Exploring Segmentation Approaches for Neural Machine Translation of Code-Switched Egyptian Arabic-English Text

Marwa Gaser,[1] **Manuel Mager,**[2*] **Injy Hamed,**[3,4]
Nizar Habash,[4] **Slim Abdennadher,**[1] **Ngoc Thang Vu**[3]
[1]The German University in Cairo, [2]AWS AI Labs
[3]University of Stuttgart, [4]New York University Abu Dhabi
{marwa.saleh,slim.abdennadher}@guc.edu.eg, magerlm@amazon.com,
{injy.hamed,nizar.habash}@nyu.edu, thang.vu@ims.uni-stuttgart.de

Abstract

Data sparsity is one of the main challenges posed by code-switching (CS), which is further exacerbated in the case of morphologically rich languages. For the task of machine translation (MT), morphological segmentation has proven successful in alleviating data sparsity in monolingual contexts; however, it has not been investigated for CS settings. In this paper, we study the effectiveness of different segmentation approaches on MT performance, covering morphology-based and frequency-based segmentation techniques. We experiment on MT from code-switched Arabic-English to English. We provide detailed analysis, examining a variety of conditions, such as data size and sentences with different degrees of CS. Empirical results show that morphology-aware segmenters perform the best in segmentation tasks but under-perform in MT. Nevertheless, we find that the choice of the segmentation setup to use for MT is highly dependent on the data size. For extreme low-resource scenarios, a combination of frequency and morphology-based segmentations is shown to perform the best. For more resourced settings, such a combination does not bring significant improvements over the use of frequency-based segmentation.

1 Introduction

Code-switching (CS), i.e. the alternation of language in text or speech, has been gaining worldwide popularity, due to several reasons, including globalization and immigration. While this has been met with a growing interest in the NLP field to build systems that can handle such mixed input, work on CS machine translation (MT) is still considered in its infancy, where only a few language pairs have been investigated (Sinha and Thakur, 2005; Dhar et al., 2018; Menacer et al., 2019; Xu and Yvon, 2021; Hamed et al., 2022c).

CS Sentence	it depends بصراحة بالنسبالي ع ال situation it depends *bSrAHp bAlnsbAly E Al* situation
Translation	for me it honestly depends on the situation

Word	**Translation**	**Segmentation**
it	it	it
depends	depends	depend#s
بصراحة *bSrAHp*	honestly	ب#صراحة *b#SrAHp*
بالنسبالي *bAlnsbAly*	for me	ب#النسبا#ل#ي *b#AlnsbA#ly*
ع *E*	on	ع *E*
ال *Al*	the	ال *Al*
situation	situation	situation

Figure 1: An example sentence with code-switching (CS) between English and Egyptian Arabic. The words are contrasted with their segmentations and English translations. Arabic words are paired with their transliterations in the Buckwalter scheme (Habash et al., 2007).

In this work, we focus on the CS Egyptian Arabic (EGY)-English (EN) language pair, as we observe its usage is becoming more common. Besides being prevalent amongst Egyptian migrant communities, it is also commonly used in Egypt due to the increase in international schooling systems and educational advancements. We identify three main challenges for CS MT. First is **data sparsity**, a challenge common to many CS language pairs because of limited parallel corpora containing commissioned translations of CS text (Çetinoğlu et al., 2016; Srivastava and Singh, 2020; Tarunesh et al., 2021; Hamed et al., 2022b; Chen et al., 2022). Second is Egyptian Arabic **morphological richness**, which further exacerbates the data sparsity situation (Habash et al., 2012a,b). Third, since the matrix language (EGY) is morphologically rich, CS occurs at three **CS levels**: on the boundaries of sentences (inter-sentential CS), between words (intra-sentential CS), and within words, i.e., morphological code-switching (MCS). This mix of types of CS raises the question of how to handle them all in the same system. These challenges are further illustrated in Figure 1.

*Work done while at the University of Stuttgart.

Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 3523–3538

A common solution to handle data sparsity for MT of morphologically rich languages is morphological segmentation (Oudah et al., 2019; Ataman et al., 2017; Grönroos et al., 2020). However, this has not been investigated for CS. In this paper, we explore a wide range of segmentation approaches, covering unsupervised morphology-based segmenters, unsupervised frequency-based segmenters, and supervised morphology-based segmenters. This work aims to answer the following research questions (RQs):

- **RQ1:** Which segmentation setup performs the best in the downstream MT task across different training sizes?
- **RQ2:** Does the effectiveness of the different segmenters in the MT task differ according to the CS type of the source sentence?
- **RQ3:** Is there a correlation between a more morphologically correct segmentation and MT performance?

