Supervised and Unsupervised Evaluation of Synthetic Code-Switching

Evgeny Orlov
HSE University, Moscow, Russia
emorlov@edu.hse.ru

Ekaterina Artemova
HSE University, Moscow, Russia
Huawei Noah's Ark lab, Moscow, Russia
elartemova@hse.ru

Abstract

Code-switching (CS) is a phenomenon of mixing words and phrases from multiple languages within a single sentence or conversation. The ever-growing amount of CS communication among multilingual speakers in social media has highlighted the need to adapt existing NLP products for CS speakers and lead to a rising interest in solving CS NLP tasks. A large number of contemporary approaches use synthetic CS data for training. As previous work has shown the positive effect of pretraining on high-quality CS data, the task of evaluating synthetic CS becomes crucial. In this paper, we address the task of evaluating synthetic CS in two settings. In supervised setting, we apply Hinglish finetuned models to solve the *quality rating prediction* task of HinglishEval competition and establish a new SOTA. In unsupervised setting, we employ the method of acceptability measures with the same models. We find that in both settings, models finetuned on CS data consistently outperform their original counterparts.

1 Introduction

Code-switching (CS) is a phenomenon of mixing words and phrases from multiple languages within a single sentence or conversation[1]. It is common for multilingual speakers and happens across various language pairs across the globe, such as Spanish-English (Spanglish) and Hindi-English (Hinglish). Various studies (Baldauf, 2004) have predicted the high growth in the number of CS speakers, which would surpass the number of native speakers in various globally popular languages (e.g., English).

The advent of social media has highlighted the amount of CS communication and lead to a further increase of the number of multilingual speakers who use this pattern. This availability of CS data and the understanding that existing NLP products need to be adapted for the ever-growing number of CS speakers has resulted into a rising interest in various CS NLP tasks. Work has been done in such tasks as LID (Shekhar et al., 2020; Singh et al., 2018a; Ramanarayanan et al., 2019; Barman et al., 2014; Gundapu and Mamidi, 2020), POS tagging (Singh et al., 2018b; Vyas et al., 2014; Pratapa et al., 2018b), NER (Singh et al., 2018a; Priyadharshini et al., 2020; Winata et al., 2019a), word normalisation (Singh et al., 2018c; Parikh and Solorio, 2021), sentiment analysis (Patwa et al., 2020; Joshi et al., 2016), NLI (Khanuja et al., 2020a), machine translation (Srivastava and Singh, 2020; Dhar et al., 2018) and QA (Chandu et al., 2019; Thara et al., 2020).

Various studies have shown that CS data may pose a challenge for contemporary multilingual models (Birshert and Artemova, 2021). Finetuning on CS data can alleviate this problem (e.g. Ansari et al., 2021). As social media can be noisy and not readily available to build a large scale corpus, various techniques of generating synthetic CS have been proposed (see Section 2). However, it was shown that the performance of the models crucially depends on the quality of CS text used for pretraining (Santy et al., 2021). This creates the task of synthetic CS evaluation which is the main focus of current paper.

CS evaluation methods range from computing intrinsic text metrics to measuring downstream task performance depending on the CS data used for pretraining and human evaluation (see Section 2). Srivastava and Singh (2021a) show that most CS evaluation metrics fail to capture the linguistic diversity which leads to poorly estimating the quality of CS text. Thus, human evaluation remains as a reliable method. Srivastava and Singh (2021b) propose HinGE, a dataset of Hinglish sentences with human quality ratings and organise HinglishE-

[1] Some works make a distinction and refer to intrasentential (within a single sentence) code alternation as "code-mixing" (CM) and intersentential (at or above the sentence level) as "code-switching" (CS). It is also common, however, to use the term "CS" for both cases. Intrasentential code alternation is the focus of this paper and we refer to it as "CS".

Proceedings of the 2022 COLING Workshop: The 8th Workshop on Noisy User-generated Text (W-NUT 2022), pages 113–123

val shared task based on it (Srivastava and Singh, 2021c). In our paper, we address HinglishEval *quality rating prediction* task with Hinglish models proposed in Nayak and Joshi (2022). Moreover, we add an unsupervised setting of the task. Our main contributions are:

- We perform a series of experiments on unsupervised CS evaluation, employing the method of acceptability measures (Lau et al., 2015). To our knowledge, this is the first such attempt.
- We perform a series of experiments on supervised CS evaluation and establish a new SOTA for HinglishEval *quality rating prediction* task.
- We find that models finetuned on CS data consistently outperform their original counterparts.

2 Related works

Generating synthetic CS As large amounts of real-world CS data may be difficult to extract, various generating methods have been proposed. Simplistic methods include re-writing of some words in the target script (Gautam et al., 2021) and various rule-based algorithms used as baselines in the literature (e.g., Tarunesh et al., 2021; Srivastava and Singh, 2021b). The vast majority of methods utilize machine translation engines (Singh et al., 2019), parallel datasets (Jawahar et al., 2021; Gautam et al., 2021; Gupta et al., 2021; Winata et al., 2019b) or bilingual lexicons (Tan and Joty, 2021) to replace the segment of the input text with its translations. Bilingual lexicons may be induced from the parallel corpus with the help of soft alignment, produced by attention mechanisms (Lee and Li, 2020; Liu et al., 2020). Pointer networks can be used to select segments for further replacement (Gupta et al., 2020; Winata et al., 2019b). If natural CS data is available, such segments can be identified with a sequence labeling model (Gupta et al., 2021). A number of works employ popular architectures like VAE (Samanta et al., 2019) and GANs (Garg et al., 2018; Chang et al., 2019). Other methods produce synthetic CS text that grammatically adheres to a linguistic theory of code-switching. Pratapa et al. (2018a) leverage the equivalence constraint (EC) theory (Poplack, 1980), while Rizvi et al. (2021) use EC and Matrix-language (Carol, 1993) theories.

