The University of Edinburgh's Submission to the WMT22 Code-Mixing Shared Task (MixMT)

Faheem Kirefu **Vivek Iyer** **Pinzhen Chen** **Laurie Burchell**
School of Informatics, University of Edinburgh
{fkirefu,vivek.iyer,pinzhen.chen,laurie.burchell}@ed.ac.uk

Abstract

The University of Edinburgh participated in the WMT22 shared task on code-mixed translation. This consists of two subtasks: i) generating code-mixed Hindi/English (Hinglish) text generation from parallel Hindi and English sentences and ii) machine translation from Hinglish to English. As both subtasks are considered low-resource, we focused our efforts on careful data generation and curation, especially the use of backtranslation from monolingual resources. For subtask 1 we explored the effects of constrained decoding on English and transliterated subwords in order to produce Hinglish. For subtask 2, we investigated different pretraining techniques, namely comparing simple initialisation from existing machine translation models and aligned augmentation. For both subtasks, we found that our baseline systems worked best. Our systems for both subtasks were one of the overall top-performing submissions.

1 Introduction

Code-mixing is the shift from one language to another within a single conversation or utterance (Sitaram et al., 2019). It is an extremely common and diverse communicative phenomenon worldwide (Doğruöz et al., 2021; Sitaram et al., 2019), though one which is currently under-served by many NLP technologies (Solorio et al., 2021).

One of the most well-known examples of code-mixing is between Hindi and English, commonly referred to as Hinglish[1]. It is extremely common amongst Hindi-English bilingual speakers in both speech and text, used across a range of genres and media (Parshad et al., 2016), and has its own distinctive features and linguistic forms (Kumar, 1986; Sailaja, 2011). The process of generating Hinglish from the written text is non-trivial, as code-mixing may happen at the phrase or word level, but Hindi and English differ substantially syntactically.

As a novel addition to the current code-mixing NLP research, we investigated lexically constraining the Hinglish output in subtask 1 to only contain words from English and Hindi sources. Through analysis, we demonstrated that transliteration mismatches could affect performance.

Another novel approach we explore for this task, particularly for subtask 2, is a denoising-based pretraining technique called Aligned Augmentation (AA) (Pan et al., 2021). AA, which trains MT models to denoise artificially generated code-mixed text, was shown by Pan et al. (2021) to boost translation performance across a variety of languages - thanks to the enhanced transfer learning brought about by code-mixed pretraining. In this work, we explored if this general-purpose approach could be useful for translating authentic, human-generated code-mixed text, focusing on Hinglish.

Despite these efforts, we found that for both subtasks our original baselines worked better and constituted our final submissions for this task, which ranked as one of the top-performing systems for both subtasks, by both automatic and human evaluation. We hope our methods, particularly Hinglish data generation, that allowed us to build these systems would be useful to the community; as would the findings from our additional research explorations.

2 Related Work

2.1 Code-mixing

Due to an increasing prevalence of code-mixed data on the Internet, there is a growing body of research into code-mixing, particularly for Hinglish, in the NLP community. Doğruöz et al. (2021) provide a comprehensive literature review of code-mixing in the context of language technologies. Whilst they highlight several challenges inherent

[1] In the scope of this paper, we designate "hg" as the language code for Hinglish.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1145 - 1157

in NLP with code-mixed text (such as understanding cultural and linguistic context, evaluation, and a lack of user-facing applications), the most notable obstacle for this shared task is the lack of data. They note that there are very few code-mixed datasets, making it challenging to build deep learning models such as those for NMT. In this work, we use backtranslation as our main data augmentation method (Edunov et al., 2020; Barrault et al., 2020; Akhbardeh et al., 2021, *inter alia*). This allows us to leverage the larger amount of monolingual data for better final model performance. The XLM toolkit (Lample and Conneau, 2019) seemed an ideal choice to backtranslate our Hinglish. This is because it has shown promising results in unsupervised and semi-supervised settings where parallel data is sparse, but monolingual data is ample. Also given that Hinglish is closely related to both languages, we believed Hinglish should be an ideal language to use in a semi-supervised setting.

2.2 Constrained decoding

Constrained decoding involves applying restrictions to the generation of output tokens during inference. Most implementations have the goal of ensuring that desired vocabulary items appear in the target side sequence (Hokamp and Liu, 2017; Hasler et al., 2018; Post and Vilar, 2018). Alternatively, Kajiwara (2019) paraphrase an input sentence by forcing the output to not include source words, and Chen et al. (2020) constrain NMT decoding to follow a corpus built in a trie data structure to find parallel sentences.

To the best of our knowledge, previous linguistics research investigated and applied the grammatical constraints in code-mixing (Sciullo et al., 1986; Belazi et al., 1994; Li and Fung, 2013), rather than the novel method in our work of introducing lexical constraints.

