Genre Transfer in NMT: Creating Synthetic Spoken Parallel Sentences using Written Parallel Data

Nalin Kumar and **Ondřej Bojar**
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
knalin55@gmail.com, bojar@ufal.mff.cuni.cz

Abstract

Text style transfer (TST) aims to control attributes in a given text without changing the content. The matter gets complicated when the boundary separating two styles gets blurred. We can notice similar difficulties in the case of parallel datasets in spoken and written genres. Genuine spoken features like filler words and repetitions in the existing spoken genre parallel datasets are often cleaned during transcription and translation, making the texts closer to written datasets. This poses several problems for spoken genre-specific tasks like simultaneous speech translation. This paper seeks to address the challenge of improving spoken language translations. We start by creating a genre classifier for individual sentences and then try two approaches for data augmentation using written examples: (1) a novel method that involves assembling and disassembling spoken and written neural machine translation (NMT) models, and (2) a rule-based method to inject spoken features. Though the observed results for (1) are not promising, we get some interesting insights into the solution. The model proposed in (1) fine-tuned on the synthesized data from (2) produces naturally looking spoken translations for written→spoken genre transfer in En-Hi translation systems. We use this system to produce a second-stage En-Hi synthetic corpus, which however lacks appropriate alignments of explicit spoken features across the languages. For the final evaluation, we fine-tune Hi-En spoken translation systems on the synthesized parallel corpora. We observe that the parallel corpus synthesized using our rule-based method produces the best results.

1 Introduction

Style transfer has been one of the well-studied tasks in the field of Artificial Intelligence (AI). Most of the earlier works using deep learning were in the domain of Computer Vision (CV; Gatys et al., 2016; Zhu et al., 2017). The task of text style transfer (TST) saw a surge in research interests after the inception of attention-based sequence-to-sequence text generation models. The essence of any of the tasks under TST is bringing changes in certain stylistic attributes while preserving the content. One such task is creating synthetic spoken parallel data using a written one (Figure 1). The task is novel, and minimal work is publicly available where both spoken and written genres are addressed distinctly.

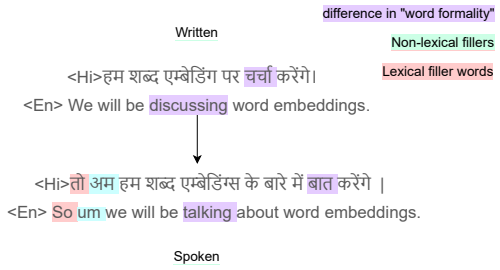

Figure 1: Text style transfer from written to spoken parallel sentences.

Because of its easy maintenance and availability, written data have been extensively experimented with. This works for most tasks; however, with the increasing popularity of processing speech, such as simultaneous translation of spoken language, the need for speech-specific data grows. However, transcribing large volumes of audio datasets is a tedious and costly process. In addition, creating a parallel dataset for such tasks requires the further step of translation. Applying methods of TST can offer a solution to this problem by utilizing the available large amount of written parallel text.

The utilization of written parallel data for the creation of spoken ones is not straightforward. A major issue with this task is having a clear distinction between spoken and written genres. Spoken genre spans a broad spectrum with spontaneous conversations or speeches at one extreme and prepared speeches at the other one. The lack of spontaneity in the latter case can make sentences indistinguishable from the written genre. The existing parallel

Proceedings of the 19th International Conference on Natural Language Processing (ICON), pages 224 - 233

datasets rarely contain genuine examples of spontaneous speech, as most of the filler words and pauses are cleaned while transcribing audio samples.

Most works in the field do not address the challenge of transfer between two similar linguistic styles. Since most of them are based on data-driven approaches, they are prone to failing in this setting due to insufficiently distinct features of the two genres. The common alternative, simple rule-based approaches model only content-independent features. We thus see a need for a combination of both approaches.

The main objective of this paper is to improve spoken translation systems. We address the lack of parallel spontaneous speech corpora using TST from written genre to spoken genre in the context of NMT. In this work, we try different data augmentation methods to construct synthetic spoken-style parallel dataset from existing written genre parallel datasets. We inject disfluencies in both languages on the phrase level. Even though not all disfluencies are consistent across languages in real life, we try to preserve them in the constructed parallel data, at least on the phrase level.

