HinGE: A Dataset for Generation and Evaluation of Code-Mixed Hinglish Text

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

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

Abstract

Text generation is a highly active area of research in the computational linguistic community. The evaluation of the generated text is a challenging task and multiple theories and metrics have been proposed over the years. Unfortunately, text generation and evaluation are relatively understudied due to the scarcity of high-quality resources in code-mixed languages where the words and phrases from multiple languages are mixed in a single utterance of text and speech. To address this challenge, we present a corpus (*HinGE*) for a widely popular code-mixed language Hinglish (code-mixing of Hindi and English languages). *HinGE* has Hinglish sentences generated by humans as well as two rule-based algorithms corresponding to the parallel Hindi-English sentences. In addition, we demonstrate the inefficacy of widely-used evaluation metrics on the code-mixed data. The *HinGE* dataset will facilitate the progress of natural language generation research in code-mixed languages.

1 Introduction

Code-mixing is the mixing of two or more languages in a single utterance of speech or text. A commonly observed communication pattern for a multilingual speaker is to mix words and phrases from multiple languages. Code-mixing is widespread across various language pairs, such as Spanish-English, Hindi-English, and Bengali-English. Recently, we observe a boom in the availability of code-mixed data with the inflation of the social media platforms such as Twitter and Facebook.

In past, we witness magnitude of work to address standard code-mixing natural language understanding (NLU) tasks such as language identification (Shekhar et al., 2020; Singh et al., 2018a; Ramanarayanan et al., 2019), POS tagging (Singh et al., 2018b; Vyas et al., 2014), named entity recognition (Singh et al., 2018a), and dependency parsing (Zhang et al., 2019a) along with sentence classification tasks like sentiment analysis (Patwa et al., 2020; Joshi et al., 2016), stance detection (Utsav et al., 2020), and sarcasm detection (Swami et al., 2018). Unlike code-mixed NLU, natural language generation (NLG) of code-mixed text is highly understudied. Resource scarcity adds to the challenge of building efficient solutions for code-mixed NLG tasks. Evaluation of the code-mixed NLG tasks also lacks standalone resources, theories, and metrics.

Recently, we observe a growing interest in the code-mixed text generation task. To generate the code-mixed data various techniques have been employed such as matrix frame language theory (Lee et al., 2019; Gupta et al., 2020; Jain et al., 2021), equivalent constraint theory (Pratapa et al., 2018), pointer-generator network (Winata et al., 2018, 2019; Gupta et al., 2020), Generative Adversarial Networks (GANs) (Gao et al., 2019), etc. The majority of the available datasets (Rijhwani et al., 2017; Solorio et al., 2014; Patro et al., 2017) employed in code-mixed NLG contains noisy code-mixed text collected from social media platforms such as Twitter. These datasets also lack the sanity check for the quality of sentences, making the systems developed on these datasets vulnerable to real-world applicability. To address the challenge of scarcity of high-quality resources for the code-mixed NLG tasks, we propose *HinGE* dataset[1] that will facilitate the community to build robust systems. The dataset contains sentences generated by humans as well as two rule-based algorithms. In Table 1, we compare *HinGE* with three other baseline datasets that can be used in the Hinglish code-mixed text generation and evaluation task.

In addition to the code-mixed NLG, the evaluation of the generated code-mixed text is a challenging task. The widely popular metrics for monolin-

[1]https://sites.google.com/view/vivek-srivastava/resources

Dataset characteristics	Banerjee et al. (2018)	Srivastava and Singh (2020)	Gupta et al. (2020)	***HinGE***
Parallel source sentences	Only ES	Only ES	✓	✓
Human-generated code-mixed sentences	✓	✓	✗	✓
Multiple human-generated code-mixed sentences for a parallel sentence	✗	✗	✗	✓
Machine-generated code-mixed sentences	✗	✗	✓	✓
Multiple machine-generated code-mixed sentences for a parallel sentence	✗	✗	✓	✓
Human ratings for the quality of generated code-mixed sentences	✗	✗	✗	✓
Dataset size	6733 UEU, 6549 UHU	13,738	1,561,840 PS, 252,330 MGHS	1,976 PS, 4,803 HGHS, 3,952 MGHS

