Challenges and Limitations with the Metrics Measuring the Complexity of Code-Mixed 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

Code-mixing is a frequent communication style among multilingual speakers where they mix words and phrases from two different languages in the same utterance of text or speech. Identifying and filtering code-mixed text is a challenging task due to its co-existence with monolingual and noisy text. Over the years, several code-mixing metrics have been extensively used to identify and validate code-mixed text quality. This paper demonstrates several inherent limitations of code-mixing metrics with examples from the already existing datasets that are popularly used across various experiments.

1 Introduction

Code-mixing is the phenomenon of mixing words and phrases from multiple languages in the same utterance of a text or speech (Bokamba, 1989). Multilingual societies observe a high frequency of code-mixed communication in the informal setting such as social media, online messaging, discussion forums, and online gaming (Tay, 1989). Various studies indicate the overwhelming growth in the number of code-mixed speakers in various parts of the world, such as India, Spain, and China (Baldauf, 2004). The phenomenal increase of the code-mixed data on various platforms such as Twitter, Facebook, WhatsApp, Reddit, and Quora, has led to several interesting research directions such as token-level language identification (Shekhar et al., 2020; Singh et al., 2018a), POS tagging (Vyas et al., 2014; Singh et al., 2018b), machine translation (Dhar et al., 2018; Srivastava and Singh, 2020), and question-answering (Chandu et al., 2019; Banerjee et al., 2016).

Despite such active participation from the computational linguistic community in developing tools and resources for the code-mixed languages, we observe many challenges in processing the code-mixed data. One of the most compelling problems with the code-mixed data is the co-existence with the noisy and monolingual data. In contrast to the monolingual languages, we do not find any platform where the code-mixed language is the only medium of communication. The co-existing nature of the code-mixed languages with the noisy and monolingual languages posits the fundamental challenge of filtering and identifying the code-mixed text relevant for a given study. Over the years, various works have employed human annotators for this task. However, employing humans for identifying and filtering the code-mixed text (in addition to the task-specific annotations) is extremely expensive on both fronts of time and cost. Also, since code-mixed languages do not follow specific linguistic rules and standards, it becomes increasingly challenging to evaluate human annotations and proficiency.

In order to address some of the above challenges, several code-mixing metrics (Das and Gambäck, 2014; Gambäck and Das, 2016; Barnett et al., 2000; Guzmán et al., 2017) have been proposed to measure the degree of code-mixing in the text. However, we observe several limitations in the metric formulations. This paper outlines several such limitations and supports our claims with examples from multiple already existing datasets for various tasks. For illustrations, we choose Hinglish (code-mixing of Hindi and English language) due to two major reasons: (i) popularity of Hinglish and (ii) active research community. Baldauf (2004) projected that number of Hinglish speakers might soon outrun the number of native English speakers in the world. This strengthens our belief that even though Hinglish (and other code-mixed languages) does not enjoy the official status, we need to build robust systems to serve the multilingual societies. With the availability of datasets and tools for the Hinglish language, we seek a boom in the active participation from the computational linguistic community to address various challenges.

Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching, pages 6–14
June 11, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-954-452-056-4_002

Data source	Task	Dataset size	Reported CMI
Singh et al. (2018c)	Named-entity recognition	3,638	Unavailable
Swami et al. (2018)	Sarcasm detection	5,520	Unavailable
Joshi et al. (2016)	Sentiment analysis	3,879	Unavailable
Patwa et al. (2020)	Sentiment analysis	20,000	25.32
Barman et al. (2014)	Language identification	771	13
Bohra et al. (2018)	Hate-speech detection	4,575	Unavailable
Dhar et al. (2018)	Machine translation	6,096	30.5
Srivastava and Singh (2020)	Machine translation	13,738	75.76
Vijay et al. (2018)	Irony detection	3,055	Unavailable
Khanuja et al. (2020)	Natural language inference	2,240	>20

Table 1: We explore 10 Hinglish code-mixed datasets to showcase the limitations of code-mixing metrics.