We propose a data-driven approach involving NMT models from both genres. We combine the encoders and decoders extracted from translation systems trained on each of the genres separately. We also propose a rule-based method for constructing spoken-style parallel data by injecting spoken features at the phrase-level to existing written parallel datasets. We check the applicability of the created synthetic parallel datasets for Hindi-English spoken language translation systems. To summarize, the main contributions of this paper are:

- We propose and implement a genre classifier for individual sentences.
- We propose a seq-to-seq model for translating from written genre to spoken genre.
- We provide a rule-based method for synthesizing spoken-style data from written genre examples.
- We evaluate the effectiveness of our genre transfer methods on a spoken language translation system for En-Hi language pair.

We discuss work related to our approach in Section 2. We provide an overview of the used datasets in Section 3. We describe our proposed genre classifier in Section 4 and continue with our data augmentation methods in Sections 5 and 6. We provide analysis of the results in Section 7. We discuss some alternatives and future directions in Section 8 and finally conclude our paper in Section 9.

We publish our source code, and pre-trained models on GitHub. [1]

2 Related Works

Our work touches upon three topics: Neural Machine Translation (Section 2.1), Text Style Transfer (Section 2.2), and Evaluation of Stylised Texts (Section 2.3).

2.1 Neural Machine Translation (NMT)

Since the advent of encoder-decoder-based methods (Cho et al., 2014), NMT has seen an uninterrupted flow of research interests. When given a large amount of training data, it has performed significantly better than the traditional methods. There has been considerable focus on techniques such as transfer learning (Zoph et al., 2016; Lakew et al., 2018) and data augmentation (Sennrich et al., 2016; Nguyen et al., 2020; Shen et al., 2020) to tackle the problem of low-resource settings. There has also been a limited amount of work on studying stylistic features in NMT outputs. For instance, Niu et al. (2017) use the lexical formality model to control the formality level of the NMT outputs. Wu et al. (2021) propose a bidirectional knowledge transfer framework to produce stylized translations.

2.2 Text Style Transfer (TST)

TST aims to control the stylistic attributes of a given text without changing the content. Stylistic attributes can range from politeness, formality, etc., to literary writing style. Some of the earlier works include Yan (2016); Ghazvininejad et al. (2016) for style transfer in poetry, Jhamtani et al. (2017) for Shakespearizing modern English, dos Santos et al. (2018) for controlling offensive language, and many more. However, due to the lack of parallel datasets, most of the solutions in TST revolve around unsupervised methods. Replacing style-specific words is one of the trivial and earlier solutions. However, the complexity of a natural language text can make this approach visibly suboptimal. One of the popular techniques is to disentangle the content and style dimensions in the latent

[1] https://github.com/knalin55/Genre-Transfer-in-NMT

	lex_fil		nlex_fil			fp		rep	ph_abbr		mean_len	
	en	hi	en	hi	en	hi	en	hi	en	hi	en	hi
Online Lec	38	37	2	0	21	21	11	4	8	-	23	25
OpenSub	1	1	0	0	18	18	0	0	22	-	6	7
Wikipedia	0	0	0	0	1	1	0	0	1	-	18	21
VOICE 2.0	26	-	21	-	25	-	25	-	12	-	15	-

Table 1: Data sources and some of their spoken features (out of 50 randomly examples).

space representing the input text (John et al., 2019; Yamshchikov et al., 2019). Although most recent works focus on this approach, they have a rather limited control over the model. Another interesting approach, which has gained popularity recently, is supervised training on a pseudo-parallel data. Prabhumoye et al. (2018) use back-translation for semantic preservation and adversarial training to generate texts in a specific style.

2.3 Evaluation of Stylised Texts

Evaluation for TST is challenging due to the subjectivity of styles. According to Mir et al. (2019), there are broadly three aspects of evaluation: style accuracy, content preservation, and natural and fluent output. In our case, along with these three aspects, we also need to ensure the translational equivalence of the sentences. Stylistic features are often independent of a set of particular lexicons. Thus, building a rule-based classifier is not so straightforward. A deep learning-based data-driven classifier can solve the issue but it faces data scarcity. Moreover, capturing content preservation using automatic evaluation metrics is even more challenging due to their reliance on similarity of the candidate translation and the reference. With style change, this similarity can be failing. For Wu et al. (2021), the objective is similar to ours. They trained a BERT-based classifier for the classification of formality. They use a language model for checking the fluency of the translated output, BLEU (Papineni et al., 2002) for the evaluation of translated outputs, and human evaluation for checking overall quality.

