Investigating Lexical Replacements for Arabic-English Code-Switched Data Augmentation

Injy Hamed,[1,2] **Nizar Habash,**[1] **Slim Abdennadher,**[3] **Ngoc Thang Vu**[2]
[1]Computational Approaches to Modeling Language Lab, New York University Abu Dhabi
[2]Institute for Natural Language Processing, University of Stuttgart
[3]Informatics and Computer Science, The German International University in Cairo
injy.hamed@nyu.edu

Abstract

Data sparsity is a main problem hindering the development of code-switching (CS) NLP systems. In this paper, we investigate data augmentation techniques for synthesizing dialectal Arabic-English CS text. We perform lexical replacements using word-aligned parallel corpora where CS points are either randomly chosen or learnt using a sequence-to-sequence model. We compare these approaches against dictionary-based replacements. We assess the quality of the generated sentences through human evaluation and evaluate the effectiveness of data augmentation on machine translation (MT), automatic speech recognition (ASR), and speech translation (ST) tasks. Results show that using a predictive model results in more natural CS sentences compared to the random approach, as reported in human judgements. In the downstream tasks, despite the random approach generating more data, both approaches perform equally (outperforming dictionary-based replacements). Overall, data augmentation achieves 34% improvement in perplexity, 5.2% relative improvement on WER for ASR task, +4.0-5.1 BLEU points on MT task, and +2.1-2.2 BLEU points on ST over a baseline trained on available data without augmentation.

1 Introduction

Code-switching (CS) is the alternation of language in text or speech. CS can occur at the levels of sentences (inter-sentential CS), words (intra-sentential CS/code-mixing), and morphemes (intra-word CS/morphological CS). Given that CS data is scarce and that collecting such data is expensive and time-consuming, data augmentation serves as a successful solution for alleviating data sparsity.

In this paper, we investigate lexical replacements for augmenting CS dialectal Arabic-English data. Researchers have investigated approaches that do not require parallel data, including translating source words into target language with the use of dictionaries (Tarunesh et al., 2021), machine translation (Li and Vu, 2020), and word embeddings (Sabty et al., 2021), as well as relying on parallel data and performing substitutions of words/phrases using alignments (Menacer et al., 2019; Appicharla et al., 2021; Gupta et al., 2021). As will be discussed in Section 2, most of the previous studies on this front have focused on one augmentation technique without exploring others, or reported results using only one type of word alignments configuration, or evaluated effectiveness of augmentation on only one downstream task.

We attempt to provide a comprehensive study where we systematically explore the use of neural-based models to decide on CS points for performing replacements using word-aligned parallel corpora versus randomly-chosen CS points, along with the interaction of different alignment configurations. We compare these approaches against dictionary-based replacements. We provide a rigorous evaluation of the different settings, where we assess the quality of the generated CS sentences through human evaluation as well as the impact on language modeling (LM), automatic speech recognition (ASR), machine translation (MT), and speech translation (ST) tasks.

Our human evaluation study shows that for the purpose of generating high-quality CS sentences, learning to predict CS points and integrating this information in the augmentation process improves the quality of generated sentences. On the downstream tasks, we report that performing alignment-based replacement outperforms dictionary-based replacement. For alignment-based replacement, utilizing a predictive model to decide on where CS points should occur as opposed to replacing at random positions both lead to similar results for ASR, MT, and ST tasks. For both approaches, we investigate different word alignment configurations, and we report that performing segment replacements using symmetrized alignments outperforms

Proceedings of the The Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023), pages 86–100

word-replacements using intersection alignments on both human evaluation and extrinsic evaluation. We also investigate controlling the amount of generated data, to eliminate the effect of random producing more data over the predictive model. Under the constrained condition, using a predictive model outperforms the random approach on the MT task.

In this work, we tackle the following research questions (RQs):

- **RQ1:** Can a model learn to predict CS points using limited amount of CS data?
- **RQ2:** Can this information be used to generate more natural synthetic CS data?
- **RQ3:** Would higher quality of synthesized CS data necessarily reflect in performance improvements in downstream tasks?

2 Related Work

Most of the work done for CS data augmentation has been focused on LM, mostly for ASR. Several techniques have been proposed based on linguistic theories (Pratapa et al., 2018; Lee et al., 2019; Hussein et al., 2023), heuristics (Shen et al., 2011; Vu et al., 2012; Kuwanto et al., 2021a), neural networks (Chang et al., 2018; Winata et al., 2018, 2019; Li and Vu, 2020), and MT (Tarunesh et al., 2021). CS data augmentation has been less investigated for MT. Previous work has mainly involved lexical replacements (Menacer et al., 2019; Song et al., 2019; Appicharla et al., 2021; Gupta et al., 2021; Xu and Yvon, 2021) and back translation (Kuwanto et al., 2021b). In this section, we discuss previous work that we find closest to ours.

Hussein et al. (2023) generated synthetic CS Arabic-English text based on the equivalence constraint (EC) theory (Poplack, 1980) using the GCM tool (Rizvi et al., 2021), as well as random lexical replacements. It was shown that while relying on the EC theory generates more natural CS sentences, as shown in human evaluation, using lexical replacements outperforms the linguistic-based approach on LM and ASR tasks.

In the direction of lexical replacements, Appicharla et al. (2021) generated synthetic CS Hindi-English sentences by replacing all source words (except for stopwords) by the corresponding target words using 1-1 alignments, achieving improvements on MT task. Gupta et al. (2021) trained a neural-based model to predict CS points on monolingual source text. Using 1-n alignments, the source word is replaced by the aligned word(s). They evaluate their approach against unigram and bigram random replacements, and test its effectiveness on MT task for CS Hindi-English. Xu and Yvon (2021) use data augmentation for MT task for CS Spanish-English and French-English. Symmetrized alignments are used to identify small aligned phrases (minimal alignment units) and phrase replacements are performed randomly. We also notice that in literature, human evaluation of generated CS data is mainly used to evaluate the synthetic data produced by the best model, rather than comparing different techniques. Such a comparison was provided by Pratapa and Choudhury (2021), where a large-scale human evaluation was presented comparing different linguistic-driven and lexical replacement techniques. However, the study was focused on human evaluation without exploring the effectiveness of those techniques on downstream tasks.

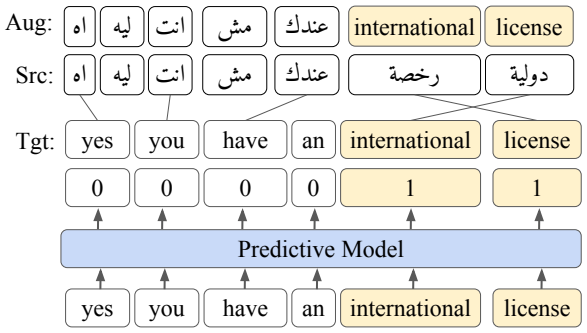

Figure 1: Data augmentation process.