Outline of the paper: We formally define Hindi-English code-mixing in Section 2. Section 3 describes several code-mixing metrics. We outline various limitations supported with multiple examples from various datasets in Section 4. We conclude and present future direction in Section 5.

2 Hinglish: Mixing Hindi with English

Hinglish is a portmanteau of Hindi and the English language. Figure 1 shows example Hinglish sentences. Also, we see two example sentences in Figure 1 that are non-code-mixed but might appear to contain words from two languages. The presence of named entities from the Hindi language does not make the sentence code-mixed.

Code-mixed sentences

SENTENCE 1: ye ek code mixed sentence ka example hai
SENTENCE 2 : kal me movie dekhne ja raha hu. How are the reviews?

Non-code-mixed sentences

SENTENCE 1: Tendulkar scored more centuries than Kohli in Delhi.
SENTENCE 2: Bhartiya Janta Party won the 2019 general elections.

Figure 1: Example code-mixed sentences with words from Hindi and English languages. The non-code-mixed sentences might get confused with the code-mixed sentence due to the presence of named entities.

In this study, we explore 10 Hinglish datasets encompassing eight different tasks, namely named entity recognition, sarcasm detection, sentiment analysis, language identification, hate-speech detection, machine translation, irony detection, and natural language inference (see Table 1 for more details). Contrasting against monolingual datasets for similar tasks, the Hinglish datasets are significantly smaller in size. We support our claims by providing illustrative examples from these datasets.

3 Code-Mixing Metrics

In this section, we describe several popular code-mixing metrics that measure the complexity of the code-mixed text. Among the following metrics, code-mixing index (CMI, Das and Gambäck (2014); Gambäck and Das (2016)) is the most popular metric.

3.1 Code-mixing Index

CMI metric (Das and Gambäck, 2014) is defined as follows:

$$CMI = \begin{cases} 100 * [1 - \frac{max(w_i)}{n-u}] & n > u \\ 0 & n = u \end{cases} \quad (1)$$

Here, w_i is the number of words of the language i, max$\{w_i\}$ represents the number of words of the most prominent language, n is the total number of tokens, u represents the number of language-independent tokens (such as named entities, abbreviations, mentions, and hashtags).

A low CMI score indicates monolingualism in the text whereas the high CMI score is an indicator of the high degree of code-mixing in the text. In the later work, (Gambäck and Das, 2016) also introduced number of code alternation points in the original CMI formulation. An *alternation point* (a.k.a. *switch point*) is defined as any token in the text that is preceded by a token with a different language tag. Let f_p denotes ratio of number of code alternation points P per token, $f_p = \frac{P}{n}$ where

$0 \leq P < n$. Let CMI$_{old}$ denotes the CMI formulation defined in Eq. 1. The updated CMI formulation (CMI$_{new}$) is defined as:

$$CMI_{new} = a.CMI_{old} + b.f_p \tag{2}$$

where a and b are weights, such that $a + b = 1$. Again, CMI_{new} = 0 for monolingual text, as CMI_{old} = 0 and $P = 0$. Hereafter, throughout the paper, we refer to CMI_{new} as CMI metric.

3.2 M-index

Barnett et al. (2000) proposed the Multilingual Index (M-index). M-index measures the inequality of the distribution of language tags in a text comprising at least two languages. If p_j is the total number of words in the language j over the total number of words in the text, and $j \in k$, where k is total number of languages in the text, M-index is defined as:

$$M - index = \frac{1 - \sum p_j^2}{(k-1)\sum p_j^2} \tag{3}$$

The index varies between 0 (monolingual utterance) and 1 (a perfect code-mixed text comprising equal contribution from each language).