Evaluating synthetic CS Despite the practical need of synthetic CS datasets, the task of evaluating synthetic CS remains relatively understudied. Some evaluation techniques involve estimating intrinsic text properties, such as code-switching ratio and length distribution. One of the most popular metrics is code-mixing index (CMI) (Das and Gambäck, 2014; Gambäck and Das, 2016), which accounts for code-switching ratio and the number of switches in a sentence. We defer to Srivastava and Singh (2021a) for a detailed overview of other metrics used for evaluating CS NLG.

Further, extrinsic measures can be used, like the perplexity of external language model. For example, Nayak and Joshi (2022) propose a finetuned Hinglish GPT model and suggest using it for evaluation. Also, downstream task performance can be measured, depending on the CS data used for augmentation (Samanta et al., 2019; Santy et al., 2021). Downstream tasks are organised into benchmarks such as GLUECoS (Khanuja et al., 2020b) and LinCE (Aguilar et al., 2020) which comprise data for popular language pairs like English-Hindi and English-Spanish.

Finally, human evaluators can be employed to assess the quality of the generated CS. There are examples of such evaluation studies in the literature which are usually performed to prove the quality of the proposed CS generation method (Bhat et al., 2016; Tarunesh et al., 2021). However, these studies are of low scale and do not result into substantial datasets which can be used in further research. In this context, HinGE dataset (Srivastava and Singh, 2021b) is unique being the largest collection of synthetic CS with human ratings to date. It is described in detail in Section 3.1. Based on HinGE, HinglishEval competition was organised (Srivastava and Singh, 2021c; see Section 3.1.1), where the task is to model the annotators' opinion on CS sentences.

Language models for CS Along with the development of language models (LMs), work has been done to adapt them for CS data. Chan et al. (2009) compare different n-gram LMs, Vu et al. (2012) suggest to improve language modeling by generating artificial CS text. A number of works propose LMs that incorporate a syntactic constraint (Li and Fung, 2012, 2014; Pratapa et al., 2018a). Another line of papers introduce LMs where the output layer is factorized into languages, and POS tags are added to the input (Adel et al., 2013a,b, 2014, 2015; Sreeram and Sinha, 2017).

With the advent of Transformers (Vaswani et al., 2017), work has shifted to applying popular architectures to CS data. Pires et al. (2019) show that m-BERT can achieve promising results in Hinglish downstream tasks when Hindi parts are written in Devanagari even in a zero-shot setup. The same, however, does not apply to romanized Hinglish, as m-BERT was pretrained on Devanagari Hindi. Both GLUECoS (Khanuja et al., 2020b) and LinCE (Aguilar et al., 2020) benchmarks provide m-BERT baselines for their leaderboards. Ansari et al. (2021) show that BERT models produce better results in CS LID when pretrained on CS sentences rather than on multiple monolingual corpora. Santy et al. (2021) find that finetuning m-BERT on natural CS data gives the best performance improvement compared to any synthetic CS. Nayak and Joshi (2022) present the first large-scale (52.93M sentences) corpus of real Hinglish CS scraped from Twitter and a line of Transformer models finetuned on it. The corpus and the models are described in detail in Section 4.1.

Acceptability measures Lau et al. (2015) present the task of unsupervised prediction of speakers' acceptability judgements and propose *acceptability measures* as a method to translate LM's probability into acceptability scores. Acceptability measures are variants of the sentence's log probability, devised to normalise sentence length and low frequency words (see Section 4.2 for additional details and equations). The effectiveness of an acceptability measure is evaluated by computing its Pearson correlation with human acceptability scores. Lau et al. (2020) further experiment with Transformer LMs and investigate the dependence of acceptability measures' scores on whether the context of the sentence is provided.

3 Data

3.1 HinGE

HinGE is a dataset of synthetic Hinglish sentences with human quality ratings proposed in Srivastava and Singh (2021b). The dataset consists of firstly, parallel English and Hindi sentences. Second, two synthetic Hinglish sentences are generated from each pair of parallel sentences by two rule-based code-mixed text generation (CMTG) algorithms:

- Word-aligned CMTG (WAC): Noun and adjective tokens are aligned between the parallel sentences. The aligned Hindi token is replaced with the corresponding English token.
- Phrase-aligned CMTG (PAC): Key-phrases of length up to three tokens are aligned between the parallel sentences. The aligned Hindi phrase is replaced with the corresponding English phrase.

For both algorithms, the Hindi parts are then transliterated into the Roman script.