3 Data Augmentation

For generating synthetic CS data, we investigate the use of word-aligned parallel sentences as well as dictionary-based replacements. In the latter approach, monolingual Arabic sentences are augmented by replacing words at random locations with their English glossary entry. In the former approach, utilizing monolingual Arabic-English parallel corpora, we inject words from the target side to the source side, where replacements are performed at random locations or using a CS point predictive model. As shown in Figure 1, the augmentation process consists of two main steps: (1) CS point prediction: identifying the target words to be borrowed, and (2) CS generation: performing the replacements. In Sections 3.1 and 3.2, we will elaborate on the methodology for both steps.

	Examples
Src	← و i was a junior ta في فترة فجربت الموضوع ف i love ال academic life شوية .
Tgt	and i was a junior ta for a period of time so i have tried this and i love the academic life a bit .
Output	0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0
Src	← ماكنتش م expect اني اشوف city زي دي اساسا
Tgt	i wasn 't expecting to see such a city in the first place .
Output	0 0 0 1 0 0 0 0 1 0 0 0 0 0

Table 1: Example showing the matching algorithm output for given source and target sentences. The matched words on the target side are underlined. The arrows show the sentence starting direction, as Arabic is read right to left.

3.1 CS Point Prediction

Similar to Gupta et al. (2021), we model the task of CS point prediction as a sequence-to-sequence classification task. The neural network takes as input the word sequence $x = \{x_1, x_2, .., x_N\}$, where N is the length of the input sentence. The network outputs a sequence $y = \{y_1, y_2, .., y_N\}$, where $y_n \in \{1,0\}$ represents whether the word x_n is to be code-switched or not. We learn CS points using ArzEn-ST corpus (Hamed et al., 2022b), which contains CS Egyptian Arabic-English sentences and their English translations. We then utilize the learnt CS model to augment a large number of monolingual Arabic-English parallel sentences by inserting the tagged words on the (English) target side into the (Egyptian Arabic) source side.

In order to learn CS points, the neural network needs to take as input monolingual sentences from either the source or target sides, along with tags representing whether this word should be code-switched or not. In Gupta et al. (2021), the authors generated synthetic monolingual sentences from CS sentences by translating CS segments to the source language, and then learning CS points on the source side. While this approach seems more intuitive, CS segments abide by the grammatical rules of the embedded language, thus direct translation of embedded words would result in sentences having incorrect structures in the matrix language in case of syntactic divergence, which is present between Arabic and English. Instead, we opt to learn CS points on the target side. This approach provides another advantage, as English is commonly used in CS, having the predictive model trained on English as opposed to the primary language (which could be low-resourced) allows for the use of available resources such as pretrained LMs.

The challenge in this approach is identifying the words on the target side which correspond to the CS words on the source side. Relying on the translators to perform this annotation task is costly, time consuming, and error-prone.[1] Relying on word alignments is also not optimal, where only 83% of CS words in ArzEn-ST train set were matched using intersection alignment. Recall could increase using a less strict alignment approach, but would be at the risk of less accurate matches. Therefore, we develop a matching algorithm that is based on the following idea: if a CS segment occurs x times in the source and target sentences, then we identify these segments as matching segments. We match segments starting with the longest segments (and sub-segments) first. When matching words, we check their categorial variation (Habash and Dorr, 2003) as well as stems to match words having slight modifications in translation.[2] This matching algorithm provides a language-agnostic approach to identify words on the target side that are code-switched segments on the source side.[3] Examples of algorithm output are shown in Table 1, where it is seen that *expect* and *expecting* are matched as a result of the categorial variation check.

3.2 CS Generation

After identifying the target words to be embedded into the source side, we rely on alignments using GIZA++ (Casacuberta and Vidal, 2007) to perform the replacements. While direct replacements can be performed in the case of single word switches, in the case of replacing multiple consecutive words, direct word replacements would produce incorrect CS structures in the case of syntactic divergence.

[1] We have tried this annotation task for ArzEn-ST and only 72% of the CS words got annotated.

[2] In case $|matches_{tgt}| > |matches_{src}|$, we first rely on alignments to make the decision, achieving 99.6% matches on ArzEn-ST train set, then we randomly pick matched target segments to cover the number of matches on the source side in order to increase recall.

[3] Code available: http://arzen.camel-lab.com/

In the case of Arabic-English, this is particularly evident for adjectival phrases. Accordingly, when performing word replacements, we maintain the same order of consecutive English words, which we refer to as the "Continuity Constraint". In Figure 2, the importance of applying this constraint is illustrated. Without such a constraint, the generated sentence outlined in Figure 2 would follow the Arabic syntactic structure resulting in "ده topic important very" (*this [is a] topic important very*).

When performing replacements, we investigate the use of intersection alignments as well as grow-diag-final alignments.[4] While intersection alignment provides high precision, relying on 1-1 alignments is not always correct, as an Arabic word can map to multiple English words and vice versa. Therefore, we investigate the use of grow-diag-final (symmetrized) alignments to identify aligned segments. The aligned segments consist of pairs of the minimal number of consecutive words (S,T) where all words in source segment (S) are aligned to one or more words in target segment (T) and are not aligned to any other words outside (T), with the same constraints applying in the opposite (target-source) direction. Afterwards, for each English word receiving a positive CS tag, the whole target segment containing this word replaces the aligned source segment. Throughout the paper, we will refer to the two approaches as using 1-1 and n-n alignments. In Figure 3, we present an example showing the results of augmentation using predictive CS models versus random CS point prediction along with using 1-1 or n-n alignments.

3.3 Augmentation Approaches

We investigate the following approaches:

DICTIONARY**:** We randomly pick x source words and replace them with an English glossary entry using MADAMIRA (Pasha et al., 2014). We set x to 19% of the source words, where this number is chosen based on the percentage of English words in CS sentences in ArzEn-ST train set, given that we would like to mimic natural CS behaviour.

[4]We experiment with relying on alignments trained on word space only, stem space only, and the merge of both alignments, where for intersection alignments, we first rely on the alignments obtained in stem space, and add remaining alignments obtained from word space, such that 1-1 alignments are retained, and for grow-diag-final alignments, we take the union of alignments in both spaces. We find that merging alignments in both spaces achieves higher alignment coverage as well as better results in extrinsic tasks. Therefore, we will only be presenting the results using the merged alignments.

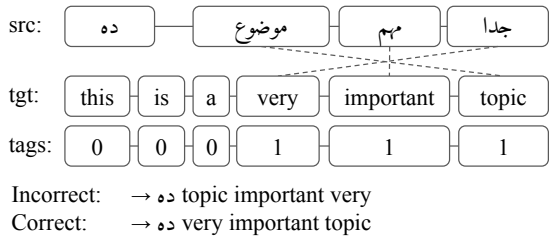

Figure 2: Data augmentation under the Continuity Constraint.

	.	الضهر	بعد	بكرة	معاك	معاد	عاوز
i							
'd							x
like							
an							
appointment						x	
with					x		
you							
tomorrow				x			
afternoon		x					
.	x						

Rand (1-1)	← عاوز معاد معاك بكرة بعد afternoon . *I'd like an appointment with you tomorrow after afternoon*
Rand (n-n)	← عاوز معاد معاك بكرة afternoon . *I'd like an appointment with you tomorrow afternoon*
Pred (1-1)	→ i 'd appointment معاك بكرة بعد الضهر . *I'd appointment with you tomorrow afternoon*
Pred (n-n)	→ i 'd like an appointment معاك بكرة بعد الضهر . *I'd like an appointment with you tomorrow afternoon*

Figure 3: Example showing 1-1 and n-n alignments. The intersection alignments are marked with 'x' and the grow-diag-final alignments are highlighted. We show the generated sentences with translations for each setup.