While our results show that there is no correlation between correct morphological segmentation and MT performance, we find that the performance ranking between the MT systems varies across different training data sizes and sentence types (monolingual vs. code-switched). We show that applying a combination of supervised morphology-based and unsupervised frequency-based segmentations consistently gives best results, with statistical significance under low data sizes. While common-wisdom suggests that Byte-Pair Encoding (BPE) is the best approach, our experiments highlight the importance of integrating morphological knowledge in the case of extreme low-resource settings. We believe that the insights and methodology we follow will be useful to researchers working with low-resource languages. An additional contribution of our research is the creation of a gold standard morphologically annotated CS Egyptian Arabic-English dataset which we make publicly available.[1]

The paper is organized as follows. Section 2 discusses related work. Section 3 describes the dataset we annotated. Section 4 describes and evaluates the different segmenters used. Section 5 describes and evaluates the various MT systems. In Section 6, we answer our research questions.

[1] http://arzen.camel-lab.com/

2 Related Work

Several researchers have investigated the effect of applying different morphological and agnostic segmentation approaches on the MT performance for monolingual languages. Roest et al. (2020); Saleva and Lignos (2021) show that unsupervised morphology-based segmentation like Linguistically Motivated Vocabulary Reduction (LMVR) (Ataman et al., 2017), Morfessor (Smit et al., 2014), and FlatCat (Grönroos et al., 2014) for Nepali–English, Sinhala–English, Kazakh–English, and Inuktitut–English language pairs show either no improvement or no significant improvement over the agnostic BPE segmentation (Sennrich et al., 2016) in translation tasks. Meanwhile, Mager et al. (2022) and Ataman et al. (2017) show that for polysynthetic and highly agglutinative languages, unsupervised morphology-based segmentation outperforms BPEs (Sennrich et al., 2016) in MT tasks in both directions. Nevertheless, applying BPEs on top of morphology-based segmentation for Turkish–English, Uyghur–Chinese, and Arabic–English has shown to bring improvements over solely using BPEs or morphology-based segmentation for neural MT task (Pan et al., 2020; Tawfik et al., 2019). A similar result was achieved by (Ortega et al., 2020), using a morphological guided BPE for polysynthetic languages. However, Oudah et al. (2019) show that such an approach is beneficial in the case of statistical machine translation (SMT), and does not improve results for neural machine translation (NMT). For other natural language processing (NLP) tasks, Al-Thubaity and Al-Subaie (2015) show that utilizing word segmented Arabic dataset leads to improvements in text classification task over utilizing unsegmented dataset in terms of accuracy, precision, recall, and F-measure.

As for work on CS MT, there are many efforts (Sinha and Thakur, 2005; Dhar et al., 2018; Mahata et al., 2019; Menacer et al., 2019; Song et al., 2019; Tarunesh et al., 2021; Xu and Yvon, 2021; Chen et al., 2022; Hamed et al., 2022c). To the best of our knowledge, none of these efforts presented an extensive comparison covering different segmentation techniques. With regards to the languages covered, only Menacer et al. (2019) worked on CS Arabic-English. However, since they used carefully edited UN documents, the text only included the Modern Standard Arabic variety, and contained limited types of CS.

With regards to similar corpora, Balabel et al.

Case	Stem	Ending	Example
Irregular	modified	Irregular: es	monki+es
Irregular	modified	Regular: s,ed,ing,en	car+ing
Irregular	modified	Irregular: <nil>	went
Irregular	unmodified	Irregular: es	church+es
Regular	unmodified	Regular: s,ed,ing,en	car+s

Table 1: The ordered list of rules we follow to segment the English words.

	EGY	EN
Test Words	3,414	501
Dev Words	3,069	567
Total Words	6,483	1,068
Total Segmented Words	1,206	146
Total Morphs	7,911	1,214
Total Unique Morphs	1,192	432
% of Total Segmented Words	18.6%	13.7%
Morphs/Word	1.220	1.137
Maximum Morphs per Word	5	2

Table 2: Statistics on *ArzEnSEG* corpus.

(2020) annotated CS Egyptian Arabic-English data (Hamed et al., 2018) with tokenization (canonical segmentation), lemmatization, and POS tags. However, their corpus does not contain translations.