Label	# sentences	Binary label	# sentences
1	0	0	2279
2	9		
3	61		
4	250		
5	394		
6	633		
7	932		
8	960	1	1673
9	587		
10	126		
Total #	3952		

Table 1: Hinge All classes statistics

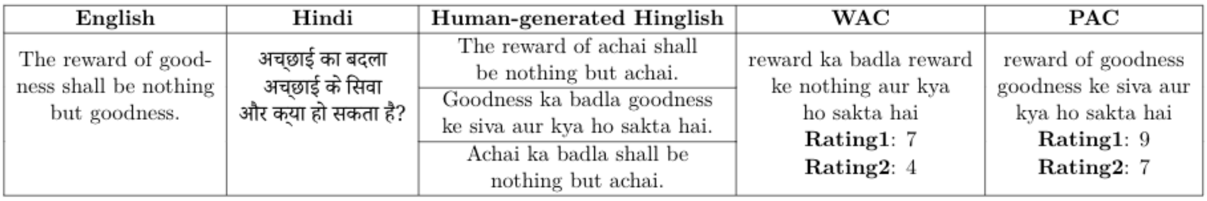
Third, an average of two human quality ratings on a scale of 1-10 is assigned to each synthetic Hinglish sentence. Refer to Table 1 for class balance information.

Fourth, annotators' disagreement is given, which is calculated as the absolute difference between the human quality ratings and ranges 0-9. Finally, for each pair of parallel sentences, at least two human-generated Hinglish sentences are provided. Figure 1 demonstrates an example of the described fields of the dataset.

Overall, HinGE contains 1976 parallel Hindi–English, 3952 synthetic CS and 4803 human-generated CS sentences. All synthetic CS sentences have human scores assigned to them, and HinGE is the largest such dataset to date. We refer to the synthetic part of the dataset as *Hinge All*.

3.1.1 HinglishEval competition

The authors also organized HinglishEval shared task based on the HinGE dataset (Srivastava and Singh, 2021c), which includes two subtasks: *quality rating prediction* and *annotators' disagreement prediction*. Both are classification tasks, but are evaluated with MSE in addition to weighted F1-score. Besides, Cohen's Kappa (CK) is computed for *quality rating prediction*. The dataset is split in the ratio 70:10:20 with 2766, 395 and 791 synthetic CS sentences in train, validation, and test, respectively. We refer to this dataset as *HinglishEval*.



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. Goodness ka badla goodness ke siva aur kya ho sakta hai. Achai ka badla 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

Figure 1: Example pair of parallel sentences with corresponding human-generated and synthetic CS from HinGE dataset. Picture from Srivastava and Singh (2021b).

For both tasks, the participants can use all the data in HinGE, including the English, Hindi and human-generated Hinglish sentences. Participants are also asked to implicitly answer questions about the reasons influencing the quality of synthetic CS. We seek to answer some of these in our work.

3.2 TCS

The dataset we refer to as *TCS* is a collection of 750 Hinglish sentences with human scores from Tarunesh et al. (2021). It contains Hinglish sentences from five sources (250 sentences each): human-generated CS, two rule-based algorithms, and supervised and unsupervised versions of the Transformer-based generation method proposed in Tarunesh et al. (2021). Each Hinglish sentence is provided with an average of three human scores on a scale of `1-5` under three heads: "Syntactic correctness","Semantic correctness" and "Naturalness". For our experiments, we also take the average of these three scores under the name of "Mean human score".

The original TCS sentences have their Hindi parts in Devanagari script, and we refer to this dataset as *TCS Devanagari*. We also transliterate the sentences into Roman script using `indic-transliteration` library[2] with ITRANS scheme[3] and refer to this dataset as *TCS transliterated*.

4 Experimental setup

4.1 Models

This subsection describes the LMs we experiment with in this work. All of them are taken from the Hugging Face Hub[4]. First, we employ a line of popular Transformers architectures: **BERT** (Devlin et al., 2018); **CoLA BERT**, a BERT model trained on CoLA dataset (Warstadt et al., 2019) and released by Morris et al. (2020); **XLM-RoBERTa** (Conneau et al., 2019); **m-BERT** (Devlin et al., 2018); **GPT-2** (Radford et al., 2019); and **mGPT** (Shliazhko et al., 2022).

Further, we employ Hinglish LMs introduced in Nayak and Joshi (2022). All of them are trained on L3Cube-HingCorpus proposed in the same paper. L3Cube-HingCorpus was collected as follows. First, CS sentences were filtered from continuously scraped tweets using a shallow subword-based LSTM LID classifier which was iteratively improved as the dataset increased. Then a BERT LID classifier was finetuned on the resulting 44455 sentences and was further used to collect the main corpus. The final dataset contains 52.93M sentences (1.04B tokens) of natural Hinglish CS. A Devanagari version of the dataset was created using an in-house transliteration model. Here we list the finetuned Hinglilsh models with their original counterparts in parentheses: **HingBERT** (BERT), **HingMBERT** (m-BERT), **HingRoBERTa** (XLM-RoBERTa), **HingGPT** (GPT-2). There are also two mixed versions of the models, which are pretrained on both Devanagari and roman scripts (**HingMBERT-mixed** and **HingRoBERTa-mixed**), and a model which is trained completely on Devanagari script (**HingGPT-devanagari**).