Table 1: Comparison between the various datasets available for the Hinglish NLG tasks. UEU: Unique English Utterance, UHU: Unique Hinglish Utterance, ES: English Sentences, PS: Parallel Sentences, HGHS: Human-Generated Hinglish Sentences, MGHS: Machine-Generated Hinglish Sentences.

gual languages fail to capture the linguistic diversity present in the code-mixed data, such as spelling variation and complex sentence structuring. The quality ratings of the sentences generated by the rule-based algorithms in *HinGE* dataset will help to develop the metrics and theories for evaluating the code-mixed NLG tasks. Our main contributions are:

- We create high-quality human-generated code-mixed Hinglish sentences corresponding to the parallel Hindi-English sentences. Each pair of parallel sentences has at least two human-generated Hinglish sentences.
- In addition to the human-generated code-mixed sentences, we propose two rule-based algorithms to generate the Hinglish sentences.
- We demonstrate the inefficacy of five widely popular metrics for the NLG task with the code-mixed text.
- To develop efficient metrics for the code-mixed NLG tasks, we provide the human ratings corresponding to the code-mixed sentences generated by the rule-based algorithms.

2 Human-Generated Hinglish Text

The scarcity of high-quality code-mixed datasets limits current research in various NLU tasks such as text generation and summarization. To address this challenge, we create a human-generated corpus of Hinglish sentences corresponding to parallel monolingual English and Hindi sentences. We use the IIT Bombay English-Hindi Parallel Corpus (hereafter *'IIT-B corpus'*) (Kunchukuttan et al., 2018). The IIT-B corpus has 1,561,840 parallel sentence pairs in English and Hindi. The English sentences are written in the Roman script, and the Hindi sentences are written in the Devanagari script. We randomly select 5,000 sentence pairs, in which the number of tokens in both the sentences is more than five to create a human-generated parallel corpus.

To create the gold-standard dataset, we employ five human annotators. Each annotator has expert-level proficiency in writing, speaking, and understanding English and Hindi languages. The objective of the annotation is to generate at least two unique Hinglish code-mixed sentences corresponding to the parallel English and Hindi sentence pairs. The annotators can also generate more than two code-mixed sentences for each sentence pair. We shuffle, pre-process, and share the sentence pairs with the annotators to generate the corresponding Hinglish sentences. A single annotator annotates each sentence pair.

We assign 1,000 unique sentence pairs to each annotator with the following annotation guidelines:

- The Hinglish sentence should be written in Roman script.
- The Hinglish sentence should have words from both the languages, i.e., English and Hindi.
- Avoid using new words, wherever possible, that are not present in both the sentences.
- If the source sentences are not the translation of each other, mark the sentence pair as "#".

Post annotation, we remove the sentence pairs marked as "#" or are missing an annotation. Note that due to the complexity of generating code-mixed sentences, such as the inability to identify two unique Hinglish sentences, complex sentence structuring, and usage of difficult words in monolingual sentences, annotators do not provide two unique sentences Hinglish sentences for each monolingual sentence pair. We obtain 1,978 sentence pairs with two or more unique Hinglish sentences. On average, 2.5 code-mixed sentences are created for each Hindi-English sentence pair. Figure 1 shows an example of two code-mixed sentences generated by the annotator for a given sentence pair.