3 Data

3.1 Pre-existing Datasets

We use the *ArzEn* parallel corpus (Hamed et al., 2020, 2022b), which consists of speech transcriptions gathered through informal interviews with bilingual Egyptian Arabic-English speakers, as well as their English translations. The corpus consists of 6,213 sentences, where 4,154 (66.9%) are code-mixed, 1,865 (30.0%) are monolingual Arabic, and 194 (3.1%) are monolingual English. Among the code-mixed sentences, there are 1,781 (28.7%) sentences with morphological code-switching. We follow the predefined dataset splits, containing 3,341 (53.8%), 1,402 (22.6%), and 1,470 (23.7%) sentences for train, dev, and test sets, respectively. For training purposes, we also use 308k monolingual parallel sentences obtained from MADAR (Bouamor et al., 2018) and the following *LDC* corpora: (Gadalla et al., 1997; LDC, 2002b,a; Chen et al., 2017; Tracey et al., 2021; BBN Technologies et al., 2012; Chen et al., 2019). The preprocessing steps we apply are outlined in Appendix A. We use *ArzEn* train set as well as the monolingual parallel corpora to train both the segmenters and MT systems. For tuning and testing the MT systems, we use the *ArzEn* dev and test sets. For tuning and testing the segmenters, we annotated a new dataset, discussed next.

3.2 A New Dataset: ArzEn Surface Segmentation (ArzEnSEG) Corpus

To facilitate our research, we created a code-switched Egyptian Arabic-English morphologically annotated dataset which we use for tuning and testing. The dataset comprises the first 500 lines of *ArzEn* dev set. Unlike Balabel et al. (2020), we opt for surface form segmentation to allow for evaluating the segmenters. We also opt for extending *ArzEn* dataset as it contains translations and is used in our MT experiments.

For Arabic word segmentation, we use the Arabic Treebank (ATB) segmentation scheme (Maamouri et al., 2004; Habash, 2010). We choose this scheme as it is the standard tokenization scheme used in different treebanks (Maamouri et al., 2004, 2012; Taji et al., 2017; Habash et al., 2022). It has also shown to be efficient in Oudah et al. (2019) and has demonstrated its competitiveness in Habash et al. (2013).

For English word segmentation, we follow five rules in sequential order depending on whether the word has a regular or irregular stem and whether the word has a regular or irregular ending. Table 1 exhibits the five English rules we follow in order.

All annotation decisions were made in context by two bilingual speakers who collaborated on initial annotations and quality checks. Figure 1 presents an annotation example. We divide the sentences randomly into dev and test sets (250 sentences each). In Table 2, we display statistics about *ArzEnSEG*.

4 Segmentation Experiments

4.1 Experimental Setup

We explore three categories of segmenters: unsupervised morphology-based, unsupervised frequency-based, and supervised morphology-based segmentation. For the unsupervised morphology-based segmenters, we use MorphA-Gram in addition to three segmenters from the Morfessor family: Morfessor, LMVR, and FlatCat. For unsupervised frequency-based segmenters, we use BPE. Figure 2 summarizes the process of training these segmenters. For the supervised morphology-based segmenters, we use MADAMIRA (Pasha et al., 2014), where we exploit the segmentation schemes designed for Egyptian Arabic.

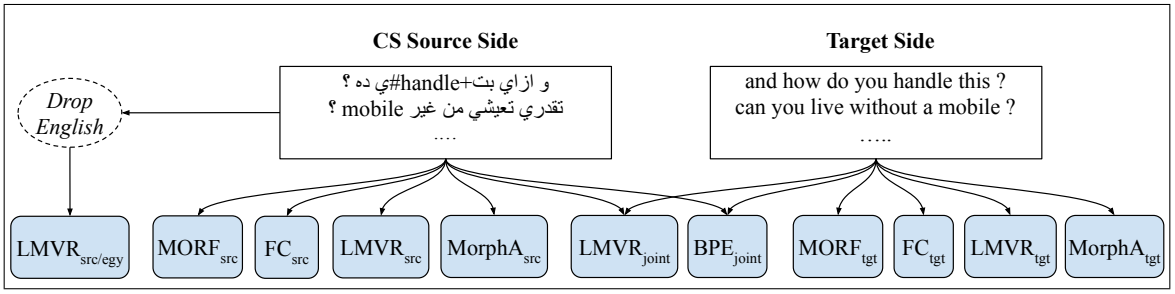


Figure 2: The unsupervised segmentation models we study in this paper and their training data dependencies. We use four systems: Morfessor (MORF), FlatCat (FC), LMVR, and MorphAGram (MorphA). The subscripts specify the training data: source (src), target (tgt), source+target (joint), and source without English, i.e., Egyptian, (src/egy).

4.2 Segmentation Systems

In this section, we introduce the segmentation systems used for the study. Details about the hyperparameter tuning for each system family can be found in Appendix B. The different segmentation models and their training dataset are displayed in Figure 2.

Morfessor Family We exploit three Morfessor family tools for unsupervised morphology-based segmentation in this research: Morfessor, (Smit et al., 2014), FlatCat (Grönroos et al., 2014), and LMVR (Ataman et al., 2017).