4.2 Unsupervised approach

We employ the concept of acceptability measures proposed in Lau et al. (2015) to assess the quality of CS in both TCS datasets and Hinge All. Table 2 presents equations for different acceptability measures. Of all the methods, we compute only *LP*, *MeanLP*, and *PenLP*, as *NormLP* and *SLOR* require an additional unigram LM. It should not be oversignificant, however, because for considered models (BERT and GPT-2) the best performance

[2] `https://github.com/indic-transliteration/indic_transliteration_py`

[3] ITRANS scheme showed the best performance scores among others in our preliminary experiments. See `https://www.aczoom.com/itrans/`

[4] `https://huggingface.co/models`

Acc. Measure	Equation
LP	$\log P(s)$
MeanLP	$\frac{\log P(s)}{\|s\|}$
PenLP	$\frac{\log P(s)}{((5+\|s\|)/(5+1))^{\alpha}}$
NormLP	$-\frac{\log P(s)}{\log P_{\mathrm{u}}(s)}$
SLOR	$\frac{\log P(s) - \log P_{\mathrm{u}}(s)}{\|s\|}$

Table 2: Acceptability measures for predicting the acceptability of a sentence. $P(s)$ is the sentence probability, computed by a LM; $P_{\mathrm{u}}(s)$ is the sentence probability estimated by a unigram LM; and $\alpha = 0.8$.

was mostly achieved by *PenLP* in the original paper (Lau et al., 2020). To compute the acceptability measures of considered Transformer models, we rely on the code from Lau et al. (2020). To evaluate the effectiveness of each acceptability measure, we compute its Pearson correlation with human acceptability scores in our datasets.

4.3 Supervised approach

We also run our models in a supervised setting on HinglishEval data, particularly the *quality rating prediction* task. As the original 10-way classification task has proved to be quite difficult in our preliminary experiments and the results of the competition, we add two simplified versions of it:

- *Binary classification*: We binarize the labels (`1-7` are converted to `0` and `8-10` to `1`[5]) and perform binary classification. Classes numbers are given in Table 1.
- *Regression*: We perform regression on the original labels. MSE is computed with the models' initial predictions, while the predictions for F1-score and CK are rounded.

All models are trained for 5 epochs with a learning rate of 2e−5, batch size of 32. The best model is then chosen with validation F1-score. For all models, we repeat training 10 times with 10 different seeds (0–9, respectively). We report mean and standard deviation of all metrics over 10 runs.

5 Results

5.1 Unsupervised approach

Acceptability measures' performance on TCS Devanagari and TCS transliterated is given in Tables 3 and 4, respectively. For both versions of TCS, among the three scales, the highest correlations are achieved with *Mean syntactic correctness* score, which may indicate that syntax structure is the easiest for the models to grasp.

For TCS Devanagari, predictably, a substantial advantage is on the side of the models which were exposed to Devanagari during pretraining (mBERT, mGPT, HingMBERT-mixed, and HingGPT-devanagari). The best *Mean human score* correlations are shared by HingMBERT-mixed and notably mGPT which was not pretrained on any CS data.

For TCS transliterated, multilingual models cannot rely on their Devanagari knowledge. HingBERT is a clear leader, as it was exposed to romanized Hinglish during pretraining. Overall, the correlations of Hinglish models are lower than on TCS Devanagari. A possible explanation could be that the transliteration scheme we used to transliterate TCS differs from the way Hinglish is written on social media, whose data was used to finetune Hinglish models.

Acceptability measures' performance on Hinge All is given in Table 5. Here, the best correlations are also predictably achieved by the models which were finetuned on Hinglish CS data.

Comparing different acceptability measures with each other, we observe that unnormalized *LP* works quite well, but is usually outperformed by *PenLP*. In general, however, unidirectional (GPT-like) models benefit more from normalization. These observations support the findings of Lau et al. (2020). In general, we note that CS finetuned models consistently perform better than their original counterparts.

5.2 Supervised approach

Table 6 shows the results of 10-class classification on HinglishEval data. To be consistent with the participants of HinglishEval competition, we report both validation and test results and round the scores to thousandths. Here, HingMBERT-mixed achieves the best score and beats current SOTA (0.261) as reported in HinglishEval leaderboard[6]. It outperforms HingMBERT, although all Hindi data in HinGE is romanized.

Although the best model for regression (see Table 7) is still chosen based on F1-score, this kind of

[5] We choose the boundary so that the classes are of relative sizes.

[6] `https://codalab.lisn.upsaclay.fr/competitions/1688`

model	Mean syntactic correctness			Mean semantic correctness			Mean naturalness			Mean human score		
	LP	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*
BERT	0.08	0.03	0.1	0.07	0.03	0.09	0.05	0.04	0.08	0.07	0.04	0.09
m-BERT uncased	0.33	0.13	0.28	0.31	0.12	0.26	0.28	0.12	0.25	0.31	0.13	0.26
m-BERT cased	0.28	0.15	0.26	0.26	0.14	0.24	0.24	0.14	0.24	0.26	0.15	0.25
GPT-2	0.09	0.16	0.31	0.08	0.16	0.29	0.06	0.16	0.28	0.08	0.16	0.3
mGPT	0.35	0.21	**0.41**	0.33	0.2	**0.39**	0.3	0.2	**0.37**	0.33	0.2	**0.39**
HingBERT	0	-0.08	-0.04	0	-0.07	-0.04	-0.02	-0.07	-0.06	-0.01	-0.08	-0.05
HingMBERT	0.08	-0.07	0.02	0.08	-0.07	0.02	0.07	-0.05	0.02	0.07	-0.06	0.02
HingMBERT mixed	**0.41**	0.28	0.39	**0.39**	0.27	0.37	**0.37**	0.27	0.36	**0.39**	0.28	0.37
HingGPT	-0.02	-0.18	-0.06	-0.03	-0.18	-0.06	-0.04	-0.19	-0.07	-0.03	-0.19	-0.07
HingGPT-devanagari	0.2	**0.31**	0.26	0.19	**0.3**	0.25	0.17	**0.29**	0.23	0.19	**0.3**	0.25