Qualitative evaluation of the human-generated Hinglish text: To qualitatively evaluate the generated sentences, we adapt the evaluation strategy described in (Srivastava and Singh, 2021). We randomly sample 100 Hinglish sentences generated by humans along with the source parallel monolingual English and Hindi sentences. We employ two human annotators[2] for the qualitative evaluation. We ask the annotators to rate each Hinglish sentence on two metrics:

- **Degree of code-mixing (DCM)**: The code-mixing index (CMI) proposed by (Das and Gambäck, 2014) measures the degree of code-mixing in a text based on language tags of the participating tokens. The objective of DCM follows CMI with an exception of explicit token language identification to measure the degree of code-mixing in the text. The DCM score can vary between 0 to 10. A DCM score of 0 corresponds to the monolingual sentence with no code-mixing, whereas the DCM score of 10 suggests a high degree of code-mixing.
- **Readability (RA)**: RA score can vary between 0 to 10. A completely unreadable sentence due to many spelling mistakes, no sentence structuring, or meaning yields a RA score of 0. A RA score of 10 suggests a highly readable sentence with clear semantics and easy-to-read words.

The average DCM scores by the two human annotators are 8.72 and 8.65. The average RA scores are 8.65 and 8.37. The high average scores demonstrate good quality code-mixed sentence generation. Table 2 shows example ratings provided by the two human annotators to the five Hinglish sentences.

[2]different from the human annotators who generate the Hinglish sentences.

ENGLISH: Choose the version of the Kolab Server you are using.
HINDI: कोलाब संस्करण जो आप इस्तेमाल कर रहे हैं उसे चुनें
HINGLISH 1: Kolab Server version ka jo aap isstemaal kar rahe hai use choose kare.
HINGLISH 2: Kolab Server version ka jo aap isstemaal kar rahe hai use chune.

Figure 1: Example of the code-mixed sentences generated by the annotator for an English-Hindi sentence pair.

3 Machine-Generated Hinglish Text

In addition to the human-generated code-mixed text, we also generate the Hinglish sentence synthetically by following the Embedded-Matrix theory (Joshi, 1982). We propose two rule-based Hinglish text generation systems leveraging the parallel monolingual English and Hindi sentences. In both systems, we use Hindi as the matrix language and English as the embedded language. The matrix language imparts structure to the code-mixed text with tokens embedded from embedded language. We also use several linguistic resources in both generation systems. These resources include:

- **English-Hindi Dictionary**: We curate 77,805 pairs of English words and the corresponding Hindi meanings from two sources[3,4] to construct an English-Hindi dictionary.
- **Cross-lingual Word Embedding**: We leverage multilingual word vectors for English and Hindi tokens. These word-vectors ($dim = 300$) are generated from fastText's multilingual pre-trained model (Bojanowski et al., 2017). We further map these vectors to a common space using VecMap (Artetxe et al., 2018).
- **GIZA++**: GIZA++ (Och and Ney, 2003) learns the word alignment between the parallel sentences using an HMM based alignment model in an unsupervised manner. We train GIZA++ on *IIT-B corpus*.
- **Script Transliteration**: We transliterate the code-mixed sentences containing tokens in the Devanagari script to the Roman script (Mishra, 2019).

[3]http://www.cfilt.iitb.ac.in/~hdict/webinterface_user/index.php

[4]https://jankaribook.com/verbs-list-verb-in-hindi/

Hinglish sentences	Human 1		Human 2	
	DCM	RA	DCM	RA
Media ke exposure ke aadhar par Indian rajyo ki rankings	10	10	8	8
You will be more likely to give up before the 30 minutes, aap logo se jyada he.	7	8	9	8
Shighra hi maansingh british ke saath saude baazi kar rha tha.	8	10	8	8
par there's another way, aur mai aapko bata kar ja rahi hun.	9	10	9	9
"Aren't you a tiny bit andhvishavaasi?"	9	9	9	9

Table 2: Example DCM and RA ratings provided by the two human annotators to the human-generated Hinglish sentences. We color code the tokens in the Hinglish sentence based on the language with the scheme: English tokens with orange, Hindi tokens with blue and language independent tokens with black color.