2.3 Aligned augmentation

Several recent works (Yang et al., 2020a,b; Lin et al., 2020; Pan et al., 2021) have explored enhancing cross-lingual transfer learning by pretraining models on the task of 'denoising' artificially code-mixed text. Methods to create the necessary code-mixed data vary, and include bilingual or multilingual datasets and word aligners (Yang et al., 2020a, 2021), lexicons (Yang et al., 2020b; Lin et al., 2020; Pan et al., 2021), or combining code-mixed noising with traditional masked noising approaches (Li et al., 2022).

The most successful among these methods is Aligned Augmentation (AA) (Pan et al., 2021), which randomly substituting words in the source sentence with their word-level translations, as obtained from a MUSE (Lample et al., 2018) dictionary. Pan et al. (2021) showed that their technique can effectively align multilingual semantic word representations and boost performance across various languages. However, these methods focus on training general-purpose MT models. In this work, we investigate their utility for translating real human-generated code-mixed text.

2.4 Automatic evaluation metrics

Automatic translation evaluation is usually done using BLEU (Papineni et al., 2002), yet there is no comprehensive study on its suitability for code-switched translation. Specifically in this task, the organisers announced that the participating systems will be evaluated using ROUGE-L (Lin, 2004) and word error rate (WER). Nonetheless, the packages implementing these metrics were not specified. Since ROUGE comes with different language, stemming and tokenisation settings, we instead used BLEU, ChrF++ (Popović, 2017), translation error rate (TER), and WER[2] for our internal validation. The first three are as implemented with sacreBLEU (Post, 2018). We stick to the default configurations, except that the ChrF word n-gram order is explicitly set to 2 to make it ChrF++. In addition, the organisers performed a small-scale human evaluation on 20 test instances for all submissions.

In this work, we advocate for a character-based metric when evaluating the Hinglish output in subtask 1. This is because for the code-switched language, there is no formal spelling or defined grammar, and words may have a diverse range of acceptable transliterations and lexical forms.

3 Subtask 1: Translating into Hinglish

Good quality Hinglish data is hard to come by, and parallel Hinglish data with Hindi or English even more frugal. Therefore, for both subtasks we concentrated our efforts on generating good Hinglish backtranslation. We planned to use the model which produced the highest quality Hinglish for subtask 1 as our backtranslator for subtask 2, hence we focused our efforts on each subtask sequentially.

[2] https://github.com/jitsi/jiwer

3.1 Data cleaning and preprocessing

After deduplicating the data, we removed non-printing characters and normalised the punctuation. We then ran rule-based filters, removing any sentences with fewer than two or more than 150 words, where fewer than 40% of the words are written in the relevant script, or where over 50% of characters are not letters in the relevant script. For English and Hindi, we ran `fasttext` language ID and removed any sentence which was not classified as the relevant language.[3] For Hinglish, we also removed any sentence with a predicted probability of English greater than 0.99 in order to remove sentences that were solely in English. We tokenised English and Hinglish using Moses scripts (Koehn et al., 2007) and tokenised Hindi using the `indicnlp` library (Kunchukuttan, 2020).

We decided to add explicit preprocessing and postprocessing capabilities for handling social media text, given that this was the domain for subtask 2. On both source and target sides, we replaced URLs, Twitter handles, hashtags and emoticons each with their own placeholder tokens[4], to be replaced back from the source after inference. These placeholders made up 1.7% of the validation set tokens for subtask 2, far higher than would appear in general domain data.

3.1.1 The HinGe dataset

The primary dataset for subtask 1 was the HinGe dataset (Srivastava and Singh, 2021), which consisted of hi-en-hg parallel sentences, with some examples synthetic and some human-generated. This was provided to us pre-split into training and development sets for both data types. However, we noticed that these sets were not mutually exclusive, and after deduplication and filtering on the synthetic data human annotations[5], we had 6,727 hi-en-hg examples in total.

3.1.2 Base hi↔en translation models

Firstly, we trained four Transformer-base (Vaswani et al., 2017) models with different seeds using Marian (Junczys-Dowmunt et al., 2018) for both hi→en and en→hi directions, using the data from the hi-en parallel Samanantar corpus[6] (Ramesh et al., 2021). Given the findings of Ding et al. (2019) with regard to vocabulary choice for low-resource scenarios, and that our task inherently contains transliteration, we opted for a low BPE (Sennrich et al., 2016) merge size of 4k, resulting in a small joint vocabulary of 7.9k. We used the hi-en FLORES development set (Goyal et al., 2022) for validation and early stopping, and noticed our model produced surprisingly good quality translations in both directions[7]. We used these models (along with vocabulary) to both initialise subsequent models and generate backtranslation for more training data.