There are undoubtedly multiple parameters for judging the quality of stylized texts. Thus, we need to have multiple metrics covering all aspects. In our work, use BLEU to check translation quality, a genre classifier for style quality, and manual evaluation to check fluency and overall quality.

3 Data

We use the Samanantar corpus for our experiments. We split the corpus into written and spoken parts using the data sources. We consider sources from lectures (*Coursera*, *NPTEL*, *KhanAcademy*, and *Kurzgesagt*), and *OpenSubtitles* belonging to the spoken genre, and *Wikipedia* to the written genre. There are abundant examples where sentences have lexical filler words; however, very few contain non-lexical fillers or repetitions.

We have a total of $171,416$ parallel sentences for the spoken genre and $216,183$ for the written one. We separate $5,000$ sentences as a test set, $10,000$ as validation set, and the remaining sentences as a training set. We ensure no training sentence appears in the test or validation data.

Table 1 provides some statistics for spoken features of randomly sampled 50 sentences from each of our data sources. We calculate mean length of sentences over the whole data source.

3.1 Online Lectures

Online learning platforms are a great source of spoken genre data. Though they cannot be categorized as entirely spontaneous and are certainly well-prepared, they can contain a substantial amount of filler words. We cluster the sources $KhanAcademy$, $Coursera$, $Nptel$, and $Kurzgesagt$ together as online lectures. Table 1 shows they have longer sentences, and a considerable number of them have lexical filler words (lex_fil). However, all these filler words are in the form of sentence connectors (like, so, OK, well, etc.), and almost none of the examples had non-lexical fillers (nlex_fil) like, ermm, umm, uh, etc. 42% of the examples were in first-person (fp), 22% of the En examples had repetitions (8% of Hi examples; we suspect repetitions were removed while generating translations) and only 16% contained phonological abbreviations (ph_abbr). In summary, 43 out of 50 randomly sampled En sentences contained some form of listed spoken features. Though the spoken style quality and spontaneity of the spoken sources are substandard, these examples are the best we could have for the Hi language in the given genre.

Feature	Input text	en_{voice}	en_{base}
None	She later expressed regret for having cited an inaccurate study	0	0
Sentence conn	*So* she later regretted for having cited an inaccurate study	1	1
First person	*I* later regretted for having cited an inaccurate study	1	1
Filler word	She later regretted *er erm* for having cited an inaccurate study	1	0
Filler word	She later regretted *umm* for having cited an inaccurate study	0	0
Repetition	She later regretted *for for* having cited an inaccurate study	0	0

Figure 2: En genre classifiers' predictions compared against different spoken features. The first example is of written genre (0, as predicted by both models en_{base} and en_{voice}), while the rest are from spoken genre (1, which is not always predicted).

3.2 OpenSubtitles

OpenSubtitles is a collection of multilingual parallel corpora compiled from an extensive database of movies and TV subtitles. Since the dialogues and conversations are well rehearsed and prepared, they seem to have fewer fillers and repetitions (see Table 1). They also have shorter sentences. The sentences are understandably of the spoken genre; however, the lack of fillers and other spoken features might hurt the classifier and our MT model.

3.3 Wikipedia

Wikipedia is one of the most experimented written genre datasets available in the field. As expected, none of the randomly sampled examples contained any filler words or repetition.

3.4 VOICE 2.0

We use an additional source of En spoken monolingual corpus, VOICE 2.0 (Vienna-Oxford International Corpus of English), to train our En genre classifier. VOICE 2.0 is a collection of English spoken data. We use it as our additional data source for training our En genre classifier. Table 1 shows the dataset has a considerable amount of repetitions and filler words. The fillers contained a mix of both non-lexical and lexical words. 46 of the 50 randomly sampled examples had at least one of the spoken features mentioned in table. We take 62348 En sentences from the dataset for our classification experiment.

4 Genre Classifier

Classifying genre plays a vital role in the evaluation of genre transfer tasks. Majority of the papers in TST use deep-learning-based classifiers to classify the specific style. Along the lines of existing approaches, we train a BERT-based genre classifier. We train two models: with and without VOICE 2.0 dataset for En classifier. Since we have parallel sentences, we can analyze the results for En and draw also some conclusions for Hi language.