MAPRAND**:** We randomly pick x target words having source-target intersection alignments. We set x to 19% of the source words. We use word and segment replacements, where the models are referred to as MAPRAND$_{1-1}$ and MAPRAND$_{n-n}$.

MAPPRED**:** We fine-tune pretrained mBERT model using NERDA framework (Kjeldgaard and Nielsen, 2021) to predict the target words to be injected into the source side.[5] We use 1-1 and n-n alignments to perform replacements, where the models are referred to as MAPPRED$_{1-1}$ and MAPPRED$_{n-n}$.[6] For finetuning mBERT, we set the epochs to 5, drop-out rate to 0.1, warmup steps to 500, batch size to 13, and learning rate to 0.0001.

[5]We maintain the original tokenization of the input text, where we project further tokenization performed on the output into the original tokenization.

[6]For training the predictive models, we also tried using BERT models, which gave slightly lower results.

4 Experiments

4.1 Data

We use ArzEn-ST corpus (Hamed et al., 2022b) as our CS corpus. The corpus contains English translations of an Egyptian Arabic-English code-switched speech corpus (Hamed et al., 2020) that is gathered through informal interviews with bilingual speakers. The corpus is divided into train, dev, and test sets having $3.3k$, $1.4k$, and $1.4k$ sentences (containing $2.2k$, $0.9k$, and $0.9k$ CS sentences), respectively. We follow the same data splits. In Appendix A, we provide an overview of ArzEn-ST corpus.

We also utilize the following Egyptian Arabic-English parallel corpora: Callhome Egyptian Arabic-English Speech Translation Corpus (Gadalla et al., 1997; LDC, 2002b,a; Kumar et al., 2014), LDC2012T09 (Zbib et al., 2012), LDC2017T07 (Chen et al., 2017), LDC2019T01 (Chen et al., 2019), LDC2021T15 (Tracey et al., 2021), and MADAR (Bouamor et al., 2018). The corpora contain $308k$ monolingual parallel sentences as well as $15k$ CS parallel sentences. We use the same data splits as defined for each corpus. For corpora with no defined data splits, we use the guidelines provided in (Diab et al., 2013). Data preprocessing for ArzEn-ST and the parallel corpora is discussed in Appendix C.

Data Augmentation: For data augmentation, we use the monolingual parallel sentences and augment them into CS parallel sentences. For the CS point predictive model, we use the CS sentences in ArzEn-ST train and dev sets for training and development, respectively.

MT: The MT baseline system is trained on ArzEn-ST train set, in addition to the $308k$ monolingual parallel sentences. In the augmentation experiments, we add the augmented sentences to the baseline training data. For development and testing, we use ArzEn-ST dev and test sets.

ASR: The ASR baseline system is trained on the following Egyptian Arabic data: ArzEn speech corpus (Hamed et al., 2020), Callhome (Gadalla et al., 1997), and MGB-3 (Ali et al., 2017). A subset of 5-hours was used from each of Librispeech (Panayotov et al., 2015) (English) and MGB-2 (Ali et al., 2016) (MSA), where adding more data from these corpora deteriorated the ASR performance (Hamed et al., 2022a). The LM baseline model is trained on corpora transcriptions. For the LM models using augmented data, we append the augmented data to those transcriptions. For development and testing, we use ArzEn-ST dev and test sets.

As an extra experiment, we compare the performance of the systems relying on synthetic CS data versus using available real CS data. For MT, we use the $15k$ CS parallel sentences in addition to the baseline data. For ASR rescoring, we train the LM on the baseline data in addition to 117,844 code-switched sentences collected from social media platforms (Hamed et al., 2019). We denote these experiments as *ExtraCS* in the results.

4.2 Machine Translation System

We train a Transformer model using Fairseq (Ott et al., 2019) on a single GeForce RTX 3090 GPU. We use the hyperparameters from the FLORES benchmark for low-resource machine translation (Guzmán et al., 2019).[7] The hyperparameters are given in Appendix D. We use a BPE model trained jointly on source and target sides with a vocabulary size of $16k$ (which outperforms $1, 3, 5, 8, 32, 64k$).[8] The BPE model is trained using Fairseq with character_coverage set to 1.0.

4.3 Automatic Speech Recognition System

We train a joint CTC/attention based E2E ASR system using ESPnet (Watanabe et al., 2018). The encoder and decoder consist of 12 and 6 Transformer blocks with 4 heads, feed-forward inner dimension 2048 and attention dimension 256. The CTC/attention weight (λ_1) is set to 0.3. SpecAugment (Park et al., 2019) is applied for data augmentation. For LM, the RNNLM consists of 1 LSTM layer with 1000 hidden units and is trained for 20 epochs. For decoding, the beam size is 20 and the CTC weight is 0.2.

4.4 Speech Translation System

We build a cascaded ST system using the ASR and MT models. We opt for a cascaded system over an end-to-end system due to the limitation of available resources to build an end-to-end system, in addition to the fact that cascaded systems have shown to outperform end-to-end systems in low-resource settings (Denisov et al., 2021).

[7] We follow (Gaser et al., 2022), where it was shown that FLORES hyperparameters outperform Vaswani et al. (2017) using the same datasets.

[8] For the *ExtraCS* experiment, we use a vocabulary size of $8k$, which outperforms $16k$ and $32k$.

5 Results

In order to evaluate our augmentation techniques, we provide intrinsic evaluation, extrinsic evaluation, as well as human evaluation.[9] According to human evaluation, the synthetic data generated using a CS predictive model is perceived as more natural. However, our extrinsic evaluation shows that both aligned-based approaches (random replacements and relying on a predictive model) perform equally on downstream tasks. We observe that using a predictive model generates less data than the random approach. When controlling for size, we observe that using a predictive model brings improvements on the MT task. Both aligned-based approaches outperform dictionary-based replacements on human evaluation and extrinsic evaluation. Regarding the effect of word alignment configurations, the improvements of using n-n alignments versus 1-1 alignments is confirmed in both human evaluation and extrinsic evaluation.

5.1 Intrinsic Evaluation

Predictive Model Evaluation We compare the CS point predictions provided by the predictive model against the actual CS points in the CS sentences in ArzEn-ST dev set. We present accuracy, precision, recall, and F1 scores in Table 2. While these figures give us an intuition on the performance of the predictive models, it is to be noted that false positives are not necessarily incorrect. It is also to be noted that the high accuracy values are due to the high rate of true negative predictions.

As another evaluation, we check the POS distribution of the words predicted as CS by both the random and predictive models, against that of CS words in ArzEn-ST dev set. The predictive model shows a higher correlation (0.984) versus random approach (0.938). The POS distribution of the top frequent tags is shown in Appendix B. The predictions of the learnt model are dominated by nouns, followed by verbs and adjectives, where other POS tags have lower frequencies than in ArzEn-ST. The random approach gives better coverage for POS tags, however, introduces higher frequencies for low-frequent POS tags of CS words in ArzEn-ST.