3.1.3 Hinglish data

L3Cube-HingCorpus (Nayak and Joshi, 2022) and CC-100 Hindi Romanized (Conneau et al., 2020a) are two Hinglish corpora that we wished to backtranslate into both English and Hindi. Given that we only had a small amount of parallel Hinglish data, compared to our 'monolingual' datasets, we used the XLM toolkit (Lample and Conneau, 2019) to train a semi-supervised model (see Appendix A for details). We then backtranslated the monolingual Hinglish data into both English and Hindi. However, given the noisy quality of the data and translations themselves, we decided to evaluate them using our hi→en and en→hi Marian models. Specifically, for an en-hi backtranslated (XLM) sentence pair, we translated the en/hi into hi/en respectively, then evaluated the double translated output using ChrF, with the XLM backtranslations as the references. We then took a mean of the English and Hindi ChrF score to get our final confidence value. We used the resulting hg-en-hi sentence trios with values at least 0.4, to compromise between the quality and quantity of data available to use as training. Most of the sentences scored quite poorly, and filtering on 0.4 yielded 2.1M sentences, only about 12% of the original Hinglish monolingual dataset.

3.1.4 Transliteration

In order to best leverage the Samanantar hi-en parallel corpus, we transliterated the Hindi side into Roman script[8], on the word level. Although this

[3] Our cleaning scripts are adapted from those provided by the Bergamot project. https://github.com/browsermt/students/tree/master/train-student Specifically, we add support for Hindi and Hinglish text.

[4] `<URL>`, `<TH>`, `<HT>` and `<EMO>` respectively

[5] We only kept sentences with an average rating greater than 4, and annotator disagreement less than 5

[6] Each sentence was annotated with the LaBSE (Feng et al., 2022) Alignment Score (between 0 and 1), so we filtered out values less than 0.65, resulting in around 10.1M sentences

[7] sacreBLEU: 33.8 for hi→en and 32.7 for hi→en on FLORES development set

[8] In the scope of this paper, we use "ht" to denote pure romanised Hindi transliteration

Beam Size	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Unconstrained				
1	17.8	42.8	65.3	**81.5**
4	**18.1**	**44.0**	**64.5**	85.7
12	18.0	43.8	64.8	86.0
24	18.0	43.7	65.0	85.5
36	17.9	43.5	65.1	85.4
48	18.0	43.6	65.2	85.5
Constrained				
1	10.8	33.1	76.1	75.1
2	12.2	35.6	74.9	69.1
4	13.2	36.6	74.2	63.5
6	14.1	37.7	73.5	60.8
12	14.6	38.1	73.7	58.6
24	14.8	38.5	73.5	57.2
36	14.9	38.7	73.6	**56.7**
48	15.0	38.7	73.6	57.0

Table 1: Experimental results on the validation set with unconstrained and constrained decoding for subtask 1.

forward transliteration was not likely to contain much code-mixed text, it would still be useful training data for our model, given that both the Hindi and English sources are assumed to be either the original sources or human translationese.

We used the AI4Bharat Indic transliterator (Madhani et al., 2022), to convert (on the word level) all romanised tokens contained in our monolingual Hinglish datasets into Devanagari script. This tool is a neural-based model with beam search capabilities, therefore we generated the top 4 results in Hindi for each Hinglish token. We used the top 4 instead of the most likely candidate as, upon inspection, we found that the correct corresponding Hindi token was not always predicted first. We also used a human-generated list of Hinglish-English pairs form the Xlit-Crowd corpus (Khapra et al., 2014) which we treated as the gold standard.

To summarise, our training data for our hi→ht transliterator[9] consists of 5.3M Hinglish-Hindi word pairs (1.3M unique Hinglish words), and 15k from XlitCrowd, of which we use 1k as a validation set for early stopping. We train a small transformer model with Marian on the **character-level** for both input and output. When forward transliterating the Hindi side of the Samanantar corpus, we copied over non-standard strings (such as numbers, punctuation etc.), or else we looked up the token (if it existed) in our gold standard list. Otherwise, we used our transliteration model as a final back-off. In hindsight, one disadvantage of our approach was that we did not generate multiple candidates for each Hindi word, to reflect the diversity of possible romanised candidate tokens.

We also used this transliteration model as part of our constrained decoding experiments later (see Section 3.3).

