to each parallel sentence. The second subtask aims to predict the disagreement on a scale of 0–9 between the two annotators who have annotated the synthetically generated Hinglish sentences. Various factors influence this human disagreement, and we seek to investigate the reasoning behind this behavior.

2.1 Dataset

As outlined in Section 1, the code-mixed NLG task observes a scarcity of high-quality datasets. Consequently, the quality evaluation of the generated code-mixed text remains unexplored. To address this challenge, we propose a new dataset with Hinglish sentences generated synthetically and rated by human annotators. We create the dataset in two phases.

Phase 1: Human-generated Hinglish sentences: We select the English-Hindi parallel sentences from the IIT-B parallel corpus (Kunchukuttan et al., 2018) to generate the Hinglish sentences. The parallel corpus has 1,561,840 sentence pairs. We randomly select 5,000 sentence pairs in which the number of tokens in both the sentences is more than five. We employ five human annotators and assign each 1,000 sentence pairs. Table 1 shows the annotation guidelines to generate the Hinglish sentences. Post annotation, we obtain 1,976 sentence pairs for which the annotators have generated at least two Hinglish sentences.

Phase 2: Synthetic Hinglish sentence generation and quality evaluation: We synthetically generate the Hinglish sentence corresponding to each of the parallel 1,976 English-Hindi sentence pairs. We employ two different code-mixed text generation (CMTG) techniques:

- Word-aligned CMTG (WAC): Here, we align the noun and adjective tokens between the parallel sentences. We replace the aligned Hindi token with the corresponding English token and transliterate the Hindi sentence to the Roman script.
- Phrase-aligned CMTG (PAC): Here, we align the key-phrases of length up to three tokens between the parallel sentences. We replace the aligned Hindi phrase with the corresponding English phrase and transliterate the Hindi sentence to the Roman script.

For the token alignment between parallel sentences, we use the online curated dictionaries, GIZA++ (Och and Ney, 2003) trained on the remaining IIT-B corpus, and cross-lingual word embedding trained on English and Hindi word vectors from FastText (Bojanowski et al., 2017). We employ eight human annotators[2] to provide a rating between 1 (low quality) to 10 (high quality) to the generated Hinglish sentences. Table 1 shows the annotation guidelines to rate the sentences. Figure 2a and 2b shows the distribution of the annotators' rating and their disagreement, respectively.

Data format: Table 2 shows an instance from the dataset. In total, we have 3,952 instances[3] in the dataset where each data instance i for subtask-1 (see Section 2.2.1) is represented as $\mathbf{X1}_i$={Eng$_i$, Hin$_i$, Synthetic_Hing$_i$} and $\mathbf{Y1}_i$={Average_rating$_i$}.

[2]Different from the annotators in Phase 1. Each annotator gets 247 sentences generated by PAC and WAC, each corresponding to the same set of parallel sentences.

[3]Two synthetic Hinglish sentences are generated for each parallel sentence pair.

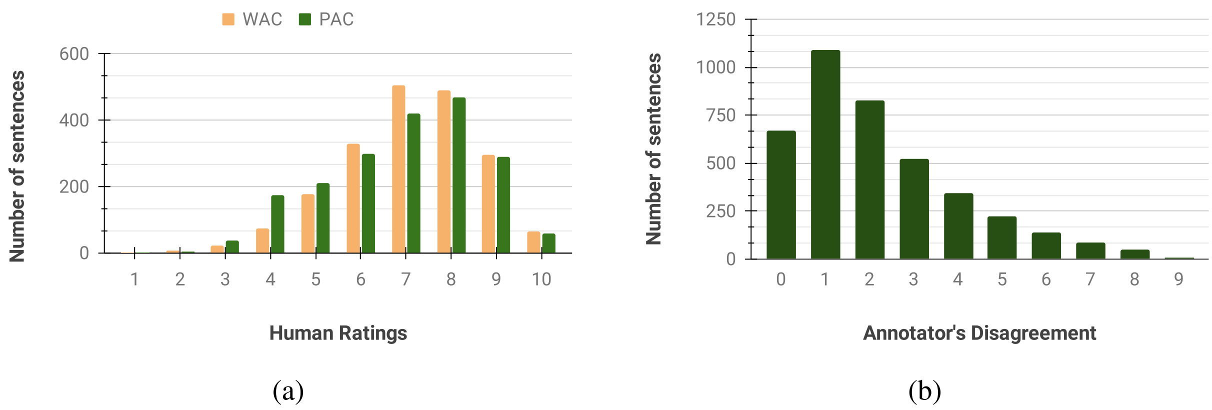

Figure 2: Distribution of (a) human evaluation scores and (b) disagreement in human scores in the synthetically generated Hinglish sentences.