CS Synthetic Data Analysis We look into how similar the synthetic data is to naturally occurring CS sentences. In Table 3, we evaluate the synthetic data in terms of the percentage of English words, the Code-Mixing Index (CMI) (Das and Gambäck, 2014), the average length of CS segments, as well as the percentage of monolingual English sentences generated. We observe that using 1-1 alignments, the generated CS sentences are close to natural occurring CS sentences in ArzEn-ST in terms of CS metrics. Using n-n alignments, the amount of CS in the synthetic data increases considerably.

[9]The MT models require around 4 hours for training. The ASR system required around 48 hours for training, as well as 6 hours for ASR rescoring. The CS predictive model using mBERT required around 10 hours for inference.

Model	Accuracy	Precision	Recall	F1
Random	77.1	18.8	21.0	0.198
Predictive	**91.9**	**76.6**	**57.4**	**0.656**

Table 2: Evaluating the performance of the predictive model on the code-switched sentences in ArzEn-ST dev set.

Model	%En (words)	CMI	av.|CS|	%En (sent.)
DICTIONARY	21.1	0.23	1.2	0.0
MAPRAND$_{1-1}$	19.9	0.22	1.14	0.0
MAPPRED$_{1-1}$	16.7	0.22	1.23	6.3
MAPRAND$_{n-n}$	27.7	0.25	2.26	6.8
MAPPRED$_{n-n}$	28.9	0.26	2.84	18.3
ArzEn-ST	18.6	0.19	1.88	3.7

Table 3: Evaluating augmented sentences in terms of CS metrics against ArzEn-ST train set.

5.2 Extrinsic Evaluation

We evaluate the improvements achieved through data augmentation on LM, ASR, MT, and ST tasks. Results are shown in Table 4. We present perplexity (PPL) for LM and Word Error Rate (WER) and Character Error Rate (CER) for ASR. For MT and ST, we use BLEU (Papineni et al., 2002), chrF, chrF++ (Popović, 2017), and BERTScore (F1) (Zhang et al., 2019). BLEU, chrF and chrF++ are calculated using SacrebleuBLEU (Post, 2018). In Table 4, we present the chrF++ scores. We present the results for all metrics in Appendix E.

Language Modeling PPL reductions are observed when using n-n over 1-1 alignments for random-based replacements. While MAPRAND$_{n-n}$ generates more data than MAPPRED$_{n-n}$, both approaches achieve similar PPL, outperforming DICTIONARY. Overall, we achieve a 34% reduction in PPL over baseline.

		LM	ASR		MT		ST	
Model	**|Train|**	**PPL$_{All}$**	**WER$_{All}$**	**CER$_{All}$**	**chrF++$_{All}$**	**chrF++$_{CS}$**	**chrF++$_{All}$**	**chrF++$_{CS}$**
Baseline		415.1	34.7	20.0	53.0	54.0	39.4	40.4
+DICTIONARY	+240,678	313.3	33.2	19.1	52.6	53.5	40.1	41.0
+MAPRAND$_{1-1}$	+240,869	306.1	32.9	19.0	55.2*	57.0*	41.0*	42.1*
+MAPPRED$_{1-1}$	+177,633	273.4	33.2	19.1	55.5†	57.4†	40.9†	42.2†
+MAPRAND$_{n-n}$	+207,026	273.8	**32.9**	**18.9**	**56.0***	**57.9***	41.4*	42.7*
+MAPPRED$_{n-n}$	+138,544	274.5	33.0	**18.9**	**56.0**†	57.8†	**41.5**†	**42.8**†
+ExtraCS		228.1	33.3	19.0	55.7	57.6	41.6	42.9
Constrained Experiments								
+c[DICTIONARY]	+99,725	324.2	33.5	19.3	52.3	53.3	39.4	40.1
+c[MAPRAND$_{n-n}$]	+99,725	293.4	33.1	19.0	55.6*	57.3*	41.2	42.6
+c[MAPPRED$_{n-n}$]	+99,725	270.4	33.0	18.9	56.0*	57.9*	41.2	42.6

Table 4: We report the results of the extrinsic tasks on ArzEn-ST test set. For language modeling, we report PPL on all sentences. For ASR, we report WER and CER on all sentences. For MT and ST, we report chrF++ on all and CS sentences. We report the results of using all augmentations (non-constrained), followed by the constrained experiments. The best performing approach in the non-constrained setting is bolded. The best performing approach in the constrained setting is underlined. We run statistical significance tests between MAPRAND and MAPPRED as well as 1-1 and n-n experiments, and mark models that are statistically significant (*p-values*< 0.05) with superscript symbols ($*, \dagger, \star$).

ASR All models utilizing augmented data outperform the baseline. The best results are achieved using MAPPRED$_{n-n}$ and MAPRAND$_{n-n}$, which perform equally well, achieving 5.2% absolute WER reduction over baseline. We observe that these models slightly outperform those trained on extra real CS data.[10]

Machine Translation Evaluation Results show that using n-n alignments outperforms 1-1 alignments on all settings. However, using a predictive model does not outperform random replacements. We observe that dictionary-based replacement negatively affects the MT systems. We also observe that our top two models perform equally well as the model utilizing real CS data, confirming the effectiveness of data augmentation, achieving 3-3.9 chrF++ points over the baseline.

MT Qualitative Analysis When looking into the translations provided by the baseline model, we observe that many CS words get dropped in translation or get mistranslated. When checking the translations provided by the MT systems trained using augmentations, we observe that the majority of the CS words are retained through translation. We also observe that these MT systems are able to retain CS OOV words, where the words are not available in the baseline training data, nor introduced in the synthetic data. This shows that by adding CS synthetic sentences to the training set, the models learn to retain English words in translation. Examples are shown in Appendix F.

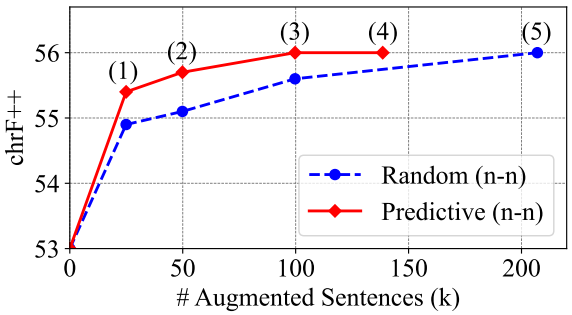

Figure 4: The chrF++ scores reported on ArzEn-ST test set when adding: (1) 25% of the sentences in the constrained experiment (=24.9k), (2) 50% of the sentences in the constrained experiment (=49.8k), (3) 100% of the sentences in the constrained experiment (=99.7k), (4) all sentences generated by MAPPRED$_{n-n}$ (=138.5k), and (5) all sentences generated by MAPRAND$_{n-n}$ (=207k).