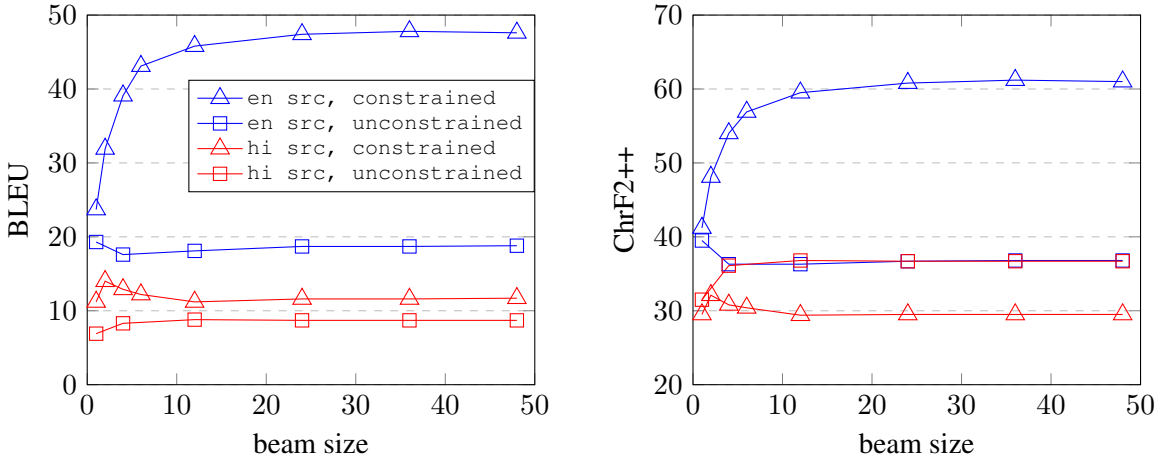
3.2 Baseline (unconstrained decoding)

We decided to use a dual encoder setting given that we have two inputs in this task, and initialise our model from our previously trained Marian MT systems. We used hi→en to initialise the Hindi-decoder and the English-encoder cross attention parameters, whereas en→hi was used to initialise the English-encoder and all other decoder parameters. Our vocabulary was the same as the pretrained models.

Early stopping with patience 10 on the HinGe dataset was used for convergence - for all of the experiments mentioned in this paper. Our training regime consisted of two stages:

- General domain - The training datasets used were the backtranslated Hinglish and forward transliterated Samanantar corpora. We used all of the HinGe dataset as a validation set.
- Finetuning - We continue training on a sub-

[9]We decided to build our own transliterator as we found existing tools in this direction to be of poor quality



(a) BLEU (b) ChrF++

Figure 1: Validation BLEU and ChrF++ of the constrained and unconstrained outputs scored against English and transliterated Hindi *sources* separately.

set of HinGe dataset, using a distinct smaller subset (1k) of it as a validation set.

3.3 Constrained decoding

After analysing the training data, we hypothesized that nearly all the output words should either be from the English source, or as a transliteration of a word from the Hindi source, with likely little change in sentence structure. This inspired us to use the technique of constrained decoding when generating Hinglish.

Unlike standard constrained decoding where a model is forced to incorporate certain words in the output, our proposal is to exclude vocabulary words that do not exist in English or transliterated Hindi source sentences. Following Chen et al. (2020)'s notion, we applied pre-expansion pruning: disallowed word paths are assigned an extremely small score before hypotheses are ranked and expanded. Specifically, to obtain Hindi transliteration, we used our transliteration model described in Section 3.1.4.

We performed beam searches with constrained decoding and reported automatic scores on the validation set in Table 1. Unfortunately, constrained decoding does not beat unconstrained decoding. As a general trend, WER and TER do not change much as beam size increases, while BLEU and ChrF++ significantly improve.

To better understand the impact of constrained decoding, we score the validation outputs against English and transliterated Hindi sources separately, then plot BLEU and ChrF++ numbers in Figure 1a and Figure 1b. We observe that with increasing beam sizes, constrained decoding prefers to generate English tokens instead of transliterated Hindi. Unconstrained decoding achieves a much better balance.

One hypothesis is that the quality of Hindi transliteration is not perfect, resulting in the model preferring English tokens from the vocabulary. Hence, we compute the percentage of words in the gold reference as well as in the unconstrained (baseline) output that come from neither the English nor the transliterated Hindi source. Surprisingly, on average 45.1% of the total words in the unconstrained output do not appear in the sources; as for the gold reference, it is 39.8% which is slightly lower. It is worth noting that the numbers might be inflated as we computed the word overlap after outputs are detokenised. Yet it implies that many of the reference words do not exactly appear in the lexical constraints determined from the source senteneces.

Finally, we visualise the first five validation sentences in Table 2. We highlight in *red* the target words that do not exist in the source sentences; we also label the possible corresponding tokens from the sources in *blue*. It can be confirmed that most mismatches are due to differences in Hindi transliteration and letter cases. This indicates that the lexical constraint idea is suitable in theory, but it is hindered by the error propagation in transliteration. This may have been alleviated by running multiple transliteration schemes on the Hindi source to make the constraints more diversified.