Table 3: Acceptability measures' correlations on TCS Devanagari

model	Mean syntactic correctness			Mean semantic correctness			Mean naturalness			Mean human score		
	LP	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*	*LP*	*MeanLP*	*PenLP*
BERT	0.03	-0.02	0.01	0.02	-0.03	0	0.01	0	0	0.02	-0.02	0
m-BERT uncased	0.02	-0.05	-0.01	0.01	-0.06	-0.02	0	-0.04	-0.01	0.01	-0.05	-0.01
m-BERT cased	0.01	-0.09	-0.04	0	-0.1	-0.05	0	-0.07	-0.04	0	-0.09	-0.05
GPT-2	0.03	0	0.04	0.02	0	0.02	0	0.01	0.02	0.02	0	0.02
mGPT	0.05	0.02	0.06	0.04	0.02	0.05	0.02	0.04	0.05	0.03	0.03	0.05
HingBERT	**0.18**	**0.2**	**0.22**	**0.16**	**0.18**	**0.2**	**0.16**	**0.21**	**0.22**	**0.17**	**0.2**	**0.22**
HingMBERT	0.15	0.12	0.19	0.13	0.11	0.17	0.13	0.13	0.18	0.14	0.12	0.18
HingMBERT mixed	0.16	0.13	0.2	0.14	0.12	0.18	0.14	0.14	0.2	0.15	0.13	0.2
HingGPT	0.07	0.02	0.07	0.06	0.01	0.06	0.04	0.04	0.06	0.06	0.02	0.07
HingGPT-devanagari	0.05	0.05	0.05	0.04	0.03	0.04	0.02	0.02	0.03	0.04	0.03	0.04

Table 4: Acceptability measures' correlations on TCS transliterated

model	*LP*	*MeanLP*	*PenLP*
BERT	0.19	-0.04	0.15
m-BERT uncased	0.19	-0.07	0.14
m-BERT cased	0.19	-0.08	0.14
GPT-2	0.19	-0.06	0.2
mGPT	0.2	-0.06	0.21
HingBERT	0.22	0.08	0.2
HingMBERT	0.22	0.1	0.21
HingMBERT mixed	**0.23**	0.1	0.21
HingGPT	0.2	0.1	**0.25**
HingGPT-devanagari	0.18	**0.11**	0.19

Table 5: Acceptability measures' correlations on Hinge All

problem statement allows to reduce the MSE score as compared to 10-class classification. A low MSE, however, does not lead to a higher F1-score. The best F1-scores are achieved by HingMBERT and HingRoBERTa, but are insufficient to overcome the level of 10-class classification.

Binarizaton of the problem (see Table 8) allows to significantly raise the F1-scores. The best result here is achieved by HingMBERT-mixed. We observe that CoLA BERT performs better than BERT base model, which may indicate transfer learning from English acceptability task.

We note that similarly with unsupervised setting, CS models consistently outperform their original counterparts in all supervised problem statements.

6 Discussion

Our experiments show that both in unsupervised and supervised setups, models pretrained on Hinglish data consistently outperform their original counterparts. This goes in line with previous studies which have shown that pretraining on CS data yields better results than monolingual pretraining (Santy et al., 2021; Ansari et al., 2021).

On HinglishEval 10-class classification, our HingMBERT-mixed establishes new SOTA, surpassing the m-BERT baseline from Srivastava and Singh (2021c) which was trained solely on Hinglish sentences from Hinge. Moreover, our Hinglish models trained solely on Hinglish sentences produce scores competitive with the participants of HinglishEval shared task which use all available information from HinGE (original Hindi and English sentences and annotators' disagreement; Furniturewala et al., 2022; Guha et al., 2022; Kodali et al., 2022; Singh, 2022).

6.1 Error analysis

In this subsection, we look for sources of errors of our best performing model, HingMBERT-mixed. We analyze its predictions on the test subset of HinglishEval 10-class classification. We put three