Model	Test f1 (%)
hi_{base}	96.07
en_{base}	97.84
en_{voice}	98.12

Table 2: Performance of genre classifiers

We use DistilBERT (Sanh et al., 2019) as our pretrained model for En language. It is a fast, cheap and light Transformer model trained by distilling BERT base. It has 40% fewer parameters and is 60% faster than bert-base-uncased. We train the model on En sentences from Samanatar corpus (en_{base}) and Samanantar + VOICE 2.0 corpus (en_{voice}). We use HuggingFace trainer for our experiments. We use the same tokenizer as DistilBert and set the sequence max length to 256. We train the models for 2 epochs with a batch size of 8 while keeping the best checkpoints at each 500 steps. We do not train the model further as its performance stopped improving after 2 epochs. We use 10000 sentences as test (comprised of examples from Samanantar and VOICE 2.0), 20000 sentences as validation, and the remaining sentences in the dataset as training data.

We use IndicBERT (Kakwani et al., 2020) as our pretrained model for Hi genre classification. IndicBERT is a multilingual AlBERT (Lan et al., 2019) trained on 12 Indic languages. The model has SOTA performance when compared to other multilingual models. We train the model on Hi sentences from *OpenSubtitles*, Online Lectures, and *Wikipedia* in Samanantar corpus (hi_{base}). We keep the hyperparameters similar to the En classifiers.

4.1 Results

We evaluate our En models on the Samanantar + VOICE 2.0 dataset test set. Since the training data has examples from diverse domains, there is less possibility for any bias towards a specific domain.

The En classifiers, en_{base} and en_{voice}, give F1

Algorithm 1: Algorithm to create a spoken sentence pair using a written sentence pair

Input $e_{written}$, $f_{written}$
Procedure `gen_spoken` ($e_{written}$, $f_{written}$) **:**
  e_{spoken}, $f_{spoken} \leftarrow list(), list()$
  $alignments$ = $get\_word\_alignments(e_{written}, f_{written})$
  $phrase\_align$ = $get\_phrase(alignments)$
  $phrase\_align.insert((init\_filler_e, init\_filler_f), index = 0)$ with some probability p_i
  for *phrase* $\in phrase\_align$ **do**
    phrase $\leftarrow add\_spoken\_features$(*phrase*) with some probability **P**
  $e_{spoken}.add(phrase_e \forall phrase_e \in phrase\_align)$
  $f_{spoken}.add(phrase_f \forall phrase_f \in phrase\_align)$
  return e_{spoken}, f_{spoken}
Function `get_phrase` ($word\_alignments$) :
  while *len(word_alignments) does not decrease* **do**
    for $align_i, align_{i+1} \in word\_alignments$ **do**
      if *step == 1* **then**
        concat($align_i, align_{i+1}$) if *set*($tgt(align_i) \bigcup tgt(align_{i+1})$) contains consecutive indices
      else
        concat($align_i, align_{i+1}$) if $src(align_i), src(align_{i+1})$ *overlap*
  return $word\_alignments$
Function `add_spoken_features` (*phrase,* ***P***) :
  $fillers_e$, $fillers_f$ = set of filler words in lang e, lang f
  with prob $\mathbf{P}_{fill}$, $phrase_e, phrase_f \leftarrow phrase_e.append(fillers_e[i]), phrase_f.append(fillers_f[i])$ for some i
  with prob $\mathbf{P}_{rep}$, $phrase_e, phrase_f \leftarrow phrase_e.append(phrase_e[i:]), phrase_f.append(phrase_f[i:])$ for some i
  return $word\_alignments$

scores of 97.84 and 98.12 respectively (Table 2). We consider en_{voice} for our further experiments, as it has slightly better performance than the other one. To confirm the dependency of the model on spoken features, we check its behavior on similar spoken and written genre sentences. Figure 2 clearly shows the dependence of the model en_{voice} on filler words and the use of the first person for the given example. A similar conclusion can also be drawn from Figure 3. It shows the last layer's mean attention scores for en_{voice} model corresponding to [CLS] token. It can be observed that the tokens corresponding to *So*, *er*, *erm* have slightly higher attention scores than the others.