- **YAKE**: We use YAKE (Campos et al., 2020), an unsupervised automatic keyword extraction method, to extract the key-phrases from the monolingual English and Hindi sentences.

We further extend the English-Hindi dictionary by incorporating parallel sentences in the *IIT-B corpus*. We leverage VecMap's shared representation to identify the closest word in English and the corresponding Hindi sentence. In addition, we also leverage GIZA++ to align the parallel sentences resulting in aligned English-Hindi tokens. Both of these steps extend the initial dictionary from 77,809 to 1,52,821 words and meaning pairs. We use this extended dictionary in the following two code-mixed text generation systems:

- **Word-aligned code-mixing (WAC)**: Here, we align the noun and adjective tokens between the parallel sentences using the extended dictionary. We replace all the aligned Hindi tokens with the corresponding English noun or adjective token and transliterate the resultant Hindi sentence to the Roman script. Figure 2 demonstrates the example Hinglish text generated from the parallel monolingual English and Hindi sentences using the WAC procedure.

- **Phrase-aligned code-mixing (PAC)**: Here, we align the keyphrases of length up to three tokens between the parallel sentences. To identify the keyphrases, we use the YAKE tool. We replace all the aligned Hindi phrases with the corresponding longest matching English phrase and transliterate the resultant Hindi sentence to the Roman script. Figure 3 demonstrates the example Hinglish text generated from the parallel monolingual English and Hindi sentences using the PAC procedure.

It should be noted that the code-mixed sentences generated using the WAC and PAC procedures have added noise (spelling variations, grammatical inconsistencies, etc.) and are not an exact representation of the high-quality code-mixed text generated by humans. This noisy machine-generation of the code-mixed text is intentional such that the efficacy of the evaluation metrics could be studied (see Section 4) for the wide spectrum of sentences and should not be limited to good quality code-mixed sentences.

English sentence: Cannot send message: no recipients defined.

Hindi sentence: संदेश नहीं भेज सकता है: कोई प्राप्तकर्ता परिभाषित नहीं है.

Alignment candidates in English sentence: {message, recipients}

Alignment candidates in Hindi sentence: {संदेश, प्राप्तकर्ता}

Aligned words using extended dictionary: {message→ संदेश, recipients→प्राप्तकर्ता, }

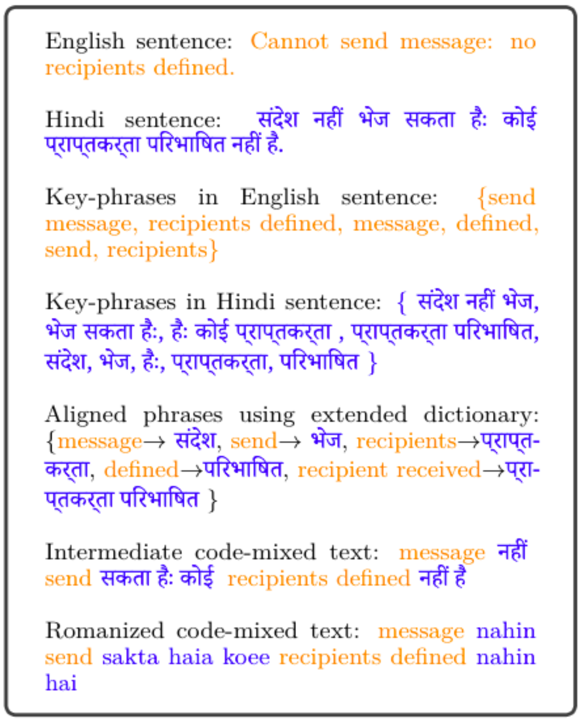
Intermediate code-mixed text: message नहीं भेज सकता है: कोई recipients परिभाषित नहीं है

Romanized code-mixed text: message nahin bhej sakta haia koee recipients paribhasit nahin hai

Figure 2: An example Hinglish code-mixed sentence generated using WAC method.