model	Val			Test		
	F1	CK	MSE	F1	CK	MSE
BERT	0.232±0.013	0.069±0.015	2.812±0.146	0.238±0.011	0.082±0.014	2.778±0.215
CoLA BERT	0.238±0.014	0.081±0.014	2.774±0.274	0.225±0.019	0.065±0.016	2.76±0.327
m-BERT uncased	0.255±0.016	0.102±0.013	2.867±0.193	0.238±0.016	0.086±0.014	2.826±0.115
m-BERT cased	0.245±0.015	0.08±0.02	2.944±0.215	0.237±0.013	0.078±0.017	2.878±0.149
XLMRoBERTa	0.229±0.014	0.081±0.02	2.957±0.187	0.203±0.013	0.045±0.016	2.878±0.194
GPT-2	0.216±0.013	0.056±0.018	3.182±0.173	0.204±0.017	0.036±0.022	3.175±0.243
HingBERT	0.253±0.005	0.106±0.007	2.689±0.123	0.248±0.012	0.101±0.015	2.839±0.134
HingMBERT	**0.262±0.015**	**0.11±0.015**	2.663±0.213	0.253±0.019	0.1±0.02	2.613±0.182
HingMBERT-mixed	0.253±0.014	0.1±0.02	**2.627±0.23**	**0.267±0.01**	**0.119±0.011**	**2.526±0.184**
HingRoBERTa	0.245±0.012	0.099±0.015	2.682±0.102	0.251±0.024	0.109±0.027	2.734±0.16
HingGPT	0.237±0.009	0.066±0.01	3.116±0.15	0.25±0.014	0.087±0.016	3.031±0.199
HingGPT-devanagari	0.209±0.006	0.051±0.008	3.29±0.141	0.196±0.016	0.037±0.018	3.195±0.154
m-BERT baseline	0.202	0.003	2.797	0.256	0.092	2.628

Table 6: 10-class classification results on HinglishEval. m-BERT baseline from Srivastava and Singh (2021c)

model	Val			Test		
	F1	CK	MSE	F1	CK	MSE
BERT	0.222±0.011	0.063±0.014	2.371±0.086	0.218±0.013	0.055±0.018	2.219±0.123
CoLA BERT	0.219±0.008	0.059±0.012	2.364±0.078	0.222±0.018	0.056±0.018	2.226±0.058
m-BERT uncased	0.223±0.011	0.06±0.016	2.341±0.065	0.215±0.009	0.051±0.015	2.213±0.079
m-BERT cased	0.217±0.006	0.049±0.011	2.391±0.08	0.215±0.008	0.05±0.011	**2.205±0.035**
XLMRoBERTa	0.189±0.007	0.019±0.012	2.453±0.05	0.197±0.009	0.033±0.011	2.396±0.063
GPT-2	0.211±0.012	0.042±0.015	2.411±0.077	0.22±0.013	0.053±0.011	2.246±0.054
HingBERT	0.232±0.016	0.069±0.016	2.359±0.14	0.244±0.016	0.081±0.018	2.331±0.149
HingMBERT	**0.239±0.024**	**0.083±0.028**	2.401±0.088	**0.25±0.014**	**0.093±0.015**	2.37±0.092
HingMBERT-mixed	0.226±0.03	0.066±0.037	2.437±0.146	0.235±0.025	0.075±0.026	2.388±0.154
HingRoBERTa	0.236±0.013	0.08±0.012	**2.276±0.133**	**0.25±0.02**	0.092±0.017	2.276±0.128
HingGPT	0.247±0.008	0.076±0.009	2.389±0.1	0.256±0.007	0.086±0.01	2.278±0.095
HingGPT-devanagari	0.194±0.008	0.027±0.014	2.625±0.188	0.191±0.014	0.027±0.014	2.545±0.228

Table 7: Regression results on HinglishEval

model	Val			Test		
	F1	CK	MSE	F1	CK	MSE
BERT	0.639±0.011	0.253±0.023	0.357±0.013	0.662±0.011	0.304±0.021	0.333±0.011
CoLA BERT	0.633±0.007	0.246±0.018	0.365±0.008	0.673±0.013	0.329±0.026	0.325±0.014
m-BERT uncased	0.646±0.011	0.27±0.025	0.353±0.011	0.648±0.01	0.278±0.022	0.348±0.01
m-BERT cased	0.626±0.015	0.23±0.031	0.371±0.016	0.637±0.01	0.254±0.021	0.359±0.01
XLMRoBERTa	0.623±0.025	0.222±0.058	0.371±0.019	0.639±0.015	0.258±0.036	0.356±0.013
GPT-2	0.612±0.02	0.211±0.041	0.388±0.022	0.619±0.016	0.228±0.026	0.379±0.019
HingBERT	0.665±0.007	0.324±0.02	0.336±0.007	0.648±0.015	0.287±0.026	0.353±0.016
HingMBERT	0.682±0.011	0.354±0.021	0.318±0.012	0.672±0.015	0.333±0.24	0.327±0.016
HingMBERT-mixed	0.682±0.008	0.353±0.014	0.318±0.009	**0.681±0.008**	**0.352±0.019**	**0.319±0.008**
HingRoBERTa	**0.689±0.013**	**0.369±0.026**	**0.312±0.013**	0.668±0.011	0.323±0.021	0.332±0.011
HingGPT	0.642±0.009	0.269±0.02	0.358±0.009	0.643±0.011	0.269±0.022	0.355±0.012
HingGPT-devanagari	0.574±0.01	0.116±0.021	0.42±0.011	0.609±0.008	0.193±0.017	0.383±0.012

Table 8: Binary classification results on HinglishEval

factor	**mean**		**statistically significant**
	correct	**incorrect**	
sentence length	17.0	19.2	✗
Hindi fraction	0.63	0.67	✓
# of switch points	5.5	6.1	✗

Table 9: Error source factors for HinglishEval 10-class classification, model is HingMBERT-mixed

factors under consideration: sentence length in words, fraction of Hindi words in a sentence and number of code switches within a sentence. To compute the latter two values, we annotate HinGE test subset with HingBERT-LID model proposed in Nayak and Joshi (2022). We compare the mean value of the factors depending on the correctness of model's prediction (see Table 9). We find that the mean of all three factors is greater for incorrect predictions, which means that the model tends to consistently make mistakes on more complex sentences. However, computing the t-test shows that only the difference in fraction of Hindi words is statistically significant. These results can be seen as an answer to the questions about the reasons influencing the quality of synthetic CS posed in (Srivastava and Singh, 2021c), e.g. "Does the dominance of a language (English or Hindi) present in the Hinglish sentence impact the rating provided by the humans?".