Morfessor is a morphological-based segmentation model which we train in an unsupervised manner. Three components form the system: the model, the cost function, and the training and decoding algorithms (Virpioja et al., 2013). The model is mainly concerned with the grammar and lexicon where the latter holds the attributes of the subwords and the grammar controls how these subwords are combined to form the word. Morfessor's grammar assumes that the subwords that form the word are independent of each other and that a word has at least one subword.

FlatCat is a variant of Morfessor which we also train in an unsupervised manner. Even though FlatCat builds on Morfessor and shares the same model component, they differ in their morphotactics (the set of rules that determine how the word's morphemes are arranged). FlatCat morphotactics is based on the Hidden Markov model (Baum and Petrie, 1966) which considers context. On the contrary, Morfessor's morphotactics algorithm is based on a unigram model which is not context-sensitive.

LMVR is a morphology-based segmenter that is built upon FlatCat and we train in an unsupervised manner. Nonetheless, LMVR takes into consideration the desired segmentation output vocabulary size during training.

For each tool, two models are generated; one trained on the source side; thus capable of segmenting CS data, and the other trained on the target side of the training data; thus capable of segmenting English data only. We add a *src* and *tgt* subscript to the segmenters' names to distinguish between both settings. Hence, MORF$_{src}$, FC$_{src}$, and LMVR$_{src}$ resemble Morfessor, FlatCat, and LMVR respectively, where the segmenters are trained on the source side. MORF$_{tgt}$, FC$_{tgt}$, and LMVR$_{tgt}$ resemble the segmenters trained on the target side.

MorphAGram We also include in this study the unsupervised morphology segmenter MorphAGram (Eskander et al., 2020) which is based on Adaptor Grammars. We use the *PrStSu+SM* grammar, which represents a word as a sequence of prefixes followed by a stem then a sequence of suffixes, in the unsupervised *Standard* learning setting to train the segmenters.

BPE The SentencePiece (Kudo and Richardson, 2018) implementation of BPE (Gage, 1994; Sennrich et al., 2016) is a frequency-based unsupervised segmenter. We train the BPE model jointly, on the concatenation of the source and target sides of the training parallel corpus. Previous work has shown that this approach is better suited for low resource settings (Guzmán et al., 2019). We refer to our joint BPE segmenter as BPE$_{joint}$.

MADAMIRA For supervised morphology-based segmenters, we use MADAMIRA's Egyptian Arabic model (Pasha et al., 2014), which was trained on the Egyptian Arabic Treebank (parts 1 through 6) (Maamouri et al., 2012). Specifically, we use MADAMIRA's *ATB_BWFORM* and *D3_BWFORM* schemes, henceforth MDMR$_{ATB}$ and MDMR$_{D3}$, respectively. Both schemes apply Alif/Ya normaliza-

Segmenter	EMMA F1 Score		
	EGY	EN	All
raw	0.806	0.953	0.838
MorphA$_{src}$	0.682	0.942	0.737
MORF$_{src}$	0.814	0.888	0.832
FC$_{src}$	0.821	0.961	0.851
LMVR$_{src}$	0.836	0.961	0.863
LMVR$_{src/egy}$	0.838	0.953	0.863
MorphA$_{tgt}$	0.806	0.953	0.838
MORF$_{tgt}$	0.147	0.951	0.327
FC$_{tgt}$	0.806	0.952	0.838
LMVR$_{tgt}$	0.806	0.966	0.842
LMVR$_{joint}$	0.841	0.963	**0.868**
BPE$_{joint}$	0.678	0.814	**0.707**
MDMR$_{ATB}$	0.935	0.953	**0.939**
MDMR$_{D3}$	0.868	0.953	0.887

Table 3: EMMA F1 score calculated on *ArzEnSEG* test set for the raw data as well as the segmented data using the different segmenters. The Arabic gold segmentation is based on the ATB segmentation scheme. We show the overall score (All) and language-specific scores calculated on the Egyptian Arabic (EGY) and English (EN) words separately. Segmenter names with a *src*, *tgt*, and *joint* subscripts represent segmenters that are trained on the source, target, and source+target sides respectively. The best performing segmenters from each category are highlighted in bold.

tion and segment the Arabic clitics. MDMR$_{D3}$ splits the Arabic definite article ال *Al* (the), while MDMR$_{ATB}$ does not.

4.3 Segmentation Results

To evaluate the performance of the segmenters, we use EMMA F1 score (Spiegler and Monson, 2010). Results in Table 3, reported on *ArzEnSEG* test set, show overall and language-specific scores.