Speech Translation Evaluation Similar to previous results, both MAPPRED and MAPRAND outperform DICTIONARY. We observe improvements for using n-n alignments over using 1-1 alignments. However, no improvements are achieved by using predictive model over random predictions.

[10] It is to be noted that the data collected from social media platforms is noisy, however, it still brings improvements in LM and ASR tasks.

Understandability	
1	No, this sentence doesn't make sense.
2	Not sure, but I can guess the meaning of this sentence.
3	Certainly, I get the meaning of this sentence.
Naturalness	
1	Unnatural, and I can't imagine people using this style of code-mixed Arabic-English.
2	Weird, but who knows, it could be some style of code-mixed Arabic-English.
3	Quite natural, but I think this style of code-mixed Arabic-English is rare.
4	Natural, and I think this style of code-mixed Arabic-English is used in real life.
5	Perfectly natural, and I think this style of code-mixed Arabic-English is very frequently used.

Table 5: The evaluation dimensions for human evaluation, following (Pratapa and Choudhury, 2021).

Constrained Experiments In order to control existing variables, such as the number of generated sentences, and how similar they are to the test set, we conduct further experiments where we restrict the augmented sentences in each approach to the CS sentences that are generated across the three techniques: DICTIONARY, MAPRAND$_{n-n}$, and MAPPRED$_{n-n}$. We report results by training our models using these restricted augmentations ($99.7k$ sentences) in addition to the baseline training data in Table 4. We find that, under this condition, for the MT task, the predictive model outperforms random, where the improvements are statistically significant on BLEU, chrF, and chrF++, as shown in Table 10. For the ASR task, while MAPPRED$_{n-n}$ achieves lower PPL over MAPRAND$_{n-n}$, both models perform equally. In Figure 4, we show the learning curves for MAPRAND$_{n-n}$ and MAPPRED$_{n-n}$ MT scores when including 25%, 50%, and 100% of the generated sentences in the constrained setting, in addition to the scores of the non-constrained setting. We see that MAPPRED$_{n-n}$ achieves overall the same performance as MAPRAND$_{n-n}$ with half the amount of generated sentences.

5.3 Human Evaluation

We perform a human evaluation study to assess the quality of sentences generated by the five models: MAPRAND$_{1-1}$, MAPPRED$_{1-1}$, MAPRAND$_{n-n}$, MAPPRED$_{n-n}$, and DICTIONARY. Out of the sentences that get augmented in all five techniques, we randomly sample 150 sentences, and ask human annotators to judge the synthetic sentences generated by each model, giving a total of 750 sentences to be evaluated.[11] We also include 150 random CS sentences from ArzEn-ST to act as control sentences. These 900 sentences were judged by three bilingual Egyptian Arabic-English speakers. Following (Pratapa and Choudhury, 2021), the sentences are evaluated against understandability and naturalness, where the rubrics are outlined in Table 5.

MOS	ArzEn	DICT	RAND (1-1)	PRED (1-1)	RAND (n-n)	PRED (n-n)
Understandability						
$1 \leq * < 2$	2.7	62.0	32.7	32.0	21.3	16.7
$2 \leq * < 3$	97.3	38.0	67.3	68.0	78.7	83.3
Naturalness						
$1 \leq * < 2$	0.7	82.7	70.7	50.0	46.7	30.0
$2 \leq * < 3$	6.0	8.7	12.7	18.0	26.0	25.3
$3 \leq * < 4$	11.3	6.0	8.0	20.0	14.0	26.0
$4 \leq * \leq 5$	82.0	2.7	8.7	12.0	13.3	18.7

Table 6: The mean opinion score (MOS) distribution for synthetic sentences, showing the percentage of sentences falling in each evaluation range.

For each synthetic/real sentence, we calculate the mean opinion score (MOS), which is the average of the three annotators' scores for that sentence. In Table 6, we present the MOS distribution for each augmentation approach, presenting the percentage of sentences falling in each evaluation range. We observe that the annotators prefer the synthetic data generated using segment replacements (n-n alignments) over those using word replacements (1-1 alignments). The annotators also prefer the synthetic data generated using trained predictive models over those using random CS point prediction. The highest scores are achieved by MAPRAND$_{n-n}$, where 44% of the synthetic sentences are perceived as natural.

[11] The sentences are sampled uniformly across the 6 corpora used in data augmentation to have equal representation of the different data sources (web/chat/conversational).

6 Discussion

In this section, we revisit our RQs:

RQ1 - Can a model learn to predict CS points using limited amount of CS data? As shown in the intrinsic evaluation, the model learns to predict CS points to some extent, as shown in the improvements in accuracy, precision, and F1 scores over random predictions. This is also observed where the POS distribution of the CS predictions using a predictive model has higher correlation to the distribution found in natural CS sentences compared to random predictions.

RQ2 - Can this information be used to generate more natural synthetic CS data? Yes, this was confirmed though human evaluation, where annotators reported higher scores for understandability and naturalness using the predictive model over using random replacements.

RQ3 - Would higher quality of synthesized CS data necessarily reflect in performance improvements in downstream tasks? In the scope of our experiments, such an entailment does not necessarily hold. We believe two limitations are affecting the performance of the predictive model. First of all, the MAPPRED approach is based on the assumption that the data provided to the predictive model is representative enough of the CS phenomenon and includes all CS patterns. Due to the scarcity of CS corpora and the dynamic behaviour of CS (El Bolock et al., 2020), this point presents a challenge and could be restricting the potential power of this model, and it could be the case that MAPRAND is able to cover more CS patterns. This is supported by the POS distribution analysis in Section 5.1. Secondly, random has the power of generating more data as opposed to using a predictive model. When we control for size, we observe improvements in MT using the predictive model. In the future, we plan to work on improving the predictive approach to generate more CS sentences. For ASR, both approaches perform equally. It was also shown in (Hussein et al., 2023) that random lexical replacement outperforms the use of Equivalence Constraint linguistic theorem for ASR. Therefore, we believe further research is needed to draw strong conclusions about the relation between the quality of generated CS data and the improvements on different downstream tasks.

7 Conclusion and Future Work

In this paper, we investigate data augmentation for CS Egyptian Arabic-English. We utilize parallel corpora to perform lexical replacements, where CS points are either selected randomly or based on predictions of a neural-based model that is trained on a limited amount of CS data. We investigate word replacements using intersection alignments as well as segment replacements using symmetrized alignments. We compare both aligned-based replacements with dictionary-based replacements. We evaluate the effectiveness of data augmentation on LM, MT, ASR, and ST tasks, as well as assess the quality through human evaluation. Across all evaluations, we report that segment replacements outperform word replacements, and aligned-based replacements outperform dictionary-based replacements. The human evaluation study shows that utilizing predictive models produces augmented data of highest quality. For the downstream tasks, random and predictive techniques achieve similar results, both outperforming dictionary-based replacements. We observe that random has the advantage of generating more data. When controlling for the amount of generated data, the predictive technique outperforms random on the MT task. Our best models achieve 34% improvement in perplexity, 5.2% relative improvement on WER for ASR task, +4.0-5.1 BLEU points on MT task, and +2.1-2.2 BLEU points on ST task.