However, it fails to recognize repetition as a spoken feature. We suspect the lack of a significant amount of text with repetitions can be attributed to this behavior. The same can be observed for en_{base} as well. Unlike the former model, en_{base} also fails when non-lexical filler words are used. The fourth example in Figure 2 makes another interesting case. When the non-lexical filler *em erm* is replace with *umm*, the model fails to predict it correctly.

We have performance of hi_{base} similar to en_{base}. Though it has the F1 score of 96.07%, it fairly depends ONLY on first person and lexical fillers.

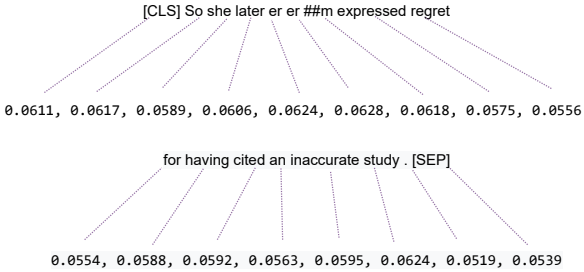

Figure 3: Attention scores of the last self-attention layer (for en_{voice})

5 Rule-Based Injection of Spoken Language Features

The ultimate objective of our work is to improve translation quality of spoken language translation systems. However, spoken parallel corpora are rare, and the existing ones clearly lack spontaneity as evident from our previous discussion. In this work, we try to create spontaneous synthetic parallel corpus using the written corpora.

We propose a rule-based data augmentation method to add spoken features to the existing written examples on the phrase level (see Algorithm 1). The spoken texts tend to have informal words. They also have lesser content words with more grammatical words. Thus, we first back-translate

En (written): Krishnapuram is a village in Krishna district of the Indian state of Andhra Pradesh.
Hi (written): कृष्णारावुपालें कृष्णा में भारत के आन्ध्रप्रदेश राज्य के अन्तर्गत के कृष्णा जिले का एक गाँव है।
En (spoken_bt): Actually here's a village in ehh the Indian state ofland.
Hi (spoken_bt): वास्तव में अफगानिस्तान में भारतीय राज्य में भारत के भीतर भाग में एक गाँव है अ।
En (spoken): So Krishnapuram is a village in ehh Krishna district of the Indian state of Andhra Pradesh.
Hi (spoken): तो कृष्णारावुपालें कृष्णा में भारत के आन्ध्रप्रदेश राज्य के अन्तर्गत के कृष्णा जिले का एक गाँव है अ।

En (written): She later expressed regret for having cited an inaccurate study.
Hi (written): बाद में उसने एक गलत अध्ययन का हवाला देते हुए खेद व्यक्त किया।
En (spoken_bt): Later he err expressed regret by quoting a wrong study.
Hi (spoken_bt): बाद में उसने अ एक ग़लत अध्ययन को उद्धृत करने के द्वारा खेद व्यक्त किया ।
En (spoken): So she later expressed regret for having cited an inaccurate study.
Hi (spoken): तो बाद में उसने एक गलत अध्ययन का हवाला देते हुए खेद व्यक्त किया।

Figure 4: Two examples after applying algorithm 1. Highlighted sentences are of the written genre. *spoken_bt* denotes examples after applying algorithm on back-translated sentences, while *spoken* is without back-translation

the written genre parallel sentences to normalize them (Prabhumoye et al., 2018). We then create word alignments between En and Hi sentences using multilingual BERT (Devlin et al., 2018). We start with bigrams in the source language. Since we are interested in phrase alignments, we keep those bigrams with their indices of corresponding alignments belonging to a continuous span of numbers. We combine two bigrams and their corresponding alignments if they have overlapping indices. This ensures we get valid phrases and their mapping from En to Hi. We perform this combination process until convergence. After identifying phrases from the given word alignments, we add spoken features like non-lexical filler words (erm, emm, umm, eh, etc.), lexical filler words (so, like, etc.), and repetitions on both source and target sides with some probability P. We also add fillers at the beginning of sentences with probability p_i.

Figure 4 illustrates that back-translated outputs don't make enough sense. This can be attributed to the low quality of our En→Hi and Hi→En NMT models. Thus, we resort to using examples without back-translation. We denote this model by M_{rule}.