Unsupervised morphology-based segmentation Results show that LMVR outperforms the other unsupervised morphology-based segmenters in terms of segmenting Arabic and English words. We perform further experiments where we train 2 additional models: i) a model trained jointly on the concatenation of the source and target sides of the parallel corpus, and ii) a model trained on the Arabic words only in the source side (where English words are dropped). Therefore, the former model is capable of segmenting both languages, while the latter is only tailored for segmenting Arabic words. We perform these experiments using LMVR, given that it outperforms the other segmenters. We refer to these models as LMVR$_{joint}$ and LMVR$_{src/egy}$ respectively, as outlined in Figure 2. Results show that joint training provides best EMMA scores.

Supervised morphology-based segmentation As shown in Table 3, both supervised morphology-based segmenters MDMR$_{ATB}$ and MDMR$_{D3}$ outperform all other segmenters. Their superiority in segmenting Arabic is expected, as they are trained on human-annotated data and hence are capable of generating infrequent morphemes. Additionally, MADAMIRA has a morphological analyzer embedded in it, which in turn enriches the inspection of Arabic words prior to segmentation. Higher EMMA scores are reported for MDMR$_{ATB}$ over MDMR$_{D3}$, which is also expected, as *ArzEnSEG* is segmented following the ATB scheme.

Unsupervised frequency-based segmentation As expected, BPE$_{joint}$ performs the worst in the morphology-based segmentation task, as it is designed for agnostic segmentation for the purpose of improving downstream tasks.

Further analysis We surprisingly find that MorphA$_{tgt}$ outperforms MorphA$_{src}$ on Arabic words and FC$_{src}$ outperforms FC$_{tgt}$ on English words. Therefore, we conduct an internal analysis where we look into the percentage of over and under segmentations.[2] In Appendix C, we present the number of under and over segmented words for each segmentation approach. Our analysis shows that MorphA$_{src}$ over segments 25% of the Arabic words. We observe that in 20% of these over segmentation cases, the Arabic definite article is segmented. For example, the word الكتب *Alktb* 'the books' is segmented to ال#كتب *Al#ktb* which is considered valid in segmentation schemes like D3. However, since we use the ATB scheme in *ArzEnSEG* annotation, the EMMA system penalizes the MorphA$_{src}$ segmenter and rewards the MorphA$_{tgt}$ segmenter which leaves most of the Arabic words and the definite article unsegmented. Another case is the segmentation of affixes, which is not done in ATB. For example, 16% of the over segmentation cases are separation of the Ta-Marbuta (feminine nominal ending) in Arabic words. The rest of the cases are grammatically incorrect segmentations. FC$_{tgt}$ is also shown to under segment around 17% more English words compared to FC$_{src}$ which can contribute to worse scores. We also observe that MORF$_{tgt}$ performs

[2]Over segmentation is a term we use to indicate that the word gets segmented to more morphemes compared to the gold standard segmentation. Meanwhile, under segmentation is a term we use to convey that the word is segmented into fewer morphemes than the gold standard segmentation.

significantly worse than the other segmenters when segmenting Arabic words, despite the fact that 81% of the Arabic words do not require segmentation. Internal analysis shows that MORF$_{tgt}$ over segments the Arabic words to the character level in an attempt to extract the underlying morphology of Egyptian Arabic, which it was not trained on.

5 Machine Translation Experiments

Since no previous research investigates the best segmentation technique for NMT of the code-switched Egyptian Arabic–English language pair, we explore training NMT models using the various segmentation setups discussed in Section 4 to answer RQ1. Moreover, we analyze the performance of the top-performing MT systems on different types of CS sentences to answer RQ2. Afterward, we compare the MT scores against the EMMA F1 scores discussed in Section 4.3 to answer RQ3.

5.1 Experimental Setup

We train Transformer models for our MT systems using Fairseq (Ott et al., 2019) on a single GeForce RTX 3090 GPU. We use the hyperparameters from the FLORES[3] benchmark for low-resource MT Guzmán et al. (2019), which we list in Appendix D. Afterwards, we evaluate the MT models on *ArzEn*'s dev and test sets using chrF2 (Popović, 2015).[4] We choose chrF2 over BLEU (Papineni et al., 2002) as it rewards partially correct translations which makes it a convenient choice for our research, and because chrF has shown to have higher correlation with human judgments over BLEU (Kocmi et al., 2021).

5.2 Machine Translation Systems

We experiment with different categories of segmentation setups. Table 4 shows all the different setups that we explore. See Table 10 in Appendix D for training time.

For the **unsupervised morphology-based segmentations**, we use MorphAGram, Morfessor, FlatCat, and LMVR to segment the source/target sides of the parallel corpus, where the segmenters were trained on each side separately (see Figure 2). For the best performing segmenter, we further investigate the best training setting, where we investigate using segmenters trained only on Arabic words on the source side as well as segmenters that are trained jointly on both sides.