4 A Study on Evaluation of Code-Mixed Text Generation

In this section, we evaluate machine-generated code-mixed text. This study demonstrates severe limitations of five widely popular NLP metrics in evaluating code-mixed text generation performance. We leverage the following metrics: (i) Bilingual Evaluation Understudy Score (**BLEU**, Papineni et al. (2002)), (ii) **NIST** (Doddington, 2002), (iii) BERTScore (**BS**, Zhang et al.



English sentence: Cannot send message: no recipients defined.

Hindi sentence: संदेश नहीं भेज सकता है: कोई प्राप्तकर्ता परिभाषित नहीं है.

Key-phrases in English sentence: {send message, recipients defined, message, defined, send, recipients}

Key-phrases in Hindi sentence: { संदेश नहीं भेज, भेज सकता है:, है: कोई प्राप्तकर्ता , प्राप्तकर्ता परिभाषित, संदेश, भेज, है:, प्राप्तकर्ता, परिभाषित }

Aligned phrases using extended dictionary: {message→ संदेश, send→ भेज, recipients→प्राप्तकर्ता, defined→परिभाषित, recipient received→प्राप्तकर्ता परिभाषित }

Intermediate code-mixed text: message नहीं send सकता है: कोई recipients defined नहीं है

Romanized code-mixed text: message nahin send sakta haia koee recipients defined nahin hai

Figure 3: An example Hinglish code-mixed sentence generated using PAC method.

(2019b)), (iv) Word Error Rate (**WER**, Levenshtein (1966)), and (v) Translator Error Rate (**TER**, Snover et al. (2006)). Higher BLEU, NIST, or BS values and lower WER or TER values represent better generation performance. We conduct two experiments to evaluate the machine-generated Hinglish text:

- **Human evaluation**: First, we perform coarse-grained qualitative evaluation by randomly sampling 100 English-Hindi sentence pairs and corresponding WAC and PAC generated sentences. We employ two human evaluators[5] who are proficient in English and Hindi languages to evaluate the quality of the generated sentences. We ask evaluators to provide one of the two labels — *Correct* and *Incorrect* — to each of the generated sentences. A sentence is marked *Correct* if it is following the semantics of the parallel sentences and has high readability. Table 3 shows the annotator's agreement on the randomly sampled set of sentences. The coarse-grained qualitative evaluation shows the correct generation in at least 50% of the cases. Table 4 shows the evaluation of the randomly sampled 100 sentences on five metrics. The results show the inefficacy of automatic evaluation metrics to capture the linguistic diversity of the generated code-mixed text due to spelling variations, omitted words, limited reference sentences, and informal writing style (Srivastava and Singh, 2020). We further conduct a fine-grained qualitative human evaluation to measure the similarity between the machine-generated Hinglish sentences and the monolingual pair, the readability, and the grammatical correctness. We employ a new set of eight human evaluators to provide a rating between 1 (low quality) to 10 (high quality) to the PAC and WAC generated Hinglish sentences based on the following three parameters:

  – The similarity between the generated Hinglish and the monolingual source sentences.
  – The readability of the generated sentence.
  – The grammatical correctness of the generated sentence.

  Each generated sentence is rated by two human evaluators. All the evaluators have expert-level proficiency in the English and Hindi languages. Table 5 shows ratings provided to a representative machine-generated Hinglish sentence. Figure 4 shows the distribution of the fine-grained human evaluation scores. For both procedures, the majority (WAC: 82.3% and PAC: 75.2%) of the sentences score in the range 6–9. None of the sentences received a rating score of 1. The results further corroborate our claim that automatic evaluation metrics undermine code-mixed text generation performance. Figure 5 shows the distribution of the disagreement in the human evaluation of the generated sentences. We calculate the disagreement as to the absolute difference between the human evaluation scores. PAC-generated sentences are more prone to high disagreement (>=5) in the human evaluation than WAC. This could be attributed to the fact that PAC-generated sentences are relatively less constrained, which leaves the evaluation to the expertise and interpretation of the Hinglish language by the annotators.