6 NMT with Genre Transfer

The previous method fails to transfer spoken features like informal words, and introducing more grammatical words. To counter this, we try data driven approaches. We stitch models trained on written and spoken genres together. We use this model to get translations across genres. We also try experiments with fine-tuning the stitched model on the augmented data created in Section 5.

We use pretrained NMT models by Tiedemann and Thottingal (2020) to fine-tune on our task. They train MarianMT (Junczys-Dowmunt et al., 2018) model on OPUS corpus (Tiedemann and Nygaard, 2004), and made the models publicly available. We fine-tune the models on the spoken and written genre datasets and label them as M_{sp} and M_{wr}, respectively, for both translation directions. We use Huggingface Trainer for our training purposes. We train the spoken and written models for 10 epochs with a batch size of 8. We evaluate at every 500 steps while using early stopping callback. During inference, we stitch M_{sp} and M_{wr} to get M_{mix}. We use the models to translate from En (and Hi) in written to Hi (and En) in spoken genre. We evaluate the generated parallel data on the written genre test set.

6.1 Model Stitching

During the training phase, for each translation direction, we train the encoder-decoder based NMT models on parallel data of styles wr (written) and sp (spoken). We get four models eventually: $M^{wr}_{en\rightarrow hi}$, $M^{sp}_{en\rightarrow hi}$, $M^{wr}_{hi\rightarrow en}$, and $M^{sp}_{hi\rightarrow en}$. We switch the encoders from sp models with wr models. We use the resulting model to translate text en of genre wr to hi of genre sp (and wr hi to sp en).

In general, each model M^j_i can be represented with $M^j_i(x) = f^{dec}_{i,j}(f^{enc}_{i,j}(x))$, for $i \in \{en \rightarrow hi, hi \rightarrow en\}$ and $j \in \{wr, sp\}$. We hypothesize that $f^{enc}_{i,j}$ encodes input x in a latent space independent of j. Thus, during inference (for x of style wr), to get translations of style sp, we can use $f^{dec}_{i,sp}$ with $f^{enc}_{i,wr}$.

6.2 Fine-tuning on Augmented Data

The previous method fails to give outputs in spontaneous spoken genre, as the existing training corpora have very few spontaneous examples. Thus, to get more natural-looking spontaneous spoken

Model	BLEU		(%) spoken	
	Hi-En	En-Hi	En	Hi
M_{wr}	34.31	35.11	2.33	3.19
M_{sp}	18.00	20.51	20.55	11.31
M_{mix}	19.27	20.88	20.98	12.32
M_{data_aug}	14.00	18.81	34.63	17.03
M_{rule}	46.93	76.09	21.70	15.05

Table 3: Overview of results. "% spoken" denotes proportion of generated outputs labelled as spoken.

	Content preservation		Fluency	
	Hi-En	En-Hi	Hi-En	En-Hi
M_{wr}	2.8	3.2	3.5	3.5
M_{sp}	1.7	2.5	3.0	3.4
M_{mix}	2.2	2.5	3.3	3.5
M_{data_aug}	2.0	2.2	2.9	3.2
M_{rule}	4.0	4.0	4.0	4.0

Table 4: Manual evaluation results

examples, we fine-tune our stitched model on the augmented data to get better spoken translations. We train it for 5 epochs while keeping other hyperparameters the same as the other experiments.

7 Results

7.1 Genre Preservation

We evaluate the performance of our spoken data generation methods using automatic and manual evaluation. We use BLEU to check the quality of translations across genres and our BERT-based genre classifiers for En and Hi to get an estimated proportion of translated outputs in spoken genre. For manual evaluation, we randomly sample 50 outputs and evaluate them for content preservation (0-4; with 4: preserving all content, 2: one of context or nouns is preserved, 0: nothing preserved) and fluency (0-4; with 4 being most fluent).

Automatic and manual evaluation results can be found in Tables 3 and 4, resp.

M_{rule} outperforms other data-driven models in terms of BLEU, content preservation, and fluency scores. This is expected, as the input sentences stay the same apart from the injected spoken features. Among the data-driven approaches, M_{data_aug} deviates the most from the input sentences. The addition of explicit spoken features might be one of the significant reasons.