For the **supervised morphology-based segmentations**, we only follow one approach and that is segmenting the source side using MDMR$_{ATB}$ or MDMR$_{D3}$ segmenters. This causes the English words to be left unsegmented.

For the **unsupervised frequency-based segmentations**, we exploit the jointly trained model, BPE$_{joint}$, to segment the source side only, target side only, or both sides of the parallel corpus.

Finally, inspired by the work of Oudah et al. (2019), we explore **combinations** between BPE and supervised morphology-based segmenters. As shown in Table 4, for the source side, we apply BPE$_{joint}$ on top of segmentations provided by either MDMR$_{ATB}$ or MDMR$_{D3}$. For the target side, we either leave it in the raw format or apply BPE$_{joint}$.

Segmentation			chrF2	
Source		Target		
EGY	EN	EN	dev	test
raw		raw	47.1	49.9
Unsupervised Morphology-based Segmenters				
MorphA$_{src}$		MorphA$_{tgt}$	47.0	49.7
MORF$_{src}$		MORF$_{tgt}$	47.4	50.8
FC$_{src}$		FC$_{tgt}$	47.2	50.6
LMVR$_{src}$		LMVR$_{tgt}$	48.3	51.7
LMVR$_{joint}$		LMVR$_{joint}$	48.8	52.5
LMVR$_{src/egy}$	LMVR$_{tgt}$	LMVR$_{tgt}$	**48.9**	**52.9**
LMVR$_{src}$	LMVR$_{tgt}$	LMVR$_{tgt}$	48.8	**52.9**
LMVR$_{src/egy}$	LMVR$_{src}$	LMVR$_{tgt}$	48.5	52.0
Frequency-based Segmenters				
BPE$_{joint}$		BPE$_{joint}$	**50.1**	**53.7**
BPE$_{joint}$		raw	47.4	50.8
raw		BPE$_{joint}$	44.3	46.9
Supervised Morphology-based Segmenters				
MDMR$_{ATB}$	raw	raw	**48.8**	**52.1**
MDMR$_{D3}$	raw	raw	47.9	51.1
Combination Segmenters				
MDMR$_{ATB}$+BPE$_{joint}$	BPE$_{joint}$	raw	46.5	50.1
MDMR$_{ATB}$+BPE$_{joint}$	BPE$_{joint}$	BPE$_{joint}$	**50.2**	**53.8**
MDMR$_{D3}$+BPE$_{joint}$	BPE$_{joint}$	raw	46.9	50.7
MDMR$_{D3}$+BPE$_{joint}$	BPE$_{joint}$	BPE$_{joint}$	49.8	53.3

Table 4: The chrF2 results of our NMT systems with different segmentation combinations on *ArzEn*'s dev and test sets. Numbers highlighted in bold show the best performing system in each category.

5.3 Machine Translation Results

Table 4 shows the different MT systems and their performance on *ArzEn*'s dev and test sets.

Amongst the **unsupervised morphology-based segmenters**, LMVR outperforms the other segmenters. We find that training language-specific

[3]FLORES hyperparameters outperform Vaswani et al. (2017) for our code-switched pair by +0.4 chrF2 points.

[4]We use sacreBLEU's (Post, 2018) implementation of chrF2.

segmenters (using LMVR$_{src/egy}$ for Arabic words and LMVR$_{tgt}$ for English words) outperforms training the segmenter jointly (LMVR$_{joint}$). This setup gives the best performing model, referred to as MT$_{LMVR}$.

Amongst the **supervised morphology-based segmenters**, the setup with MDMR$_{ATB}$ is the best, which we refer to as MT$_{ATB}$. The finding is consistent with Oudah et al. (2019)'s results.

For **unsupervised frequency-based segmenters**, using BPE$_{joint}$ to segment both source and target sides outperforms MT$_{LMVR}$ by +0.8 chrF2 points and MT$_{ATB}$ by +1.6 chrF2 points, which we refer to as MT$_{BPE}$. We observe that the ranking of these segmenters in MT performance is in reverse order compared to their ranking in segmentation task performance. We discuss this later in Section 6.

Most interestingly, contrary to (Oudah et al., 2019), we find that applying BPE$_{joint}$ on top of MDMR$_{ATB}$, which we refer to as MT$_{ATB+BPE}$, slightly improves over MT$_{BPE}$ but without statistical significance. However, MT$_{ATB+BPE}$ outperforms MT$_{ATB}$ and MT$_{LMVR}$ with statistical significance.[5] We further investigate the effectiveness and statistical significance achieved by this approach in a learning curve with varying the training data size in Section 5.4.

Finally, we note that segmenting English words on the source and target sides consistently, while controlling all other conditions, is always advantageous, as shown in Table 4.