hi source	1995 से 2004 के दौरान औसत धरातलीय तापमान 1940 से 1980 तक के औसत तापमान से भिन्न है
hi transliteration	1995 *sey* 2004 ke *Dauran* ausat *Dharatliya Tapman* 1940 *sey* 1980 tak ke ausat *Tapman sey bhinnn is*
en source	The average *geological* temperature of the earth from 1995-2004 is different than that of 1940-1980 .
constrained	The average Dharatliya tapman of the earth from 1995 -tak ke ausat tapman from bhinn.
unconstrained	1995 *se pratik dauran* average *dharatliy temperwof* the earth from *1990 se* 1980 tak ke ausat *temperwale se bhinn hai*.
reference	from 1995-2004 ke *dauran* average *geology* temperature of earth 1940 *se* 1980 tak ke ausat temperature *se* different *hai*.
hi source	धृतराष्ट्र एवं गांधारी के १०० पुत्रों में सबसे बड़े ।
hi transliteration	Dhritrashtra Evan Gandhari ke 100 putron *main* sabse bade .
en source	Dhrudharashtra and Ghandhari 's eldest among their 200 sons .
constrained	Dhrudharashtra among their 200 sons.
unconstrained	Dhrudharashtra among their 200 sons.
reference	Dhrudharashtra and Ghandhari ke 100 sons *mein* sabse bade.
hi source	इस प्रकार राजस्थान के रेगिस्तान का एक बड़ा भाग शस्य श्यामला भूमि में बदल जायेगा ।
hi transliteration	is *Prakar* rajasthan ke registan ka a badaa bhaag Shasya Shyamala bhumi *main* cange jayega .
en source	In this way a major part of the desert in Rajasthan would become a harvesting and fertile land .
constrained	In this way a major part of the desert in Rajasthan would become a harvesting and jayega.
unconstrained	In this way a major part of the desert in Rajasthan would become a harvesting and *wtile* land.
reference	is *prakar* rajasthan ke desert ka *ek* major part harvesting and fertile land *mein badal* jayega.
hi source	राष्ट्रपति की अध्यादेश जारी करने की शक्ति पे नियंत्रण
hi transliteration	Rashtrapati ki *Adhyadesh jaari* karne ki shakti pay *Niyantran*
en source	The power of the President to proclaim *Ordinance* is subject to :
constrained	Rashtrapati ki Adhyadesh jaari karne ki
unconstrained	Rashtrapati ki *adhyadesh* jaari karne ki *pratiniyantran*.
reference	President ki *ordinance jari* karne ki power *pr niyantran*.
hi source	1000 से अधिक हाथी निर्माण के दौरान यातायात हेतु प्रयोग हुए थे ।
hi transliteration	1000 *sey Adhik* haathi *Nirman* ke *Dauran* yatayat hetu *pryog huye* they .
en source	*More* than 1000 elephants were used during the time of construction for transportation .
constrained	Dauran transportation ke time yatayat hetu pryog hue the.
unconstrained	1000 *se adhik* haathi *nirman* ke *dauran* transportation hetu pryog *hue* the.
reference	*more* than 1000 elephants construction ke *dauran* transportation hetu *prayog hue* the.

Table 2: The first five validation instances: English and Hindi sources, as well as constrained, unconstrained and reference outputs. *red* denotes the target side words that do not appear in either of the source sentences from a constrained aspect; *blue* denotes possible source-target matches in a different surface form.

4 Subtask 2: Hinglish-to-English

4.1 Data cleaning and preprocessing

The primary dataset provided for this task PHINC (Srivastava and Singh, 2020) is relatively small at 13.7k English-Hinglish pairs. Therefore, we aimed to generate domain-specific parallel data using our baseline model from Subtask 1 on English monolingual data.

We analysed the source side of the validation dataset to determine the most frequent content words (see Table 3) and then selected these words (and any morphological/spelling variants) from the English WikiMatrix corpus (Schwenk et al., 2021). This yielded a total 477k English sentences and we henceforth refer to this selection of sentences as ToxicWiki. We also used Sentiment140 (Sahni et al., 2017), a dataset of 1.6M tweets in English, as the domain of our validation set is also Twitter.

Word	Validation	WikiMatrix
rape	249	23,198
hate	117	16,824
terrorism	24	11,160
khoon (blood)	21	59,526
murder	21	75,066
india	16	291,054
Total	-	476,828

Table 3: Frequency of top content words present in our validation set, and the number of sentences within WikiMatrix that contained the word (or morphological variants). The resulting sentences formed ToxicWiki

To obtain the Hinglish side of both Sentiment140 and ToxicWiki datasets, we backtranslated into Hindi using our en→hi Marian model, and then

used the en-hi pair and our baseline system for subtask 1 to obtain the corresponding Hinglish. However, many of the placeholders (such as <HT>) did not occur frequently enough during the training of subtask 1 for the model to learn to consistently copy them across; therefore the model was not able to predict them with a large degree of accuracy. Therefore, we ran a postprocessing script that corrected for placeholders on the backtranslated Hinglish, given the English source, so that our downstream model would be able to learn to simply copy these placeholders across. Specifically, we made sure that the number of each placeholder type in the backtranslated Hinglish was the same (and in roughly the same position) as that in the source sentence.