Task	Guidelines
Hinglish text generation	1. The Hinglish sentence should be written in Roman script. 2. The Hinglish sentence should have words from both the source languages. 3. Avoid using new words, wherever possible, that are not present in both sentences. 4. If the source sentences are not the translation of each other, mark the sentence pair as "#".
Quality rating	The rating depends on the following three factors: 1. The similarity between the generated Hinglish sentence and the source sentences. 2. The readability of the generated sentence. 3. The grammatical correctness of the generated sentence.

Table 1: Annotation guidelines to the annotators for the two different tasks.

For subtask-2 (see Section 2.2.2), the instance j is represented as $\mathbf{X2}_j$={Eng_j, Hin_j, Synthetic_Hing_j} and $\mathbf{Y2}_j$={$\text{Annotator\_disagreement}_j$}. In addition, we provide at least two human generated Hinglish sentences corresponding to each data instance for both the subtasks. We shuffle and split the dataset in the ratio 70:10:20 with 2766, 395, and 791 data instances in train, validation, and test respectively. The more detailed description of the dataset is available in (Srivastava and Singh, 2021b).

2.2 The Two Sub-tasks

2.2.1 Subtask 1: Quality rating prediction

The first subtask is predicting the quality rating of the code-mixed text. The participating teams can use the English, Hindi, and human-generated Hinglish sentences to predict the average rating[4] as provided by the human annotators to the synthetic Hinglish sentences.

[4]We take the greatest integer i ≤ average of the two rating scores.

2.2.2 Subtask 2: Annotators' disagreement prediction

The next subtask is predicting the disagreement between the ratings provided by the human annotators to the synthetic Hinglish sentences. We calculate the disagreement between the ratings as the absolute difference between the two rating scores.

2.3 Baseline Experiments

We created a baseline with SOTA multilingual contextual language model M-BERT (Devlin et al., 2019). We finetune the pre-trained M-BERT model by adding one hidden-layer neural network on the top. We use the Relu activation function, AdamW optimizer with 0.03 learning rate, cross-entropy loss, and a batch size of 32. We use the contextual word-embedding corresponding to the synthetic Hinglish sentences in the dataset as an input to the model. The architecture remains the same for both subtasks.

2.4 Evaluation Metrics

We use the following three evaluation metrics:

- **F1-score (FS)**: We use the weighted F1-score to evaluate the system performance. The score

English	Hindi	Human-generated Hinglish	Synthetic Hinglish 1	Synthetic Hinglish 2
The reward of goodness shall be nothing but goodness.	अच्छाई का बदला अच्छाई के सिवा और क्या हो सकता है?	The reward of achai shall be nothing but achai.	reward ka badla reward ke nothing aur kya ho sakta hai **Rating1**: 7 **Rating2**: 4	reward of goodness goodness ke siva aur kya ho sakta hai **Rating1**: 9 **Rating2**: 7
		Goodness ka badla goodness ke siva aur kya ho sakta hai.		
		Achai ka badla shall be nothing but achai.		

Table 2: Example human-generated and synthetic Hinglish sentences from the dataset along with the source English and Hindi sentences. Two different human annotators rate the synthetic Hinglish sentences on the scale 1-10 (low-high quality).

ranges from 0 (worst) to 1 (best).

- **Cohen's Kappa (CK)**: We use the Cohen's Kappa score to measure the agreement between the predicted and the actual rating. The score ranges from ≤ 0 (high disagreement) to 1 (high agreement).
- **Mean Squared Error (MSE)**: MSE suggests the difference between the actual and the predicted scores. A low MSE score is preferred, with zero being the lowest possible score.

We use all three metrics for the first subtask, whereas we use FS and MSE to evaluate the second subtask.