Acknowledgements

We would like to thank Bashar Alhafni for the helpful discussions and the reviewers for their insightful comments. This project has benefited from financial support by DAAD (German Academic Exchange Service).

References

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The MGB-2 challenge: Arabic multi-dialect broadcast media recognition. In *SLT*, pages 279–284.

Ahmed Ali, Stephan Vogel, and Steve Renals. 2017. Speech recognition challenge in the wild: Arabic MGB-3. In *Proceedings of ASRU*, pages 316–322.

Ramakrishna Appicharla, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2021. IITP-MT at CALCS2021: English to Hinglish neural machine translation using unsupervised synthetic code-mixed parallel corpus. In *Proceedings of CALCS*, pages 31–35.

Mohamed Balabel, Injy Hamed, Slim Abdennadher, Ngoc Thang Vu, and Özlem Çetinoğlu. 2020. Cairo student code-switch (CSCS) corpus: An annotated Egyptian Arabic-English corpus. In *Proceedings of LREC*, pages 3973–3977.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic Dialect Corpus and Lexicon. In *Proceedings of LREC*.

Francisco Casacuberta and Enrique Vidal. 2007. Giza++: Training of statistical translation models. *Polytechnic University of Valencia, Valencia, Spain*.

Ching-Ting Chang, Shun-Po Chuang, and Hung-Yi Lee. 2018. Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In *Proceedings of Interspeech*, pages 554–558.

Song Chen, Dana Fore, Stephanie Strassel, Haejoong Lee, and Jonathan Wright. 2017. BOLT Egyptian Arabic SMS/Chat and Transliteration LDC2017T07. Philadelphia: Linguistic Data Consortium.

Song Chen, Jennifer Tracey, Christopher Walker, and Stephanie Strassel. 2019. BOLT Arabic discussion forum parallel training data. Linguistic Data Consortium (LDC) catalog number LDC2019T01, ISBN 1-58563-871-4.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387.

Pavel Denisov, Manuel Mager, and Ngoc Thang Vu. 2021. IMS' systems for the first IWSLT 2021 low-resource speech translation task. In *Proceedings of IWSLT*.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic treebanks and associated corpora: Data divisions manual. *arXiv preprint arXiv:1309.5652*.

Alia El Bolock, Injy Khairy, Yomna Abdelrahman, Ngoc Thang Vu, Cornelia Herbert, and Slim Abdennadher. 2020. Who, when and why: The 3 Ws of code-switching. In *Proceedings of Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness*, pages 83–94.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. CALLHOME Egyptian Arabic transcripts – LDC97T19. Web Download. Philadelphia: Linguistic Data Consortium.

Marwa Gaser, Manuel Mager, Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2022. Exploring segmentation approaches for neural machine translation of code-switched Egyptian Arabic-English text. In *Proceedings of EACL*.

Abhirut Gupta, Aditya Vavre, and Sunita Sarawagi. 2021. Training data augmentation for code-mixed translation. In *Proceedings of NAACL-HLT*, pages 5760–5766.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of EMNLP-IJCNLP*, pages 6098–6111.

Nizar Habash and Bonnie Dorr. 2003. CatVar: A database of categorial variations for English. In *Proceedings of Machine Translation Summit IX: System Presentations*.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22.

Injy Hamed, Pavel Denisov, Chia-Yu Li, Mohamed Elmahdy, Slim Abdennadher, and Ngoc Thang Vu. 2022a. Investigations on speech recognition systems for low-resource dialectal Arabic–English code-switching speech. *Computer Speech & Language*, 72:101278.

Injy Hamed, Mohamed Elmahdy, and Slim Abdennadher. 2018. Collection and analysis of code-switch Egyptian Arabic-English speech corpus. In *Proceedings of LREC*.

Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2022b. ArzEn-ST: A three-way speech translation corpus for code-switched Egyptian Arabic-English. In *Proceedings of WANLP*.

Injy Hamed, Ngoc Thang Vu, and Slim Abdennadher. 2020. ArzEn: A speech corpus for code-switched Egyptian Arabic-English. In *Proceedings of LREC*, pages 4237–4246.

Injy Hamed, Moritz Zhu, Mohamed Elmahdy, Slim Abdennadher, and Ngoc Thang Vu. 2019. Code-switching language modeling with bilingual word embeddings: A case study for Egyptian Arabic-English. In *Proceedings of SPECOM*, pages 160–170.

Amir Hussein, Shammur Absar Chowdhury, Ahmed Abdelali, Najim Dehak, Ahmed Ali, and Sanjeev Khudanpur. 2023. Textual data augmentation for Arabic-English code-switching speech recognition. In *Proceedings of SLT*, pages 777–784.

Lars Kjeldgaard and Lukas Nielsen. 2021. NERDA. GitHub.

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 *Proceedings of ACL*, pages 177–180.

Gaurav Kumar, Yuan Cao, Ryan Cotterell, Chris Callison-Burch, Daniel Povey, and Sanjeev Khudan-

pur. 2014. Translations of the CALLHOME Egyptian Arabic corpus for conversational speech translation. In *Proceedings of IWSLT*.

Garry Kuwanto, Afra Feyza Akyürek, Isidora Chara Tourni, Siyang Li, and Derry Wijaya. 2021a. Low-resource machine translation for low-resource languages: Leveraging comparable data, code-switching and compute resources. *CoRR*, abs/2103.13272.

Garry Kuwanto, Afra Feyza Akyürek, Isidora Chara Tourni, Siyang Li, Alexander Gregory Jones, and Derry Wijaya. 2021b. Low-resource machine translation training curriculum fit for low-resource languages. *arXiv preprint arXiv:2103.13272*.

LDC. 2002a. 1997 HUB5 Arabic transcripts – LDC2002T39. Web Download. Philadelphia: Linguistic Data Consortium.

LDC. 2002b. CALLHOME Egyptian Arabic transcripts supplement – LDC2002T38. Web Download. Philadelphia: Linguistic Data Consortium.

Grandee Lee, Xianghu Yue, and Haizhou Li. 2019. Linguistically motivated parallel data augmentation for code-switch language modeling. In *Proceedings of Interspeech*, pages 3730–3734.

Chia-Yu Li and Ngoc Thang Vu. 2020. Improving code-switching language modeling with artificially generated texts using cycle-consistent adversarial networks. In *Proceedings of Interspeech*, pages 1057–1061.

Mohamed Amine Menacer, David Langlois, Denis Jouvet, Dominique Fohr, Odile Mella, and Kamel Smaïli. 2019. Machine translation on a parallel code-switched corpus. In *Proceedings of the Canadian Conference on Artificial Intelligence*, pages 426–432.

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 (Demonstrations)*, pages 48–53.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *Proceesings of ICASSP*, pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of Interspeech*, pages 2613–2617.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of LREC*, pages 1094–1101.

Shana Poplack. 1980. Sometimes i'll start a sentence in Spanish y termino en Español: Toward a typology of code-switching. *The bilingualism reader*, 18(2):221–256.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

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.

Adithya Pratapa, Gayatri Bhat, Monojit Choudhury, Sunayana Sitaram, Sandipan Dandapat, and Kalika Bali. 2018. Language modeling for code-mixing: The role of linguistic theory based synthetic data. In *Proceedings of ACL*, pages 1543–1553.