3.3 I-index

The Integration-index proposed by Guzmán et al. (2017) measures the probability of switching within a text. I-index approximates the probability that any given token in the corpus is a switch point. Consider a text comprised of n tokens, I-index is defined as:

$$I - index = \frac{\sum_{1 \leq i < n-1} S(i, i+1)}{n-1} \tag{4}$$

Here, $S(i, i+1) = 1$ if language tag of i^{th} token is different than the language tag of $(i+1)^{th}$ token, otherwise $S(i, i+1) = 0$. I-index varies between 0 (monolingual utterance) and 1 (a perfect code-mixed text comprising consecutive tokens with different language tag). Guzmán et al. (2017) also adapted two metrics that quantify burstiness and memory in complex systems (Goh and Barabási, 2008) to measure the complexity of code-mixed text. Next, we introduce these complex system-based metrics.

3.4 Burstiness

Burstiness (Goh and Barabási, 2008) measures whether switching occurs in bursts or has a more periodic character. Let σ_r denote the standard deviation of the language spans and m_r the mean of the language spans. Burstiness is calculated as:

$$Burstiness = \frac{\sigma_r - m_r}{\sigma_r + m_r} \tag{5}$$

The burstiness metric is bounded within the interval [-1, 1]. Text with periodic dispersions of switch points yields a burstiness value closer to -1. In contrast, text with high burstiness and containing less predictable switching patterns take values closer to 1.

3.5 Memory

Memory (Goh and Barabási, 2008) quantifies the extent to which the length of language spans tends to be influenced by the length of spans preceding them. Let n_r be the number of language spans in the utterance and τ_i denote a specific language span in that utterance ordered by i. Let σ_1 and μ_1 be the standard deviation and mean of all language spans but the last, where σ_2 and μ_2 are the standard deviation and mean of all language spans but the first.

$$Memory = \frac{1}{n_r - 1}\sum_{1}^{n_r - 1}\frac{(\tau_i - \mu_1)(\tau_{i+1} - \mu_2)}{\sigma_1 \sigma_2} \tag{6}$$

Memory varies in an interval [-1,1]. Memory values close to -1 describe the tendency for consecutive language spans to be negatively correlated, that is, short spans follow long spans, and vice-versa. Conversely, memory values closer to 1 describe the tendency for consecutive language spans to be positively correlated, meaning similar in length.

In addition to the above metrics, there exist several other code-mixing metrics such as Language Entropy and Span Entropy that can be derived from the above metrics (Guzmán et al., 2017). Due to the space constraints, we refrain from further discussing them in the paper.

Evaluating metric scores on code-mixed datasets: To understand the effectiveness of these metrics, we randomly sample one sentence each from the ten datasets and calculate the score on all the code-mixing metrics. In addition, we employ three human annotators proficient in both the languages (English and Hindi) to rate the sentences

Hinglish sentence	CMI	M-index	I-index	Burstiness	Memory	Human 1		Human 2		Human 3	
						DCM	RA	DCM	RA	DCM	RA
Deepak ji, channel ko kitna fund diya hai congress ne? 2006 me ameithi rape case kyu nahi discuss kiya kabhi?	3.53	7.59	7.27	-0.46	-0.12	8	10	9	10	10	8
4 din me 2 accidents, kuch to jhol hai, shayad politics ho rahi hai..	1.67	6.2	5	-0.19	-0.31	4	9	5	10	10	9
Bhai kasam se bata do ki shadi kab karr rahe ho warna mai kuwara marr jaunga	0	0	0	-1	-0.41	0	10	1	10	9	7
@Mariam_Jamali Nice one but logo filhal KK ki jaga Pakistan ka lagwa do. Pic is good	4.6	9.7	4.7	-0.28	-0.37	6	6	8	8	7	9
abe .,., joke marna hai hi to aur hi kahi maar .,.,. confession page ki bejaati maat ker bhai .. JOKE MARA???????????? HASU? \"haha..!\"	2	6.67	4.28	-0.08	-0.18	6	5	3	8	7	5
Wale log jante hai par atankwadiyo nafrat failane walo ke liye meri yehi language rahegi	0.6	1.42	1.42	0.09	0	4	6	2	8	7	6
mujhe hasi aa rahi thi , while I ws reading them . :P	5	9.32	2.5	-0.24	-0.64	10	10	9	10	6	6
laufed ... first u hav to correct ur english baad me sochna use !!!	3.33	6.67	3.07	0.2	-0.06	10	8	8	9	6	7
The ultimate twist Dulhan dandanate huye brings Baraat .... Dulha	4.44	6.9	5.55	-0.08	0.48	8	6	7	2	5	7
RAHUL jab dieting par hota hai toh green tea peeta hai.	3.63	6.61	5.45	-0.44	0	10	10	2	10	10	9