Interestingly, even though the spoken and written models are trained on a completely different dataset, without any overlap, the outputs generated from M_{mix} are quite good. M_{sp} has similar performance to M_{mix} for En→Hi direction. En verb forms, unlike Hi, are independent of the person. This can be one of the reasons for having a more generalized latent representation in the case of genre-specific NMT. Thus, switching encoders on the En side does not significantly affect the score.

Even though the classifier is biased towards specific spoken features, we can still use it to evaluate certain features. As expected, the genre score ("% spoken" in Table 3) for M_{wr} is pretty low. Although the scores for other models are relatively low, they are better than the written model. M_{data_aug} performs better than others in terms of the number of spoken sentences generated. The outputs look more naturally spoken than M_{rule} with less complex words and more grammatical words. However, it fails to align the filler words and repetitions in generated parallel data. This is obvious, as the models have not seen the positions of explicit spoken features in the other language. Also, since we are translating sentences across genres, the model seems to hallucinate while adding repetitions.

The model stitching method fails to inject even the lexical fillers it has seen during training. Out of 50 randomly sampled sentences, only 4 contained lexical features, contrary to the spoken training data, where almost 75% of the sampled examples contained such features. This is where M_{data_aug} and M_{rule} gets an advantage of controlling spoken features. Another issue with all data-driven approaches is the quality difference between Hi-En and En-Hi translation models. This affects the quality of parallel'ness of the augmented data.

Due to the difference in domains of written and spoken genres, the data-driven approaches struggle while handling proper nouns. This, along with unnecessary repeated addition of spoken features, results in a dip in performance of M_{data_aug}. This, however, can again be controlled by training it on augmented data with different frequencies of spoken features. The performance of M_{data_aug} improves with a decrease in the probability of the addition of spoken features. However, the alignment problem of such features still bothers the quality of generated parallel data. This is clearly handled during the rule-based approach, which makes it perform better than the other approaches.

7.2 Spoken Translation Quality

We check the utility of our spoken data generation methods via M_{sp}. We fine-tune M_{sp} further on 50k parallel data created using M_{mix}, M_{data_aug}, and our rule-based method M_{rule}. We label the models as SM_{M_Mix}, $SM_{M\_data\_aug}$, and SM_{data_aug} respec-

En: er so er i mean i i think erm you can be from er another country and i mean two different countries and then use a third language as linguafranca and then i think then it's a l- lingua franca
Hi (M_{sp}): मेरा मतलब है कि मुझे लगता है कि आप एक और देश से हो सकते हैं और मेरा मतलब है कि दो अलग-अलग देश और फिर एक तीसरी भाषा का उपयोग माफिया के रूप में करें और फिर मुझे लगता है कि यह एक ली-लिंगीका है।
Hi (SM_{data_aug}): तो अ मेरा मतलब है अम्म मुझे लगता है कि आप किसी दूसरे देश से अ हो सकते हैं और मेरा मतलब है कि दो अलग-अलग देश हैं और फिर लंगुफिया के रूप में एक तीसरी भाषा का उपयोग करें और फिर मुझे लगता है कि यह एक ल-हुआका है।

En: er i don't i don't know if a- a- am i right
Hi (M_{sp}): मैं नहीं जानता कि अगर एक-मैं सही हूँ-
Hi (SM_{data_aug}): अ मैं नहीं जानता कि मैं नहीं जानता कि क्या - मैं सही हूं

Figure 5: Two example outputs on VOICE dataset using M_{sp} and SM_{data_aug}

	hi-en	en-hi
M_{sp}	44.04	44.47
SM_{M_mix}	33.98	39.65
$SM_{M\_data\_aug}$	19.53	19.34
SM_{data_aug}	33.97	39.65

Table 5: BLEU scores of spoken translation systems on spoken test set from Samanantar (non-spontaneous)

	Cont. Pres.	Fluency	Ft_sc.
M_{sp}	2.7 ± 1.08	2.6 ± 1.07	0.14
SM_{M_mix}	1.9 ± 2.02	2.0 ± 2.11	0.14
$SM_{M\_data\_aug}$	2.6 ± 1.64	2.7 ± 1.76	0.28
SM_{data_aug}	**3.7** ± 0.42	**3.8** ± 0.42	**3.10**

Table 6: Manual evaluation results of En→Hi Spoken Translation System on VOICE 2.0. The results are calculated for 50 randomly sampled translated outputs.

tively. We first test the models on the test set of spoken parallel data from Samanantar using BLEU (Table 5). We also evaluate the En→Hi spoken translation system on VOICE 2.0 dataset manually (Table 6). Precisely, we use content preservation scores, fluency and spoken feature scores. We use spoken feature scores (Ft_sc.; +2 for fillers; +2 for repetitions) to check the quality of type and placement of filler words and repetitions.