Adithya Pratapa and Monojit Choudhury. 2021. Comparing grammatical theories of code-mixing. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 158–167.

Mohd Sanad Zaki Rizvi, Anirudh Srinivasan, Tanuja Ganu, Monojit Choudhury, and Sunayana Sitaram. 2021. GCM: A toolkit for generating synthetic code-mixed text. In *Proceedings of EACL (System Demonstrations)*, pages 205–211.

Caroline Sabty, Islam Omar, Fady Wasfalla, Mohamed Islam, and Slim Abdennadher. 2021. Data augmentation techniques on Arabic data for named entity recognition. *Procedia Computer Science*, 189:292–299.

Ali Shazal, Aiza Usman, and Nizar Habash. 2020. A unified model for Arabizi detection and transliteration using sequence-to-sequence models. In *Proceedings of the Arabic Natural Language Processing Workshop*, pages 167–177.

Han-Ping Shen, Chung-Hsien Wu, Yan-Ting Yang, and Chun-Shan Hsu. 2011. CECOS: A Chinese-english code-switching speech database. In *2011 International Conference on Speech Database and Assessments (Oriental COCOSDA)*, pages 120–123.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of ACL-IJCNLP*, pages 3154–3169.

Jennifer Tracey et al. 2021. BOLT Egyptian Arabic sms/chat parallel training data LDC2021T15. Web Download. Philadelphia: Linguistic Data Consortium.

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

Ngoc Thang Vu, Dau-Cheng Lyu, Jochen Weiner, Dominic Telaar, Tim Schlippe, Fabian Blaicher, Eng-Siong Chng, Tanja Schultz, and Haizhou Li. 2012. A

first speech recognition system for Mandarin-English code-switch conversational speech. In *Proceedings of ICASSP*, pages 4889–4892.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. 2018. Espnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2207.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2018. Learn to code-switch: Data augmentation using copy mechanism on language modeling. *CoRR*, abs/1810.10254.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of CoNLL*, pages 271–280.

Jitao Xu and François Yvon. 2021. Can you traducir this? machine translation for code-switched input. *arXiv preprint arXiv:2105.04846*.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris CallisonBurch. 2012. Machine translation of Arabic dialects. In *Proceedings of NAACL*, pages 49–59.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating text generation with bert. In *Proceedings of the International Conference on Learning Representations*.

Limitations

To the best of our knowledge, this paper presents the first comparison for the mentioned lexical replacement techniques, covering human evaluation as well as three downstream tasks; automatic speech recognition, machine translation, and speech translation. However, the study is focused on the Egyptian Arabic-English language pair, and we make no assumptions on the generalizability of results to other language pairs, nor other domains. Further investigations are needed to assess how the results would differ, especially in the case of languages with less syntactic divergence. We also note another limitation in the human evaluation, which is that code-switching is a user-dependent behaviour, that differs across different users, and thus the evaluation of the naturalness of a code-switched sentence is very subjective. We have taken this into account in our human evaluation study by having each sentence evaluated by three annotators and taking the average across the three ratings.

Ethics Statement

We could not identify potential harm from using the provided models in this work. However, one concern is that code-switched ST is yet a challenging task, and the ST models trained in this work provide low performance, and thus should not be deployed as it can mislead the users.

A ArzEn-ST Corpus

In Table 7, we provide an overview on ArzEn-ST corpus. In Table 8, we show examples from the corpus.

ArzEn-ST Speech Corpus	
Duration	12h
#Speakers	40
# Sentences	6,216
% CS sentences	63.7%
% Arabic sentences	33.2%
% English sentences	3.1%

Table 7: ArzEn-ST corpus overview.

#	**Example**
1	←انا كتبت ال project code *AnA ktbt Al* project code I wrote the project code
2	←عملت كذا internship *Emlt k*A* internship I did several internships
3	←كنت ب overload الناس اللى معايا *knt b* overload *AlnAs Ally mEAyA* I was overloading my teammates
4	←ن detect ال traffic within period معينة *n* detect *Al* traffic within period *mEynp* to detect the traffic within a certain period

Table 8: ArzEn-ST corpus examples, showing source text, its transliteration (Habash et al., 2007), and translation. The arrows beside the sentences show the sentence starting direction, as Arabic is read right to left.

B POS Intrinsic Evaluation

As an intrinsic evaluation of the CS predictive model, we check the POS distribution of the words predicted as CS words by both the random and predictive approaches, against that of CS words in ArzEn-ST dev set. We report that the natural POS distribution is in-line with the distributions reported

POS	ArzEn	Random	Predictive
NN	48.4	33.2	67.0
VB	14.5	22.9	13.6
JJ	13.1	9.3	13.6
RB	7.6	6.5	1.7
IN	5.0	8.9	0.9
PRP	3.8	4.7	0.6
DT	2.2	3.7	0.2
CC	0.9	3.8	0.1
Total	94.7	89.3	97.6

Table 9: The POS distribution (%) of the words predicted as CS words by both the random and predictive models, against that of CS words in ArzEn-ST dev set.

for CS Egyptian Arabic-English (Hamed et al., 2018; Balabel et al., 2020), where the dominating POS tags are nouns, verbs, and adjectives, followed by adverbs, pronouns, and prepositions. We report that the predictive model gives a higher correlation (0.984) versus random approach (0.938). We present the POS distribution of the top frequent tags in Table 9. We observe that the predictive model provides a percentage of nouns that is significantly higher than that occurring in ArzEn-ST. It also provides less coverage to the tags occurring less frequently in ArzEn-ST. We believe this can be due to the predictive model being trained on limited data. The random approach on the other hand, provides higher counts for less frequent POS tags, as seen in the total, where 11% of the words identified by the random prediction to be code-switched belong to POS tags that are infrequent in natural CS data.

C Data Preprocessing

Data preprocessing involved removing corpus-specific annotations, removing URLs and emoticons through *tweet-preprocessor*,[12] tokenizing numbers, lowercasing, running Moses' (Koehn et al., 2007) tokenizer as well as MADAMIRA (Pasha et al., 2014) simple tokenization (D0), and performing Alef/Ya normalization. For LDC2017T07 (Chen et al., 2017), LDC2019T01 (Chen et al., 2019), and LDC2021T15 (Tracey et al., 2021), some words have literal and intended translations. We opt for one translation having all literal translations and another having all intended translations. For LDC2017T07, we utilize the work by Shazal et al. (2020), where the authors used a sequence-to-sequence deep learning model to transliterate SMS/chat text in LDC2017T07 from Arabizi (where Arabic words are written in Roman script) to Arabic orthography.