Table 2: Measuring the complexity of various Hindi-English code-mixed text. Language independent tokens are marked with black color. We select one sentence each from the 10 datasets (in the same order as given in Table 1). Here, DCM stands for degree of code-mixing and RA stands for readability. We scale the CMI, M-index, and I-index metric scores in the range 0 to 10. The range for Burstiness and Memory score is -1 to 1.

on two parameters: the degree of code-mixing and readability. We provide the following guidelines to the annotators for this task:

- **Degree of code-mixing (DCM)**: The 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 the high degree of code-mixing.

- **Readability (RA)**: RA score can vary between 0 to 10. A completely unreadable sentence due to large number of 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.

Table 2 shows the 10 example Hinglish sentences with the corresponding metric scores and the human evaluation. Some major observations are:

- We do not observe any metric to independently measure the readability of code-mixed text as quantified by humans.

- We also observe contrasting scores given by different metrics, making it difficult to choose the best-suited metric for the given code-mixed dataset.

- At times, we observe a high disagreement even among the human ratings. This behavior indicates the complexity of the task for humans as well.

- We do not observe any significant relationship between the degree of code-mixing and the readability score as provided by humans. This observation is critical in building high-quality datasets for various code-mixing tasks.

4 Limitations of code-mixing metrics

This section describes various limitations of the existing metrics that measure the complexity of the code-mixed text. As CMI is most popular among code-mixing metrics, it is reported in five (Patwa et al., 2020; Barman et al., 2014; Dhar et al., 2018; Srivastava and Singh, 2020; Khanuja et al., 2020) out of the 10 datasets listed in Table 1. We describe major limitations of code-mixing metrics from three different perspectives:

1. **Metric formulation**: Most of the code-mixing metrics are based on the word frequency from different languages in the text.

Data source	Spelling variations	Noisy/monolingual	Readability/semantic
Singh et al. (2018c)	Ab boliye teen talak harram h ya nai aapke khud ki lady's chate h ki aap sai dur hona. Shame on u again...#TripleTalaq	#TripleTalaq Don't post this	@BJP4UP @narendramodi @AmitShah @BJPLive @bjpsamvad @BJP4India #NoteBandi ke baad ab poori
Swami et al. (2018)	Shareef wo hai jisay moqa nae milta! #irony	Resigned: Sri Lanka Cricket aniyin thodar thoalviyinaal Therivuk kuluth Thalaivar Sanath Jayasuriya ullitta athan uruppinarhal Raajinaamaa	Kudakudhinge dhuvasthamee? #Maldives #Politics
Joshi et al. (2016)	Nhi ye log apny lie ayeen change karty he ye konsa mulk k lie sochty he har koi apny lie aur apny families k lie politics me he sary chor he	#Cricket News 6 Saal Team Ki Qeyadat Karna Mare Liye Izaz Hai Ab Kisi Our Ko Aagay Aana Chahiye Sabiq Captain AB De Villiers	Hiii kam chhe
Patwa et al. (2020)	@DivyanshMohit @GulBukhari Tum apny Indian ki fikkar Karo Pakistan ko hum khud dykh lyngy. Mukti bahini 2 nahi ban	@BTS_army_Fin Also Stade de France is preparing for the concert. Looks so beautiful! See their post on Instagram https//t.co/OwhP	Now this i too much ab sare tweet arsal ke support me Jab jiya ka man nhi and wo chai nhi bana sakti yasit ke liy
Barman et al. (2014)	@Liaqat842 tum sahi keh rhy thy yeh zayda buri timings hain 3 wali match ki subah purany office bhi jna hai kaam hai	@saadiaafzaal Pagl he ye Qaom Jo misbah ka Cmprezm imraan se kr rhe he. khuda ko maano kaha misbah kaha imran.. shoib Akhtar	@aashikapokharel Haha okay. Office time, aba bus ma bore hune wala chhu. Also, Alumni ko imp kaam chha. Viber ma aaye hune. :P
Bohra et al. (2018)	Gf khoon peene k liye hoti hai aur apne babu ko thana thilane k liye bas	Mere marnay ki ya hate deni ki?	ke karya karta aise hi baithe hai.kal ye ghatna aap or Hum
Dhar et al. (2018)	Modi ji aap jesa koi nhi dhanywad aap desh ki kitni sewa karte hai jese ak beta apni ma ko poojta hai	Girna samal nai lage	Etni lambi speech sa kuch mi hotta sirf 2 word khna or unka suna sa frk atta h........ sekho i love you sallu??
Srivastava and Singh (2020)	unhone pehle pic ni dkhi ti kya tmhari jo milne k baad hi ignore kia tmhe...?	kaun hai ye zaleel insaan?	@indiantweeter Jain ration gap ho jaega.
Vijay et al. (2018)	35 sal ma koi hospital esa nai banaya jaha khud ka ilaj hosakai. .. Irony	and then the irony,, sab ko jurisakyo lahana le kahile juraucha ?	hi Vanitha Garu hai Andi this is irony , arledy rep icharu ga
Khanuja et al. (2020)	3 kam padey they	KASTURI is speaking to his son	31 minutes time hua