3 Overview of Participants and Submissions

In total, 46 participants grouped in **18** teams have registered for the shared task. This includes teams from top US universities like Stanford University and Carnegie Mellon University, companies like Tencent QQ, and top Indian universities like IISc, IITK, and IITBHU. Out of **18** teams, **nine** and **eight** teams have submitted at least once during train/validation and test phase, respectively.

We requested all teams that submitted the test scores to submit the paper illustrating the methodology. Out of eight teams, we received papers from four teams listed below:

1. **IIIT Hyderabad, India** (Kodali et al., 2022): This team comprises seven researchers. The team ranked 2^{nd} against FS and CK metrics and 1^{st} against MSE metric in Subtask 1 and 3^{rd} against FS metric and 1^{st} against MSE metric in Subtask 2. Hereafter, we refer to this team as **'IIITH'**.

2. **Manipal University, India** (Singh, 2022): This team comprises one researcher. The team ranked 3^{rd} against FS and CK metrics and 1^{st} against MSE metric in Subtask 1, and 1^{st} in Subtask 2 against both FS and MSE metrics. Hereafter, we refer to this team as **'MU'**.

3. **BITS Pilani, India** (Furniturewala et al., 2022): This team comprises five researchers. The team ranked 5^{th} against FS and CK metrics and 2^{nd} against MSE metric in Subtask 1, and 2^{nd} in Subtask 2 against both FS and MSE metrics. Hereafter, we refer to this team as **'BITS'**.

4. **Jadavpur University, India** (Guha et al., 2022): This team comprises three researchers. The team ranked 9^{th} against FS, 8^{th} against CK and 3^{rd} against the MSE metric in Subtask 1 and 6^{th} against FS, and 3^{rd} against the MSE metric in Subtask 2. Hereafter, we refer to this team as **'JU'**.

Next, we discuss each of the submissions in detail:

3.1 IIITH

IIITH team leveraged two Multilingual Large Language Models (MLLMs), XLM-R (Conneau et al., 2020) and LABSE (Feng et al., 2022) to generate embeddings for Hindi, English, synthetic, and Human-generated code-mixed Hinglish sentences. In addition to the embeddings as a feature, they computed scores from three code-mixing metrics, Code-Mixing Index (CMI, (Gambäck and Das, 2016)), Number of Switch Points, and Burstiness (Goh and Barabási, 2008). All metric scores and embeddings are combined together to generate features. The features are used to train Linear Regression, MLP Regressor, and XGBoost classifiers. Out of these three, MLP Regressor performed best.

3.2 MU

MU team leveraged LABSE (Feng et al., 2022) to create embeddings for Hindi and English sentences and BERT (Devlin et al., 2019) to create embeddings for Hinglish sentences. The obtained vectors

Team Name	Subtask 1			Subtask 2	
	FS	CK	MSE	FS	MSE
Baseline	0.26637 (1)	0.09922 (1)	2.00000 (1)	0.14323 (8)	5.00000 (3)
IIITH	0.25734 (2)	0.09858 (2)	2.00000 (1)	0.23523 (3)	3.00000 (1)
MU	0.25062 (3)	0.08153 (3)	2.00000 (1)	0.26115 (1)	3.00000 (1)
BITS	0.21796 (5)	0.07337 (5)	3.00000 (2)	0.23940 (2)	4.00000 (2)
JU	0.11582 (9)	0.00337 (8)	6.00000 (3)	0.18331 (6)	5.00000 (3)

Table 3: Comparison between the four submissions. Number inside a bracket represent relative rank in respective shared task for a particular metric.

are then concatenated and fed into a catboost-based classifier (Prokhorenkova et al., 2018) to generate final predictions.

3.3 BITS

BITS team, first-of-all, finetuned a Multilingual BERT model (Pires et al., 2019), a language model pretrained on 104 languages. Then, they utilized the deep semantic features extracted from Multilingual BERT for different sentence types to train a fully connected neural network. They used the same two-fold architecture for both tasks.

3.4 JU

JU team leveraged GloVe embeddings (Pennington et al., 2014) to represent English and Hindi sentences and one hot vectors to represent Hinglish sentences. Further, GloVe embeddings were passed through respective Bidirectional LSTMs (Bi-LSTMs). The one-hot vectors are fed to a dense layer. The combined vectors from the Bi-LSTMs and dense layers are further passed through a dense layer for final predictions. They used the same architecture for both tasks.