[12] https://pypi.org/project/tweet-preprocessor/

D MT Hyperparameters

The following is the train command:
python3 fairseq_cli/train.py $DATA_DIR –source-lang src –target-lang tgt –arch transformer –share-all-embeddings –encoder-layers 5 –decoder-layers 5 –encoder-embed-dim 512 –decoder-embed-dim 512 –encoder-ffn-embed-dim 2048 –decoder-ffn-embed-dim 2048 –encoder-attention-heads 2 –decoder-attention-heads 2 –encoder-normalize-before –decoder-normalize-before –dropout 0.4 –attention-dropout 0.2 –relu-dropout 0.2 –weight-decay 0.0001 –label-smoothing 0.2 –criterion label_smoothed_cross_entropy –optimizer adam –adam-betas '(0.9, 0.98)' –clip-norm 0 –lr-scheduler inverse_sqrt –warmup-updates 4000 –warmup-init-lr 1e-7 –lr 1e-3 –stop-min-lr 1e-9 –max-tokens 4000 –update-freq 4 –max-epoch 100 –save-interval 10 –ddp-backend=no_c10d

E MT Results

In Table 10, we present the MT and ST results of the non-constrained and constrained experiments. We report the scores on BLEU, chrF, chrF++, and BERTScore(F1). Given that each metric has its strengths and weaknesses, we also report the average of the four metrics ($AvgMT$).

F Translation Examples

In Table 11, we show examples of source-target pairs with their translations obtained from different MT models. We observe that the models trained using augmented sentences are better than the baseline MT model at retaining CS words in the source sentence in the translations.

	All Sentences					CS Sentences				
Model	**BLEU**	**chrF**	**chrF++**	**F_{BERT}**	**Avg_{MT}**	**BLEU**	**chrF**	**chrF++**	**F_{BERT}**	**Avg_{MT}**
Non-constrained Experiments										
MT										
Baseline	31.0	54.2	53.0	0.519	47.5	31.4	55.3	54.0	0.501	47.7
+DICTIONARY	30.9	53.8	52.6	0.516	47.2	31.5	54.7	53.5	0.498	47.4
+MAPRAND$_{1-1}$	34.4‡	56.6*	55.2*	0.545	50.2	35.9‡	58.5*,‡	57.0*	0.543	51.4
+MAPPRED$_{1-1}$	33.7‡,†	56.9†	55.5†	0.548	50.2	35.2‡,†	58.9†,‡	57.4†	0.549	51.6
+MAPRAND$_{n-n}$	34.7	57.2*	**56.0***	**0.552**	**50.8**	36.2	**59.2***	**57.9***	**0.552**	**52.1**
+MAPPRED$_{n-n}$	**35.0**†	**57.3**†	**56.0**†	0.550	**50.8**	**36.5**†	**59.2**†	57.8†	**0.552**	**52.1**
+ExtraCS	34.8	57.2	55.7	0.547	50.6	36.2	59.1	57.6	0.546	51.9
ST										
Baseline	15.3	41.2	39.4	0.335	32.4	15.8	42.4	40.4	0.317324	32.6
+DICTIONARY	16.3	41.9	40.1	0.344	33.2	16.8	42.8	41.0	0.324	33.2
+MAPRAND$_{1-1}$	16.5‡,*	42.8*	41.0*	0.347	33.8	17.0*	44.1*	42.1*	0.329	34.0
+MAPPRED$_{1-1}$	16.1‡,†	42.8†	40.9†	0.348	33.6	16.9†	44.2†	42.2†	0.331	34.1
+MAPRAND$_{n-n}$	**17.0***	43.3*	41.4*	0.349	**34.2**	**17.7***	44.7*	42.7*	0.332	**34.6**
+MAPPRED$_{n-n}$	16.9†	**43.4**†	**41.5**†	**0.352**	**34.2**	17.4†	**44.8**†	**42.8**†	**0.335**	**34.6**
+ExtraCS	17.4	43.4	41.6	0.353	34.4	18.0	44.7	42.9	0.336	34.8
Constrained Experiments										
MT										
+c[DICTIONARY]	30.3	53.6	52.3	0.517	47.0	31.0	54.6	53.3	0.499	47.2
+c[MAPRAND$_{n-n}$]	33.8*	56.9*	55.6*	0.553	50.4	35.1*	58.7*	57.3*	0.555	51.7
+c[MAPPRED$_{n-n}$]	35.0*	57.4*	56.0*	0.551	50.9	36.8*	59.5*	57.9*	0.554	52.4
ST										
+c[DICTIONARY]	15.2	41.2	39.4	0.341	32.5	15.5	42.1	40.1	0.319	32.4
+c[MAPRAND$_{n-n}$]	16.4	43.1	41.2	0.350	33.9	17.0	44.7	42.6	0.335	34.5
+c[MAPPRED$_{n-n}$]	16.6	43.1	41.2	0.353	34.0	17.2	44.6	42.6	0.337	34.5

Table 10: MT and ST evaluation on ArzEn-ST test set for the non-constrained (using all augmentations) and constrained experiments. We report BLEU, chrF, chrF++, F1 BERTScore (F_{BERT}), and their average (Avg_{MT}), on all sentences as well as code-switched sentences only. The best performing data augmentation approach in the non-constrained setting is bolded. The best performing approach in the constrained setting is underlined. We run statistical significance tests between pairs of models to compare the effect of using MAPRAND vs. MAPPRED and 1-1 vs. n-n alignments, and mark models that are statistically significant (*p-values*< 0.05) with superscript symbols ($*, \dagger, \ddagger, \star$).

Model	Example
Src	ما هو المفروض ال . . ال . . الناس اللي بت adjudicate يبقوا poker face فهو ماينفعش يفهمني اي حاجة بس بعديها بيبقي يعني بعرف غلطي ، بس
Tgt-Ref	those one who adjudicate should have a poker face, so i can't get any signal from them, but afterwards i know my mistake, that's all
Baseline	it's supposed to be the. the. people who are a rijudi could be powder face so it can't explain anything but after that i mean i know my mistake, that's it
DICTIONARY	the.. the. the.. the people who are hurt should be thinking about face, so it can't explain anything to me, but after that, i mean, i know my mistake, that's it
MAPRAND$_{1-1}$	the.. the.. the.. the.. the people who adjudicate become a poker of face, so he can't explain anything to me, but after that i know my mistake, that's it
MAPPRED$_{1-1}$	the.. the. the.. the people that adjudicate become the poker face, so he can't understand anything but after that i mean i know my mistake, that's it
MAPRAND$_{n-n}$	the.. the.. the.. the people who are adjudicate, they become poker face, so it can't explain anything to me after that, i mean, i know my mistake, that's it
MAPPRED$_{n-n}$	the.. the. the.. the people who are adjudicate should be poker face, so he can't explain anything to me but after that, i mean, i know my mistake, that's it
Src	انا بعمل مشروع اسمه multi-robot system task allocation
Tgt-Ref	i'm working on a project called multi-robot system task allocation.
Baseline	i make a project called multi-robot system and allocation
DICTIONARY	i'm making a project called al-gamalt system for the task of allocation
MAPRAND$_{1-1}$	i make a project called multi-robot system and allocation
MAPPRED$_{1-1}$	i am making a project called multi-robot system and allocation task
MAPRAND$_{n-n}$	i am making a project called multi-robot system allocation
MAPPRED$_{n-n}$	i am doing a project called multi-robot system task allocation

Table 11: Examples of translation outputs obtained from the MT models. The words in the translations that correspond to the CS words in the input source sentence are underlined.