For the AA experiments described in Section 4.3, we used monolingual Hindi, English and Hinglish data. For Hindi and English, we randomly sampled 20M sentences from the News Crawl corpora (Akhbardeh et al., 2021). For Hinglish, the monolingual corpora described above was used. In order to code-mix these corpora as described in the AA algorithm, we used MUSE dictionaries for the Hindi-English pair. For Hinglish-Hindi pairs, we used the data generated with AA for the transliteration model.

4.2 Baseline systems

We used a hi→en MT to initialise the baseline hg→en model.

Our training regime consisted of three stages:

1. General - Training on the backtranslated en-hg internet corpora (with confidence value at least 0.4), and ht-en side of the Samanantar corpus, where we treat the transliteration as Hinglish. We used the PHINC dataset as our validation set for early stopping.

2. We continued training on Sentiment140 and ToxicWiki corpus, using the same validation set as before, until convergence.

3. We continued training on the PHINC dataset, using a small subset (1k) of it as validation data for early stopping.

As we had multiple hi→en MT systems, we also trained an ensemble model of four, where we followed the same training regime above with parameters initialised from each of our hi→en models. Our results are shown in Table 4, with our ensemble model outperforming the single on all metrics.

4.3 Aligned Augmentation for subtask 2

Our Aligned Augmentation (AA) experiments where implemented with Fairseq (Ott et al., 2019), and we used the Transformer architecture, with 12 encoder and 12 decoder layers. Our first step consisted of pretraining these models on Hindi, English, and Hinglish corpora, with the target being the "denoised" sentence - thus training the model to reconstruct the original sentence, following the AA algorithm. For validation, we randomly sampled 1k sentences from the training corpus.

We then finetuned this model on the Hinglish-English parallel corpora mentioned above. The major AA baselines we trained and their performances are listed in Table 5 - along with a randomly initialised baseline that was trained solely on the parallel corpora. The data sources we used in our experiments were quite diverse: we started with high-quality monolingual data for pretraining followed by parallel datasets of varying domains and qualities, (the Hinglish backtranslated corpora, Sentiment140, PHINC and ToxicWiki). We attempted to explore how best these resources could be utilised. We started with our default training paradigm: we finetuned on backtranslated Hinglish, followed by the ToxicWiki and then a shuffled concatenation of the social media datasets - the Sentiment140 and PHINC datasets respectively. This was based on the intuition that the final model should be most recently trained on datasets from similar domains as the test set.

Following this paradigm, we conducted two sets of experiments: a "validation experiment" that tries to estimate the best choice of validation sets, and "training experiments" to verify the importance of some training sources empirically. The former is a crucial decision in our experiments given our use of early stopping. We find that validating on the official MixMT validation sets released for Subtask 2 ends up performing significantly worse than validating on a subset of the respective training datasets. This is surprising given the performances reported in Table 5 are evaluated on the same validation sets. This suggests that training and validating the model on corpora from different domains can help boost the final performance - even if it does not improve loss on the final validation set. In the latter body of experiments, we attempted to determine the value of the XLM backtranslated corpora on performance - which seems very noisy on manual inspection, with the target side (English) being

	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Baseline Experiments				
Single model	24.5	47.0	65.1	72.0
Ensemble (of 4)	**25.5**	**48.7**	**62.9**	**70.5**

Table 4: Baseline results for subtask 2 on the MixMT validation set.

generated through backtranslation. Surprisingly, its inclusion significantly enhances performance, by +5 BLEU points. This could be due to various reasons: its sheer size (15M sentences), the presence of word-level translations between English to Hinglish in parallel sentences (despite grammatical errors), the similarity between the source and the target encouraging "copying" which can sometimes be beneficial for this task, etc. We also find that the inclusion of hi-en along with hg-en further boosts performance, consistent with the findings of previous works on multilingual MT. We empirically found that including 'all' available hi-en sentences and 'all' available hg-en sentences was more beneficial than splitting our parallel dataset into the two respective directions – despite the target sentence being duplicated in the former.

Compared to the Random baselines, our final AA baselines show consistent improvement for all given metrics - though the improvement is not very significant with respect to BLEU o TER. A closer glance at the validation set and the generated predictions reveals the potential reason behind this - there is a significant amount of noise present in the validation sets due to the social media domain, with errors in both syntax and semantics. Given that it is not always easy to comprehend and translate such sentences well, the gold reference sentences are sometimes of relatively poor quality - containing various potential errors such as inaccurate word form predictions, grammatical errors, misspellings etc. While word-based metrics may fail to handle these cases; ChrF++, being a character-based metric, can likely alleviate noise that may have propagated to reference sentences and might be a more suitable metric for Subtask 2 as well. It is encouraging to note AA's improvement over the Random baseline in this light.