5.4 Analysis

We further analyze the performance of the top MT systems from each segmentation setup (MT$_{ATB+BPE}$, MT$_{BPE}$, MT$_{LMVR}$, and MT$_{ATB}$). We first look into the number of Out-of-Vocabulary (OOV) tokens associated with each of the top-performing MT systems to examine whether it has an impact on their final ranking. Secondly, we investigate whether the ranking of the systems is consistent across the different types of sentences. We evaluate the systems against varying morphological richness, English percentages, and CS types. Thirdly, we further investigate the effectiveness of applying BPE over ATB compared to using each segmenter on its own.

[5]We use Paired Significance Tests for Multi System Evaluation provided by SacreBLEU for the significance tests https://github.com/mjpost/sacrebleu#paired-bootstrap-resampling---paired-bs.

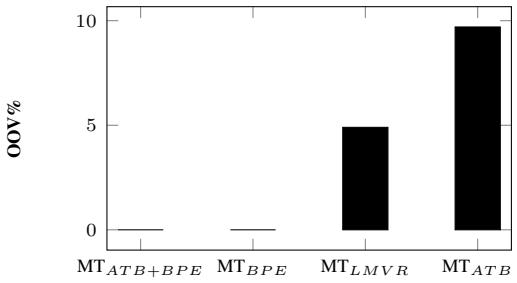


Figure 3: The percentage of the OOV words generated from each of the top-performing MT systems from each segmentation setup on *ArzEn*'s dev set.

We conduct this analysis across different CS types and sizes of training data.

OOV To further study the reason behind MT$_{BPE}$ and MT$_{ATB+BPE}$ top performance, we observe if the top-performing MT systems' ranking is linked with the percentage of OOV in the different MT systems. As shown in Figure 3, we find that for MT$_{ATB+BPE}$ and MT$_{BPE}$, the OOV percentage is 0%. However, for MT$_{LMVR}$ and MT$_{ATB}$, the percentage rises to 4.90% and 9.70%, respectively, which we believe contributes to worsening the MT systems.

Evaluating Systems Under Different Sentence Categories We evaluate the performance of the MT systems for sentences falling under different ranges of (i) morphological richness, (ii) percentage of CS English words, and (iii) sentence CS types. Morphological richness of a sentence is calculated as the quotient of the number of tokens in the segmented sentence and unsegmented original sentence. As expected, the performance of all the MT models decreases as the morphological richness increases and there is a boost in performance across all systems when the percentage of English words increases (see Appendix E). We observe that the MT$_{ATB+BPE}$ and MT$_{BPE}$ perform the best across all ranges for the first two features. We then evaluate the performance of the MT systems across sentences according to CS types: purely monolingual Arabic, CS, and CS having MCS (Hamed et al., 2022a; Mager et al., 2019). We observe that for all systems, the performance across CS sentences is higher than across monolingual Arabic sentences. We also observe that among CS sentences, the performance is reduced in the case of morphologically code-switched sentences. We believe that the following two factors can be contributing to these results. Firstly, the complex MCS

constructions might impose challenges to the MT system. Secondly, we observe that the average length of MCS sentences is higher than that of CS sentences in general. This is partially due to the fact that the tokens in MCS words are space-separated during the data preprocessing step. We report that on average, CS sentences contain 21.1 words (21.4 tokens), while MCS sentences contain 25.0 words (26.3 tokens).

Further Investigating the Effectiveness of $\text{MT}_{ATB+BPE}$ over MT_{BPE} and MT_{ATB} We study whether the ranking of $\text{MT}_{ATB+BPE}$, MT_{BPE}, and MT_{ATB} is altered when going from a low-resource to an extreme low-resource setting across different sentence types. We achieve this by varying the MT training data to 25% and 50% of its original size. The results are shown in Table 5.

We observe that the effectiveness of the $\text{MT}_{ATB+BPE}$ varies under constrained conditions. For monolingual Arabic sentences, when training the MT systems on 100% of data, we see that $\text{MT}_{ATB+BPE}$ is not statistically significant over MT_{BPE} and MT_{ATB}. Moreover, $\text{MT}_{ATB+BPE}$ was outperformed by MT_{BPE}. However, when training with 25% and 50% of data, $\text{MT}_{ATB+BPE}$ outperforms MT_{BPE} and MT_{ATB} with statistical significance across all sentence types. We further exhibit this in Figure 4 when all sentence categories are considered during analysis under different data sizes. This finding highlights the importance of combining morphology-based and frequency-based segmentations in extremely low-resource scenarios.

We also observe that across all data sizes, MT_{ATB} performs the worst on CS sentences. Our first hypothesis is that this is due to English words left unsegmented. However, results in Table 3 contradict this hypothesis. Our second hypothesis is that since MDMR_{ATB} takes into consideration the context of the word prior to segmentation, the English words in the CS sentences might break the flow of the sentence, hence negatively impacting the context of the word, thus worsening the score.