7 Conclusion and further work

In this paper, we address the task of evaluating synthetic CS in supervised and unsupervised approaches. In supervised setting, we solve HinglishEval *quality rating prediction* task with a line of finetuned Hinglish Transformer models and establish a new SOTA. In unsupervised setting, we apply the method of acceptablity measures to evaluate the synthetic CS sentences in HinGE dataset. We find that Hinglish finetuned models consistently outperform their original versions.

Several further work directions open up based on this work. First, it is promising to directly compare the unsupervised and supervised approaches presented in this paper, possibly applying the semi-supervised method of Warstadt et al. (2019) for acceptability measures. Second, it is of interest to continue the analysis presented in Section 6.1 with various CS metrics, thus repeating the study of Srivastava and Singh (2021a) on a larger scale.

8 Acknowledgements

We thank the anonymous reviewers for their valuable comments and remarks. This research was supported in part through computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021).

References

Heike Adel, Katrin Kirchhoff, Dominic Telaar, Ngoc Thang Vu, Tim Schlippe, and Tanja Schultz. 2014. Features for factored language models for code-Switching speech. In *Proc. 4th Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU 2014)*, pages 32–38.

Heike Adel, Ngoc Thang Vu, Katrin Kirchhoff, Dominic Telaar, and Tanja Schultz. 2015. Syntactic and semantic features for code-switching factored language models. *IEEE/ACM transactions on audio, speech, and language Processing*, 23(3):431–440. Publisher: IEEE.

Heike Adel, Ngoc Thang Vu, Franziska Kraus, Tim Schlippe, Haizhou Li, and Tanja Schultz. 2013a. Recurrent Neural Network Language Modeling for Code Switching Conversational Speech. In *The 38th International Conference on Acoustics, Speech, and Signal Processing*.

Heike Adel, Ngoc Thang Vu, and Tanja Schultz. 2013b. Combination of Recurrent Neural Networks and Factored Language Models for Code-Switching Language Modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 206–211, Sofia, Bulgaria. Association for Computational Linguistics.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A Centralized Benchmark for Linguistic Code-switching Evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Mohd Zeeshan Ansari, M. M. Sufyan Beg, Tanvir Ahmad, Mohd Jazib Khan, and Ghazali Wasim. 2021. Language Identification of Hindi-English tweets using code-mixed BERT. ArXiv:2107.01202 [cs].

Scott Baldauf. 2004. A Hindi-English jumble, spoken by 350 million. *Christian Science Monitor*.

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.

Gayatri Bhat, Monojit Choudhury, and Kalika Bali. 2016. Grammatical Constraints on Intra-sentential

Code-Switching: From Theories to Working Models. ArXiv:1612.04538 [cs].

Alexey Birshert and Ekaterina Artemova. 2021. Call Larisa Ivanovna: Code-Switching Fools Multilingual NLU Models. ArXiv:2109.14350 [cs].

Myers-Scotton Carol. 1993. Duelling languages: Grammatical structure in codeswitching.

Joyce Y. C. Chan, Houwei Cao, P. C. Ching, and Tan Lee. 2009. Automatic Recognition of Cantonese-English Code-Mixing Speech. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 14, Number 3, September 2009*.

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: Crowd-sourcing data and techniques. In *Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 29–38. Association for Computational Linguistics (ACL).

Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee. 2019. Code-switching Sentence Generation by Generative Adversarial Networks and its Application to Data Augmentation. *arXiv:1811.02356 [cs]*. ArXiv: 1811.02356.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR*, abs/1911.02116. ArXiv: 1911.02116.

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, Goa, India. NLP Association of India.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805. _eprint: 1810.04805.

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.

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, Portorož, Slovenia. European Language Resources Association (ELRA).

Saurabh Garg, Tanmay Parekh, and Preethi Jyothi. 2018. Code-switched language models using dual rnns and same-source pretraining. *arXiv preprint arXiv:1809.01962*.

Devansh Gautam, Prashant Kodali, Kshitij Gupta, Anmol Goel, Manish Shrivastava, and Ponnurangam Kumaraguru. 2021. CoMeT: Towards Code-Mixed Translation Using Parallel Monolingual Sentences. pages 47–55.

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*.

Sunil Gundapu and Radhika Mamidi. 2020. Word level language identification in english telugu code mixed data. *arXiv preprint arXiv:2010.04482*.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5760–5766.

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.

Ganesh Jawahar, El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, and Laks VS Lakshmanan. 2021. Exploring text-to-text transformers for english to hinglish machine translation with synthetic code-mixing. *arXiv preprint arXiv:2105.08807*.