AA appears to bring about some improvement qualitatively, especially regarding noisy input - for instance, it was able to more accurately translate misspellings and handle grammatical inconsistencies. However, the frequency of sentences where AA performs better than its randomly initialized counterparts seems relatively low. One explanation could be that fine-tuning the model on 18M parallel sentences could lead it to 'forget' the representations learned during pretraining. This is in line with the findings of (Pan et al., 2021) that observe relatively lower improvements for high-resource languages. While adding large corpora (15M sentences) such as the XLM backtranslated corpora does lead to net improvements, it is possible optimization in the size of finetuning data used could lead to even greater gains. Secondly, given that our dictionaries appear to help in noise resolution, it might be useful to incorporate various types of misspellings rigorously in the code-mixing lexicons created - thus enabling the final model to be more robust. Finally, including training corpora from other Indo-Aryan languages like Urdu or Marathi could be beneficial. Although Subtask 2 focuses on the translation of Hinglish-English, the validation and test sets (as well as training sets) contain many examples of code-mixing between related Indo-Aryan languages and English - most prominently in Urdu, which is historically and linguistically similar to Hindi.

In the end, we observe that the AA models we train are unable to beat our original single-model baseline, despite having more parameters. Curiously, this is also the case for the randomly initialized baseline in Table 5. Due to time constraints, we are unable to investigate the reasons behind these. Possible explanations could include: training paradigm differences (initializing with hi→en vs mixing hi→en with hg→en), ensembling, experimental setting disparities, inherent differences between training libraries (Fairseq vs Marian). It is possible that addressing these disparities, as well as exploring the directions suggested in the previous paragraph, could enable AA baselines to yield superior results for code-mixed translation.

	BLEU (↑)	ChrF++ (↑)	TER (↓)	WER (↓)
Validation Experiments				
AA (dev = MixMT valid)	20.5	41.2	72.7	78.6
AA (dev = train subset)	23.3	45.7	68.3	74.6
Training Experiments (dev=train subset)				
AA (train = all Hg->En minus XLM BT data)	18.3	38.4	78.3	83.4
AA (train = all Hg->En)	23.3	45.7	68.3	**74.6**
AA (train = all Hg->En + all Hi->En)	**24.4**	**46.2**	**68.2**	74.9
Random	24.3	45.2	68.4	74.6

Table 5: Aligned Augmentation experiments for subtask 2, as evaluated on the official MixMT Subtask 2 validation set. “Validation experiments" refers to experiments performed to select the best choice of the validation set for early stopping. ‘MixMT valid’ refers to the same validation set mentioned earlier (that is also used for evaluation), while ‘train subset’ refers to a subset (last 1000 sentences) of the respective training corpus. “Training experiments” seek to explore various dataset choices during training time, using a subset from the training corpus for validation.

	BLEU	ChrF++	TER	WER	ROUGE-L	Human Eval. Score
Subtask 1	26.9	52.7	55.2	56.2	57.9	3.85
Subtask 2	28.7	51.2	59.1	61.3	62.5	3.75

Table 6: Final Test Results for the University of Edinburgh’s submissions of MixMT 2022. BLEU, ChrF++ and TER were evaluated by us while WER and ROUGE-L results are from the official Codalab leaderboard. Human evaluation (on a scale of 1-5) was provided by the organisers on 20 random sentences and we report the average.

5 Test Results

The final test results for our submissions are listed in Table 6. For Subtask 1, we used unconstrained decoding with beam-size 12, and for Subtask 2 we used our baseline ensemble (4) with beam-size 36. We evaluated BLEU, ChrF++ and TER ourselves, while the other metrics are provided by the organizers. We ranked second in both subtasks on the MixMT leaderboard[10] although in both the automatic and human evaluation[11], there does not appear to be a statistically significant difference. Furthermore, we note that some participants have an exceedingly high number of test submissions and would encourage future shared tasks to put in place measures to avoid this.

6 Conclusion

In this work, we described our various findings and experiences while building NMT systems that translated between Hinglish and monolingual English/Hindi - as part of the WMT22 Code-Mixing Shared Task. We proposed various corpora that could be useful for these tasks - many of which we create as part of this work - and utilizing these, build high-performing MT systems that, for both subtasks, constituted one of the leading unconstrained models. In addition, we also explored and analysed some alternative approaches for training our models like constrained decoding and Aligned Augmentation (AA) which, despite not beating our original baselines, yielded findings that are useful for future research. Perhaps the most notable of these suggests that efforts to create Hinglish datasets, including using transliterated Hindi as an approximation, can be fruitful and pivotal to high performance. While efforts to handle noise in social media text (such as AA-based pretraining) can also help, further research is required to establish the most optimal ways to do the same.