Table 3: Examples from the 10 datasets highlighting the various inherent limitations that could lead to misleading code-mxing metric score. For the marked words in **spelling variations**, we observe multiple spellings across datasets. We observe that the **noisy** sentences have low **readability**.

This formulation makes the metric vulnerable to several limitations, such as the bag-of-words model and assigning higher metric scores to meaningless sentences that are difficult to read and comprehend.

2. **Resource limitation**: The existing code-mixed datasets too have several shortcomings, such as noisy and monolingual text (see Table 3). Besides, we observe the poor quality of the token-level language identification (LID) systems which are fundamental in calculating the various code-mixing metric scores.

3. **Human annotation**: In the absence of good quality code-mixed LID systems, various works employ human annotators to perform language identification. Evaluating human proficiency is a challenging task since code-mixed languages lacks standard syntax and semantics. Additionally, human annotation is a time and effort extensive process.

Next, we describe four major limitations that combine one or more than one perspective (see Table 4). Figure 2 shows a general flow diagram to obtain the code-mixed data from the large-scale noisy text. It shows the three major bottlenecks (metric formulation, resource limitation, and human annotation) in the entire data filtering process. The resultant code-mixed data is noisy and suffers from several other limitations (see Table 3).

Limitation	Perspective
Bag of words	MF
Code-mixed LID	MF, RL
Misleading score	MF, RL, HA
High inference time	MF, RL, HA

Table 4: Combination of perspectives for each of the limitation to code-mixing metrics. Here, MF: Metric Formulation, RL: Resource Limitation, HA: Human Annotation.

1. **Bag-of-words**: None of the code-mixing metrics consider inherent ordering between the