- **Metric-based evaluation**: Next, we analyze the performance of five evaluation metrics on

[5]The evaluators are different from the annotators employed in Section 2).

	WAC		PAC	
	Agree	**Disagree**	**Agree**	**Disagree**
Correct	56	22	55	40
Incorrect	22		5	

Table 3: Annotator agreement on the randomly sampled sentences for WAC and PAC.

	BLEU	WER	TER	NIST	BS
WAC	0.1229	0.8240	0.7830	2.2045	0.857
PAC	0.1202	0.8228	0.7981	2.0497	0.857

Table 4: Automatic performance evaluation of the WAC and PAC procedures on the randomly sampled sentences.

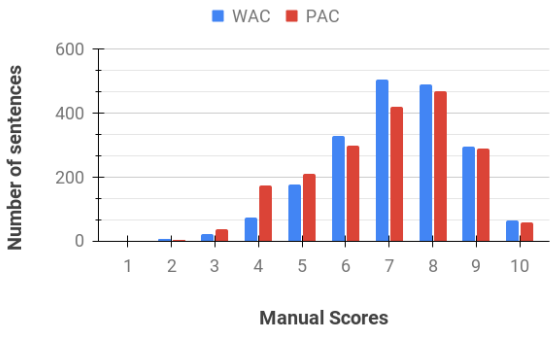


Figure 4: Distribution of human evaluation of the generated Hinglish sentences using WAC and PAC.

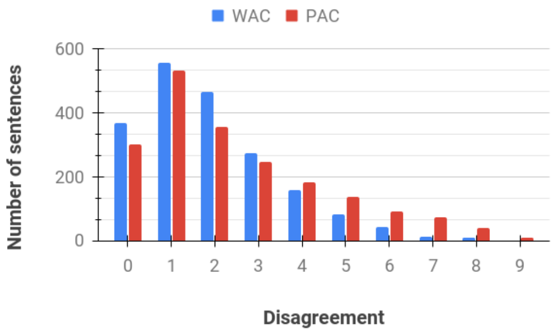


Figure 5: Distribution of the disagreement between human evaluation of the generated Hinglish sentences using WAC and PAC.

the machine-generated code-mixed text. Table 6 shows the average scores for each of the metrics against the corresponding human-provided rating. As evident from the results, the metrics perform poorly on the code-mixed data. We further analyze the correlation[6] of the metric scores with the human-provided ratings for both the text generation procedures.

[6]We experiment with Pearson Correlation Coefficient. The value ranges from -1 to 1.

For this task, we divide the human ratings into three buckets:

- Bucket 1: Human rating between 2–10.
- Bucket 2: Human rating between 2–5.
- Bucket 3: Human rating between 6–10.

Table 7 shows the results of the correlation between various metric scores and the human rating to the machine-generated code-mixed sentences using WAC and PAC procedures. Human rating for generated sentences in Bucket 3 is relatively highly correlated with the metric scores compared to Bucket 2. This behavior could be attributed to the fact that low-quality sentences are difficult to rate for the human annotators due to various reasons such as poor sentence structuring and many spelling mistakes.

5 Limitations and Opportunities

In this section, we present a discussion on the various inherent limitations associated with the proposed *HinGE* dataset. We also discuss the various opportunities for the computational linguistic community to build efficient systems and metrics for code-mixed languages. Some of the major limitations with the dataset are:

- Due to the high time and cost associated with the human annotations, the number of samples in the dataset is limited. Because of a similar reason, the other code-mixing datasets (Srivastava and Singh, 2021) suffer from the scarcity of large-scale human annotations.

- The IIT-B parallel corpus does not contain sentences mined from the social media platforms. This potentially reduces the noise in the generated sentences compared to the dataset previously compiled from the social media platforms for various tasks.