4 Results and Discussion

In this section, we compare the four submissions for both sub-tasks. Table 3 showcases the results for four systems. Note that the table contains only those entries that submitted the final methodology paper. As illustrated, no team was able to outperform the baseline for Subtask 1. MU performed best for Subtask 2. The other entries and their corresponding rankings are present on the official leaderboard[5] of the shared task.

The four teams have leveraged large-scale language models (XLM-R, LABSE or BERT). The models were either finetuned or used for generating embeddings. The embeddings were passed to a classifier model for final predictions. Subtask 2 showcases significant improvements over baseline scores.

[5] https://codalab.lisn.upsaclay.fr/competitions/1688#results

5 Conclusion and Future Directions

In this shared task, the participating teams have created systems to evaluate the quality of the code-mixed text. These systems can help develop futuristic NLP tools that filter out noisy poor, quality code-mixed text from the good quality code-mixed text. We also proposed several research questions that need to be answered implicitly with the experiments. However, none of the team has answered these questions. We plan to explore these questions in our future editions. Overall, this shared task will help the code-mixing research community build efficient and robust code-mixed text generation and evaluation systems.

Acknowledgements

We would like to thank Mr. Pritam Kadasi for helping in setting up the competition page. We would also like to thank several researchers from academia and industry for this shared task's publicity.

References

Somnath Banerjee, Kunal Chakma, Sudip Kumar Naskar, Amitava Das, Paolo Rosso, Sivaji Bandyopadhyay, and Monojit Choudhury. 2016. Overview of the mixed script information retrieval (msir) at fire-2016. In *Forum for Information Retrieval Evaluation*, pages 39–49. Springer.

Ruthanna Barnett, Eva Codó, Eva Eppler, Montse Forcadell, 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 document for preparing and analyzing language interaction data version 1.1—july, 1999. *International Journal of Bilingualism*, 4(2):131–132.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Navya Jose, Anand Kumar M, Thomas Mandl, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Hariharan R L, John P. McCrae, and Elizabeth Sherly. 2021. Findings of the shared task on offensive language identification in Tamil, Malayalam, and Kannada. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pages 133–145, Kyiv. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and mt augmentation approach. In *Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891.

Shaz Furniturewala, Vijay Kumari, Amulya Ratna Dash, Hriday Kedia, and Yashvardhan Sharma. 2022. Bits pilani at hinglisheval: Quality evaluation for code-mixed hinglish text using transformers. *arXiv preprint arXiv:2206.08680*.

Björn Gambäck and Amitava Das. 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.

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

Prantik Guha, Rudra Dhar, and Dipankar Das. 2022. Ju_nlp at hinglisheval: Quality evaluation of the low-resource code-mixed hinglish text. *arXiv preprint arXiv:2206.08053*.

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

Prashant Kodali, Tanmay Sachan, Akshay Goindani, Anmol Goel, Naman Ahuja, Manish Shrivastava, and Ponnurangam Kumaraguru. 2022. Precogiiith at hinglisheval: Leveraging code-mixing metrics & language model embeddings to estimate code-mix quality. *arXiv preprint arXiv:2206.07988*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The iit bombay english-hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, PYKL Srinivas, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. 2020. Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 774–790.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1543–1553.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Shashi Shekhar, Dilip Kumar Sharma, and MM Sufyan Beg. 2020. Language identification framework in code-mixed social media text based on quantum lstm—the word belongs to which language? *Modern Physics Letters B*, 34(06):2050086.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018. Language identification and named entity recognition in hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58.

Nikhil Singh. 2022. niksss at hinglisheval: Language-agnostic bert-based contextual embeddings with catboost for quality evaluation of the low-resource synthetically generated code-mixed hinglish text. *arXiv preprint arXiv:2206.08910*.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72.

Vivek Srivastava and Mayank Singh. 2020. Phinc: A parallel hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49.

Vivek Srivastava and Mayank Singh. 2021a. Challenges and limitations with the metrics measuring the complexity of code-mixed text. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 6–14.

Vivek Srivastava and Mayank Singh. 2021b. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.