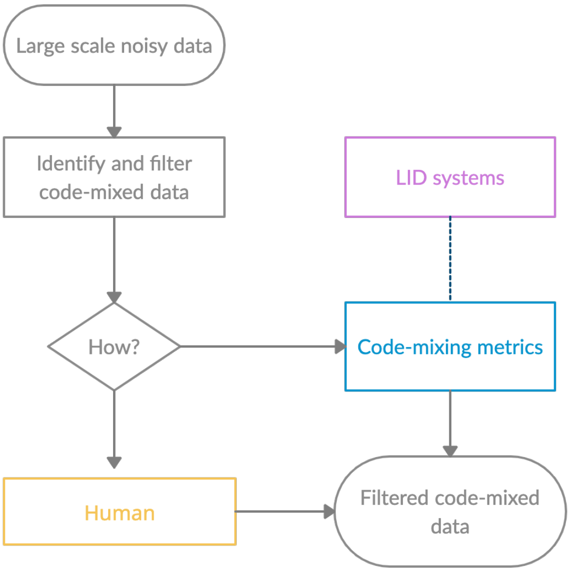

Figure 2: A general flow diagram for identifying and filtering the code-mixed data from the large scale noisy text. We observe three major limitations: metric formulation, resource limitation, and human annotation. There is a time-quality trade-off between the two paths to filter the code-mixed data. Employing humans takes more time and relatively better quality code-mixed sentences as compared to code-mixing metrics that takes less time and shows poor performance.

Pair I

HINGLISH1 (Khanuja et al., 2020): FATHER ab hospital mein hi rahenge.
HINGLISH2 (shuffled HINGLISH1) : hospital ab FATHER mein hi rahenge.
Observation: Same CMI, M-index, I-index, Burstiness and Memory scores

Pair II

HINGLISH1 (Bohra et al., 2018): TAJMAHAL p jake atka hai
HINGLISH2 (shuffled HINGLISH1) : atka p jake TAJMAHAL hai
Observation: Same CMI, M-index, and I-index scores

Figure 3: Example to demonstrate the *bag of words* assumption of code-mixing metrics. We shuffle tokens in HINGLISH1 to get HINGLISH2. Observation shows that metric scores remain unchanged after the shuffling while the semantic of the original sentence is lost.

words in the code-mixed sentence[1]. This limitation makes these metric scores vulnerable to multiple challenges, such as poor grammatical structure. Figure 3 shows examples of good quality code-mixed sentences and corresponding noisy sentences, both having the same metric scores.

2. **Code-mixed language identification**: The presence of more than one language in the code-mixed text presents several challenges for the various downstream NLP tasks such as POS tagging, summarization and named entity recognition. Identifying the token-level language of the code-mixed text is the fundamental step in calculating the code-mixing metric scores. Often various works have employed human annotators to obtain the token-level language tags. However, both human annotators and the language identification systems suffer from the poor token-level language tagging. Table 5 shows the variation in the output of five multilingual/code-mixed LID systems (Langdetect[2], Polyglot[3], CLD3[4], FastText[5], and iNLTK[6]) on the code-mixed text against human-annotated language tags. Contrasting human-annotated tag sequence, the same metric yields significantly different scores due to variation in the language tag sequence obtained from different LID tools. We identify three major reasons for the poor performance of humans and the LID systems in identifying the language of the code-mixed text:

   - **Spelling variations and non-contextual LID**: Spelling variation is one of the most significant challenges in developing code-mixed LID systems. Due to the lack of standard grammar and spellings in code-mixed language, we observe multiple variations of the same word across datasets (see Table 3). For example, Hindi tokens *'hn'* or *'hay'* can also be written as *'hun'* or *'hai'*, respectively. As outlined in Table 5, we observe incorrect language identification by popular multilingual and code-mixed LID systems. This behavior could be highly attributed to the spelling

[1] Note that, *Burstiness* and *Memory* metric only considers span length and not the word ordering within a span.

[2] https://pypi.org/project/langdetect/

[3] https://github.com/aboSamoor/polyglot

[4] https://github.com/google/cld3/

[5] https://fasttext.cc/blog/2017/10/02/blog-post.html

[6] https://inltk.readthedocs.io/en/latest/index.html

	@user	bus	office	me	hn	,	Sat	thora	thanda	hota	hay	kaam	k	point	of	view	say	you	know	:)
Langdetect	et	id	en	nl	vi	unk	tl	en	en	cs	so	so	sw	fi	en	af	tl	sw	en	unk
Polyglot	en	en	en	en	da	un	en	en	en	to	es	fy	en	en	en	en	en	en	en	un
CLD3	no	la	ja	mi	sv	ja	sd	la	ko	mi	es	et	sl	de	en	en	id	en	en	ja
FastText	en	en	en	en	en	ru	pt	war	en	en	es	az	ja	en	en	en	en	en	en	uz
iNLTK	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en	en
Human	univ	en	en	hi	hi	univ	en	hi	hi	hi	hi	hi	hi	en	en	en	hi	en	en	univ