- The annotators generating the code-mixed sentences were constrained only to include words from the parallel source sentences. This could potentially limit the observations compared to the real-world datasets collected from social media platforms that are more linguistically diverse due to the presence of a large number of multilingual speakers.

- Due to the high annotation cost and time, the WAC and PAC generated sentences are only rated on a single scale encompassing multiple

English	Hindi	Human-generated Hinglish	WAC	PAC
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 5: Example human-generated and machine-generated 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

Human	**WAC**					**PAC**				
Score	**BLEU**	**WER**	**TER**	**NIST**	**BS**	**BLEU**	**WER**	**TER**	**NIST**	**BS**
2	0.144	0.741	0.667	0.092	0.851	0.126	0.672	0.698	0.176	0.8603
3	0.138	0.735	0.708	0.070	0.852	0.146	0.765	0.696	0.086	0.851
4	0.133	0.695	0.666	0.103	0.849	0.143	0.744	0.703	0.100	0.8464
5	0.135	0.711	0.681	0.110	0.853	0.153	0.726	0.680	0.114	0.8515
6	0.141	0.697	0.670	0.102	0.852	0.164	0.689	0.646	0.124	0.8558
7	0.161	0.663	0.630	0.111	0.856	0.176	0.661	0.618	0.121	0.8581
8	0.177	0.621	0.589	0.127	0.859	0.177	0.639	0.605	0.128	0.8598
9	0.212	0.571	0.538	0.150	0.865	0.184	0.614	0.590	0.129	0.8638
10	0.291	0.509	0.493	0.157	0.878	0.242	0.551	0.543	0.146	0.8731

Table 6: Comparison of various metric scores with the human score for WAC and PAC.

	Bucket 1		**Bucket 2**		**Bucket 3**	
	WAC	PAC	WAC	PAC	WAC	PAC
BLEU	0.810	0.910	-0.861	0.878	0.941	0.844
WER	-0.936	-0.822	-0.785	0.457	-0.993	-0.973
TER	-0.891	-0.963	0.000	-0.610	-0.998	-0.970
NIST	0.913	0.127	0.642	-0.559	0.986	0.846
BS	0.844	0.710	0.227	-0.689	0.953	0.937

Table 7: Comparison of correlation between evaluation metrics and human scores for WAC and PAC.

dimensions such as grammatical correctness, readability, etc.

Even with the presence of the above limitations, the *HinGE* dataset could be effectively used for various purposes such as:

- The dataset could be effectively used in developing code-mixing text generation systems. Currently, the dataset supports only one code-mixed language, i.e., Hinglish, but it could be extended using various techniques such as weak supervision and active learning.
- The machine-generated sentences and the corresponding human ratings will be useful in designing metrics and systems for the effective evaluation of various code-mixed NLG tasks. It could also be used to investigate the factors influencing the quality of the code-mixed text.
- The dataset could also be used in investigating the reasoning behind the disagreement in the human scores to the machine-generated sentences.
- The code-mixed text (human or machine-generated) could be useful in the multitude of other code-mixing tasks such as language identification and POS tagging.
- With the recent thrust in code-mixed machine translation, the *HinGE* dataset would be extremely useful in designing and evaluating the machine-translation systems. The multiple code-mixed sentences corresponding to a given pair of parallel monolingual sentences would help to build robust translation systems.

6 Conclusion

In this paper, we present a high-quality dataset (*HinGE*) for the text-generation and evaluation task in the code-mixed Hinglish language. The code-mixed sentences in the *HinGE* dataset are generated by humans and rule-based algorithms. We demonstrate the poor evaluation capabilities of five widely popular metrics on the code-mixed data. Along with the human-generated sentences, the machine-generated sentences (as described in Section 3) and the human ratings of these code-mixed sentences could facilitate building the highly scalable and robust evaluation metrics and strategies for the code-mixed text. The multiple human-generated sentences corresponding to a pair of parallel monolingual sentences will pave the way in designing natural language generation systems robust to adversaries and linguistic diversities such as spelling variation and matrix language.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Suman Banerjee, Nikita Moghe, Siddhartha Arora, and Mitesh M Khapra. 2018. A dataset for building code-mixed goal oriented conversation systems. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3766–3780.