Table 5 shows a drop in performance for models fine-tuned on augmented data on the spoken test set. The addition of sentences from new domain, along with updated syntactic structure seems to bring noise when compared with the spoken genre dataset. $SM_{M\_data\_aug}$ has the worst dip in the performance on the Samanantar test set. It tends to add extra filler words even if they are not present in the source sentence. This might be due to the misalignments of fillers and repetitions in the synthetic parallel sentences. The other two models SM_{data_aug} and SM_{M_mix} have comparable performance on the test set.

We check the En-Hi spoken translation systems quality on the VOICE dataset. SM_{data_aug} performs the best, and the placement of filler words and repetitions are also relatively accurate. The other three models struggled with such features. Figure 5 shows some translation outputs of SM_{data_aug}.

8 Discussion

The En→Hi spoken translation system fine-tuned on the synthesized data from our rule-based method performs quite well for spontaneous En examples. It not only applies the repetitions well, but also introduces the filler words at correct places. On the other hand, the baseline model failed to recognize and add fillers and repetitions. Since we didn't have any spontaneous examples in Hi, we could not empirically evaluate Hi→En. However, we expect similar performance from that as well.

We note that our output style evaluation relies on our genre classifier. For En, it has a varied training set but still it fails to work with unseen non-lexical fillers. For Hi, the classifier is even less reliable and should be used with caution. Clearly, there is a need for better genre classifiers.

It would be interesting to see the results for M_{mix} fine-tuned on back-translated data with added spoken features. Due to lower quality back-translation outputs, we could not perform the experiments with it. One approach can be finetuning the models on the Samanantar dataset for the Hi-En language pair and then using the model for back-translation. Another interesting experiment can also be checking the dependency of the model on controlled spoken features in training data and the extent to which they can be added without disturbing the content.

9 Conclusion

This paper proposes two main methods to synthesize parallel spoken data using existing written-genre parallel texts. Given the written input, the data-driven method produces a naturally-looking spoken output; however, it fails to ensure appro-

priate parallelism of explicit spoken features. The alternative rule-based approach has precise alignments of such features. Furthermore, we check the usability of the created parallel texts by fine-tuning Hi-En NMT models on merely 50k sentence pairs. The model fine-tuned on the synthetic corpus created using our rule-based method gives the best results. For En→Hi translation, it produces relatively decent results, and unlike the baseline model, it introduces correct non-lexical fillers at the proper places.

10 Acknowledgement

This work was supported by the GAČR grant Neural Representations in Multi-modal and Multilingual Modelling (NEUREM3, 19-26934X).

References

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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.

Cicero dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423.

Marjan Ghazvininejad, Xing Shi, Yejin Choi, and Kevin Knight. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1191.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*, pages 10–19.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

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.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

SM Lakew, A Erofeeva, M Negri, M Federico, and M Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. In *15th International Workshop on Spoken Language Translation*, pages 54–62.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 495–504.

Xuan-Phi Nguyen, Shafiq Joty, Kui Wu, and Ai Ti Aw. 2020. Data diversification: A simple strategy for neural machine translation. *Advances in Neural Information Processing Systems*, 33:10018–10029.

Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark. 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.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version

of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96. Association for Computational Linguistics (ACL).

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Jörg Tiedemann and Lars Nygaard. 2004. The OPUS corpus - parallel and free: `http://logos.uio.no/opus`. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Xuanxuan Wu, Jian Liu, Xinjie Li, Jinan Xu, Yufeng Chen, Yujie Zhang, and Hui Huang. 2021. Improving stylized neural machine translation with iterative dual knowledge transfer. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3971–3977. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ivan P. Yamshchikov, Viacheslav Shibaev, Aleksander Nagaev, Jürgen Jost, and Alexey Tikhonov. 2019. Decomposing textual information for style transfer. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 128–137, Hong Kong. Association for Computational Linguistics.

Rui Yan. 2016. i, poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema. In *IJCAI*, volume 2238, page 2244.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.