Table 5: Example to demonstrate the limitations of LID systems in calculating the code-mixing metric scores. Hinglish sentence is from the dataset used in (Barman et al., 2014). The language name corresponding to the language code can be found at the corresponding LID system's web page.

Token	@	nehantics	Haan	yaar	neha	kab	karega	woh	post	Usne	na	sach	mein
Language	O	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Hin
Token	photoshoot	karna	chahiye	phir	woh	post	karega	. . .	https	//	tco	/	5RSlSbZNtt
Language	Eng	Hin	Hin	Hin	Hin	Hin	Hin	O	Eng	O	Eng	O	Eng

(a) Example sentence from Patwa et al. (2020)

Token	are	cricket	se	sanyas	le	liya	kya	viru	aur	social	service	suru
Language	Hin	Eng	Hin	Hin	Hin	Hin	Hin	Hin	Hin	Eng	Eng	Hin
Token	kardiya	.	khel	hi	bhul	gaye	.	2	innings	0	n	0
Language	Hin	O	Hin	Hin	Hin	Hin	O	O	Hin	O	Hin	O

(b) Example sentence from Swami et al. (2018)

Table 6: Example sentences to demonstrate the limitations with the language tags in the current code-mixed datasets. We use the color coding to represent three major reasons for such behaviour: ambiguous, annotator's proficiency, and non-contextual. 'O' in the language tag represent the tag 'Other'.

variation of words. Additionally, the non-contextual language tag sequence generation by LID systems and humans leads to a similar set of challenges (see Table 6). In both the examples in Table 6, we observe the incorrect language tag to words like *'tco'* and *'n'* due to the missing context by the human annotator. Also, as observed in Table 6, incorrect LID by humans could be attributed to considering the code-mixed tokens out of context.

- **Ambiguity**: Ambiguity in identifying named-entities, abbreviations, community-specific jargons, etc., leads to incorrect language identification. Table 6 shows the example sentences having incorrect language tags due to ambiguity in the code-mixed sentences. For example, tokens like *'nehatics'*, *'neha'*, and *'viru'* are person named-entities, incorrectly tagged with *hi* tag.

- **Annotator's proficiency**: Evaluating the human proficiency for a code-mixed language is much more challenging as compared to the monolingual languages due to lack of standard, dialect variation, and ambiguity in the text. Table 6 shows an example of incorrect language annotation by the human annotators, which could be attributed to low human proficiency/varied interpretation of the code-mixed text. For example, English tokens like *'post'* and *'innings'* are tagged as *hi* tokens by human annotators.

3. **Misleading score**: We observe several inconsistencies in the interpretation of the code-mixing metric scores. We identify three major reasons for this inconsistent behavior:

   - **Coherence**: Coherency in a multi-sentence code-mixed text is one of the fundamental properties of good quality data. Future works in code-mixed NLP, such as text summarization, question-answering, and natural language inference, will require highly coherent datasets. However, the current metrics cannot measure the coherency of the code-mixed text. We witness a large number of real scenarios where the code-mixing metric scores for multi-sentence text are high, but the coherency is very poor. In such cases, the code-mixing metrics in the present form will lead to undesirable behavior. For instance, we query a Hinglish question-answering system *WebShodh*[7] (Chandu et al., 2017) with the question: *India ka PM kaun hai? Cricket dekhne jaana hai?* The list of eight probable answers (*'ipl', 'puma', 'kohli', 'sports news feb', ''amazoncom', 'sport news nov', 'hotstar vip', 'rugged flip phone unlocked water shock proof att tmobile metro cricket straight talk consumer cellular carrier cell phones'*) shows the poor performance of the system due to low coherency in the question text (in addition to other architectural limitations) even though the question text is highly code-mixed on various metrics.