7 Acknowledgements

We would like to give special thanks to Nikita Moghe for her valuable feedback and insights throughout this research.

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics. The experiments in this pa-

[10] https://tinyurl.com/codalab-ldbd
[11] https://tinyurl.com/heval-mixmt

per were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service[12], and using the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick. Sulis is funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

Hedi M. Belazi, Edward J. Rubin, and Almeida Jacqueline Toribio. 1994. Code switching and x-bar theory: The functional head constraint. *Linguistic Inquiry*, 25(2):221–237.

Ondřej Bojar, Vojtěch Diatka, Pavel Rychlý, Pavel Straňák, Vít Suchomel, Aleš Tamchyna, and Daniel Zeman. 2014. HindEnCorp - Hindi-English and Hindi-only Corpus for Machine Translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Pinzhen Chen, Nikolay Bogoychev, Kenneth Heafield, and Faheem Kirefu. 2020. Parallel sentence mining by constrained decoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1672–1678, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

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

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

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

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Trans. Assoc. Comput. Linguistics*, 10:522–538.

[12] www.csd3.cam.ac.uk

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Mitesh M. Khapra, Ananthakrishnan Ramanathan, Anoop Kunchukuttan, Karthik Visweswariah, and Pushpak Bhattacharyya. 2014. When transliteration met crowdsourcing : An empirical study of transliteration via crowdsourcing using efficient, non-redundant and fair quality control. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 196–202. European Language Resources Association (ELRA).

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Ashok Kumar. 1986. Certain aspects of the form and functions of Hindi-English Code-Switching. *Anthropological Linguistics*, 28(2):195–205.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022. Universal conditional masked language pre-training for neural machine translation.

Ying Li and Pascale Fung. 2013. Improved mixed language speech recognition using asymmetric acoustic model and language model with code-switch inversion constraints. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7368–7372.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Yash Madhani, Sushane Parthan, Priyanka A. Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *ArXiv*, abs/2205.03018.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed hindi-english dataset and bert language models. *arXiv preprint arXiv:2204.08398*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rana D Parshad, Suman Bhowmick, Vineeta Chand, Nitu Kumari, and Neha Sinha. 2016. What is india speaking? exploring the "hinglish" invasion. *Physica A: Statistical Mechanics and its Applications*, 449:375–389.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Tapan Sahni, Chinmay Chandak, Naveen Reddy Chedeti, and Manish Singh. 2017. Efficient twitter sentiment classification using subjective distant supervision. In *9th International Conference on Communication Systems and Networks, COMSNETS 2017, Bengaluru, India, January 4-8, 2017*, pages 548–553. IEEE.

Pingali Sailaja. 2011. Hinglish: code-switching in indian english. *ELT J*, 65(4):473–480.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Anne-Marie Di Sciullo, Pieter Muysken, and Rajendra Singh. 1986. Government and code-mixing. *Journal of Linguistics*, 22(1):1–24.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Thamar Solorio, Shuguang Chen, Alan W. Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan, editors. 2021. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Online.

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, Online. Association for Computational Linguistics.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jian Yang, Shuming Ma, Dongdong Zhang, ShuangZhi Wu, Zhoujun Li, and Ming Zhou. 2020a. Alternating language modeling for cross-lingual pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9386–9393.

Jian Yang, Yuwei Yin, Shuming Ma, Haoyang Huang, Dongdong Zhang, Zhoujun Li, and Furu Wei. 2021. Multilingual agreement for multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 233–239, Online. Association for Computational Linguistics.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.

A XLM details

In order to backtranslate the Hinglish data, we hoped to train a good quality semi-supervised system using the XLM toolkit (Conneau et al., 2020b). We use Masked Language Modelling

(MLM) to pretrain a transformer encoder model on English, Hindi and Hinglish monolingual data. The model consisted of 6 layers, 1024 embedding dimensions, batch size 128, and a 0.1 dropout rate. We use 16.5M sentences of English WikiMatrix (Schwenk et al., 2021), 20M of HindiMono (Bojar et al., 2014) and 18.8M of Hinglish from L3Cube-HingCorpus (Nayak and Joshi, 2022) and CC-100 Hindi Romanized (Conneau et al., 2020a). Vocabulary and data preprocessing is the same as for the Marian models (4k BPE merges).

We then initialised a full transformer model with the pretrained encoder, and further trained with denoised autoencoding, MLM, machine translation[13], and backtranslation[14]objectives. We use the Samanantar corpus (10.1M) for the hi↔en translation objective, the 6.7k HinGe sentences as validation for hg↔en and hg↔hi directions, and the hi-en FLORES development set for hi↔en.

[13]hi↔en directions only

[14]Only direction involving hg: hi-hg-hi, en-hg-en, hg-hi-hg, hg-en-hg