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.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

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.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Yingying Gao, Junlan Feng, Ying Liu, Leijing Hou, Xin Pan, and Yong Ma. 2019. Code-switching sentence generation by bert and generative adversarial networks. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association, INTERSPEECH 2019*, pages 3525–3529.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2020. A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2267–2280, Online. Association for Computational Linguistics.

Dhruval Jain, Arun D Prabhu, Shubham Vatsal, Gopi Ramena, and Naresh Purre. 2021. Codeswitched sentence creation using dependency parsing. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 124–129, Los Alamitos, CA, USA. IEEE Computer Society.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of hindi-english code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491.

Aravind Joshi. 1982. Processing of sentences with intra-sentential code-switching. In *Coling 1982: Proceedings of the Ninth International Conference on Computational Linguistics*.

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)*.

Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically Motivated Parallel Data Augmentation for Code-Switch Language Modeling. In *Proc. Interspeech 2019*, pages 3730–3734.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Ritwik Mishra. 2019. devanagari-to-roman-script-transliteration. `https://github.com/ritwikmishra/devanagari-to-roman-script-transliteration`.

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

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

Jasabanta Patro, Bidisha Samanta, Saurabh Singh, Abhipsa Basu, Prithwish Mukherjee, Monojit Choudhury, and Animesh Mukherjee. 2017. All that is English may be Hindi: Enhancing language identification through automatic ranking of the likeliness of word borrowing in social media. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2264–2274, Copenhagen, Denmark. Association for Computational Linguistics.

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.

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, Melbourne, Australia. Association for Computational Linguistics.

Vikram Ramanarayanan, Robert Pugh, Yao Qian, and David Suendermann-Oeft. 2019. Automatic turn-level language identification for code-switched spanish–english dialog. In *9th International Workshop on Spoken Dialogue System Technology*, pages 51–61. Springer.

Shruti Rijhwani, Royal Sequiera, Monojit Choudhury, Kalika Bali, and Chandra Shekhar Maddila. 2017. Estimating code-switching on Twitter with a novel generalized word-level language detection technique. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1971–1982, Vancouver, Canada. Association for Computational Linguistics.

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. 2018a. Language identification and named entity recognition in hinglish code mixed tweets. In *Proceedings of ACL 2018, Student Research Workshop*, pages 52–58.

Kushagra Singh, Indira Sen, and Ponnurangam Kumaraguru. 2018b. A twitter corpus for hindi-english code mixed pos tagging. In *Proceedings of the Sixth International Workshop on Natural Language Processing for Social Media*, pages 12–17.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 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, Doha, Qatar. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2020. Phinc: a parallel hinglish social media code-mixed corpus for machine translation. *arXiv preprint arXiv:2004.09447*.

Vivek Srivastava and Mayank Singh. 2021. 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.

Sahil Swami, Ankush Khandelwal, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A corpus of english-hindi code-mixed tweets for sarcasm detection. *arXiv preprint arXiv:1805.11869*.

Jethva Utsav, Dhaiwat Kabaria, Ribhu Vajpeyi, Mohit Mina, and Vivek Srivastava. 2020. Stance detection in hindi-english code-mixed data. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, CoDS COMAD 2020, page 359–360, New York, NY, USA. Association for Computing Machinery.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 974–979.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Learn to Code-Switch: Data Augmentation using Copy Mechanism on Language Modeling. *arXiv e-prints*.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 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.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2019a. Cross-lingual dependency parsing using code-mixed treebank. 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 996–1005.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.