- **Readability**: The co-existence of the code-mixed data with the monolingual and the noisy text results in the poor readability of the code-mixed text. The code-mixing metrics do not take into account the readability of the code-mixed text. Low readability of the code-mixed text will also lead to incorrect annotations by the annotators, which will eventually lead to incorrect metric scores for the given data. Table 3 shows example sentences from multiple datasets with low readability.
- **Semantics**: The last column in Table 3 shows example sentences from multiple datasets where it is extremely difficult to extract the meaning of the code-mixed sentence. Due to the current formulation of the code-mixing metrics where we consider the independent language tokens and the bag-of-words approach, it is not feasible to identify such low semantic sentences.

4. **High inference time**: We require an efficient automatic NLP system that identifies and filters the code-mixed text from a large-scale noisy text or monolingual text. Even though theoretically, the code-mixing metrics can help identify text with high levels of code-mixing, but practically they fail due to inefficiencies in LID systems. We showcase the inability of LID systems to detect correct language tags (see point 2 above). One possible remedy is to employ humans in language identification. However, human involvement significantly increases the time and the cost of performing the labeling task. Also, human annotations are also prone to errors (see Table 6). We might also need task-specific annotations (e.g., POS tags, NER, etc.) which will further increase the time and cost of the annotation task. Due to this reason, we see majority of the datasets (see Table 1) relatively smaller in size (<5000 data points). Human annotation significantly increases the inference time in calculating the code-mixing metric scores.

5 Conclusion and Future Work

In this paper, we extensively discuss the limitations of code-mixing metrics. We explored 10 Hinglish datasets for presenting examples to support our claims. Overall, we showcase the need for extensive efforts in addressing these limitations. In the future, we plan to develop a robust code-mixing metric that measures the extent of code-mixing and quantifies the readability and grammatical correctness of the text. Also, we aim to create a large-scale Hinglish dataset with manual token-level language annotation.

References

Scott Baldauf. 2004. A hindi-english jumble, spoken by 350 million. *The Christian Science Monitor*, 1123(1):3.

Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. The first cross-script code-mixed question answering corpus. In *MultiLingMine@ ECIR*, pages 56–65.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Doha, Qatar. Association for Computational Linguistics.

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.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of hindi-english code-mixed social

[7] http://tts.speech.cs.cmu.edu/webshodh/cmqa.php

media text for hate speech detection. In *Proceedings of the second workshop on computational modeling of people's opinions, personality, and emotions in social media*, pages 36–41.

Eyamba G Bokamba. 1989. Are there syntactic constraints on code-mixing? *World Englishes*, 8(3):277–292.

Khyathi Chandu, Ekaterina Loginova, Vishal Gupta, Josef van Genabith, Günter Neumann, Manoj Chinnakotla, Eric Nyberg, and Alan W Black. 2019. Code-mixed question answering challenge: Crowdsourcing data and techniques. In *Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38. Association for Computational Linguistics (ACL).

Khyathi Raghavi Chandu, Manoj Chinnakotla, Alan W Black, and Manish Shrivastava. 2017. Webshodh: A code mixed factoid question answering system for web. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 104–111. Springer.

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.

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.

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.

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.

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.

Simran Khanuja, Sandipan Dandapat, Sunayana Sitaram, and Monojit Choudhury. 2020. A new dataset for natural language inference from code-mixed conversations. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 9–16.

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.

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.

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018c. Named entity recognition for hindi-english code-mixed social media text. In *Proceedings of the seventh named entities workshop*, pages 27–35.

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.

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*.

Mary WJ Tay. 1989. Code switching and code mixing as a communicative strategy in multilingual discourse. *World Englishes*, 8(3):407–417.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset for detecting irony in hindi-english code-mixed social media text. *EMSASW@ ESWC*, 2111:38–46.

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.