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. 2020a. A new dataset for natural language inference from code-mixed conversations. *arXiv preprint arXiv:2004.05051*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020b. GLUECoS : An Evaluation Benchmark for Code-Switched NLP. *arXiv:2004.12376 [cs]*. ArXiv: 2004.12376.

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*.

P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. 2021. HPC Resources of the Higher School of Economics. *Journal of Physics: Conference Series*, 1740(1):012050. Publisher: IOP Publishing.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. 2020. How Furiously Can Colorless Green Ideas Sleep? Sentence Acceptability in Context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2015. Unsupervised Prediction of Acceptability Judgements. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1618–1628, Beijing, China. Association for Computational Linguistics.

Grandee Lee and Haizhou Li. 2020. Modeling code-switch languages using bilingual parallel corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 860–870.

Ying Li and Pascale Fung. 2012. Code-Switch Language Model with Inversion Constraints for Mixed Language Speech Recognition. In *Proceedings of COLING 2012*, pages 1671–1680, Mumbai, India. The COLING 2012 Organizing Committee.

Ying Li and Pascale Fung. 2014. Language Modeling with Functional Head Constraint for Code Switching Speech Recognition. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 907–916, Doha, Qatar. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8433–8440. Issue: 05.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP. ArXiv:2005.05909 [cs].

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A Code Mixed Hindi-English Dataset and BERT Language Models. ArXiv:2204.08398 [cs].

Dwija Parikh and Thamar Solorio. 2021. Normalization and back-transliteration for code-switched data. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 119–124.

Parth Patwa, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas PYKL, 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, Barcelona (online). International Committee for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Shana Poplack. 1980. Sometimes I'll start a sentence in Spanish Y TERMINO EN ESPAÑOL: toward a typology of code-switching. 18(7-8):581–618. Publisher: De Gruyter Mouton Section: Linguistics.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018a. 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.

Adithya Pratapa, Monojit Choudhury, and Sunayana Sitaram. 2018b. Word embeddings for code-mixed language processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3067–3072.

Ruba Priyadharshini, Bharathi Raja Chakravarthi, Mani Vegupatti, and John P McCrae. 2020. Named entity recognition for code-mixed Indian corpus using meta embedding. In *2020 6th international conference on advanced computing and communication systems (ICACCS)*, pages 68–72. IEEE.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

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.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A Toolkit for Generating Synthetic Code-mixed Text. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 205–211, Online. Association for Computational Linguistics.

Bidisha Samanta, Sharmila Reddy, Hussain Jagirdar, Niloy Ganguly, and Soumen Chakrabarti. 2019. A Deep Generative Model for Code-Switched Text. *arXiv:1906.08972 [cs]*. ArXiv: 1906.08972.

Sebastin Santy, Anirudh Srinivasan, and Monojit Choudhury. 2021. BERTologiCoMix: How does Code-Mixing interact with Multilingual BERT?

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. Publisher: World Scientific.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mGPT: Few-Shot Learners Go Multilingual.

Jasdeep Singh, Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. Xlda: Cross-lingual data augmentation for natural language inference and question answering. *arXiv preprint arXiv:1905.11471*.

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.

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.

Rajat Singh, Nurendra Choudhary, and Manish Shrivastava. 2018c. Automatic normalization of word variations in code-mixed social media text. *arXiv preprint arXiv:1804.00804*.

Ganji Sreeram and Rohit Sinha. 2017. Language modeling for code-switched data: Challenges and approaches. *arXiv preprint arXiv:1711.03541*.

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. 2021a. Challenges and Limitations with the Metrics Measuring the Complexity of Code-Mixed Text. Technical Report arXiv:2106.10123, arXiv. ArXiv:2106.10123 [cs] type: article.

Vivek Srivastava and Mayank Singh. 2021b. HinGE: A Dataset for Generation and Evaluation of Code-Mixed Hinglish Text. Technical Report arXiv:2107.03760, arXiv. ArXiv:2107.03760 [cs] type: article.

Vivek Srivastava and Mayank Singh. 2021c. Quality Evaluation of the Low-Resource Synthetically Generated Code-Mixed Hinglish Text. Technical Report arXiv:2108.01861, arXiv. ArXiv:2108.01861 [cs] type: article.

Samson Tan and Shafiq Joty. 2021. Code-mixing on sesame street: Dawn of the adversarial polyglots. *arXiv preprint arXiv:2103.09593*.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text. Technical Report arXiv:2107.06483, arXiv. ArXiv:2107.06483 [cs] type: article.

S Thara, E Sampath, Phanindra Reddy, and others. 2020. Code mixed question answering challenge using deep learning methods. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 1331–1337. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR*, abs/1706.03762. ArXiv: 1706.03762.

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A first speech recognition system for Mandarin-English code-switch conversational speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4889–4892. ISSN: 2379-190X.

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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *arXiv:1805.12471 [cs]*. ArXiv: 1805.12471.

Genta Indra Winata, Zhaojiang Lin, and Pascale Fung. 2019a. Learning multilingual meta-embeddings for code-switching named entity recognition. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 181–186.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019b. Code-Switched Language Models Using Neural Based Synthetic Data from Parallel Sentences. *arXiv:1909.08582 [cs]*. ArXiv: 1909.08582.