System Selection As per our findings, $\text{MT}_{ATB+BPE}$ is always the best choice across all sentence types in extreme low-resource settings. However, when training on 100% of the data, MT_{BPE} improves slightly over $\text{MT}_{ATB+BPE}$ on monolingual Arabic sentences. Therefore, we create a system selection setup which uses both, $\text{MT}_{ATB+BPE}$ and MT_{BPE}, to investigate if it would lead to further improvements. In this setup, the CS and monolingual English sentences are translated using $\text{MT}_{ATB+BPE}$, while monolingual Arabic sentences are translated using MT_{BPE}. Despite the hybrid system showing an overall improvement of +0.1 chrF2 points over $\text{MT}_{ATB+BPE}$, the improvement is not statistically significant.

Size	MT System	All	EGY	CS	MCS
25%	$\text{MT}_{ATB+BPE}$	**39.8 (1)**	**36.6 (1)**	**40.6 (1)**	**40.0 (1)**
	MT_{BPE}	38.4 (2)	35.6 (3)	39.1 (2)	38.5 (2)
	MT_{ATB}	36.9 (3)	35.9 (2)	37.0 (3)	36.0 (3)
50%	$\text{MT}_{ATB+BPE}$	**45.9 (1)**	**42.1 (1)**	**46.8 (1)**	**46.4 (1)**
	MT_{BPE}	44.5 (2)	40.7 (3)	45.5 (2)	44.8 (2)
	MT_{ATB}	44.0 (3)	41.4 (2)	44.7 (3)	44.0 (3)
100%	$\text{MT}_{ATB+BPE}$	**50.2 (1)**	44.4 (2)	**51.5 (1)**	**51.3 (1)**
	MT_{BPE}	50.1 (2)	**44.6 (1)**	51.3 (2)	51.1 (2)
	MT_{ATB}	48.8 (3)	44.2 (3)	49.8 (3)	49.4 (3)

Table 5: We compare the results of the best performing MT system ($\text{MT}_{ATB+BPE}$) which utilizes BPE on top of ATB segmentation against the MT systems that utilize BPE (MT_{BPE}) or ATB (MT_{ATB}) only on *ArzEn*'s dev set. We report chrF2 results when training on 25%, 50%, and 100% of the training data. Results are shown for different types of sentences: monolingual Egyptian Arabic (EGY), code-switched (CS), and morphologically code-switched (MCS), as well as all sentences (All). The ranking of the MT systems with respect to each other is represented by the numbers between parentheses, where (1) is the best rank and (3) is the worst.

6 Discussion

We revisit the RQs we outlined in our introduction.

RQ1 - Which segmentation setup performs the best in the downstream MT task across different training sizes? Results show that frequency-based segmentation applied on top of morphology-based segmentation outperforms the other segmentation techniques, with statistical significance on lower resource settings. The superiority of this approach is seen across sentences with varying morphological richness, percentage of English words, and across sentences with different CS types. We believe the strength of the combination is because it exploits complementarity of both methods. On one hand, supervised morphology-based segmenters bring in high correctness; however, they are not always robust, having high OOV rates. On the other hand, while BPE segmentation is not necessarily morphologically correct, it achieves high robustness. The robustness of BPE is consistent with the findings in Banerjee and Bhattacharyya (2018).

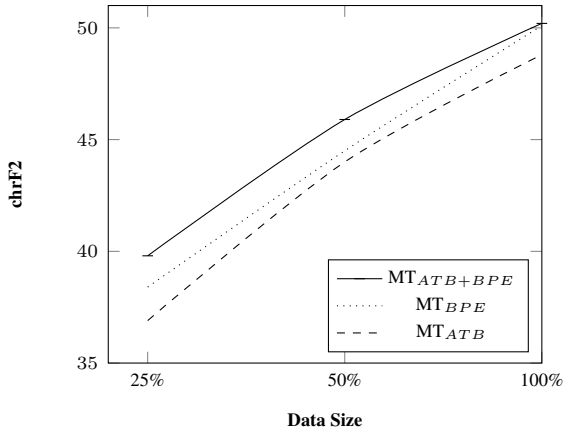


Figure 4: Demonstrates the effectiveness of applying BPE on top of ATB segmentation (MT$_{ATB+BPE}$) as opposed to using either approaches separately (MT$_{BPE}$ and MT$_{ATB}$), which is confirmed when reducing the amount of training data. Results are reported on *ArzEn*'s dev set.

RQ2 - Does the effectiveness of the different segmenters in the MT task differ according to the CS type of the source sentence? We observe that the effectiveness of the different segmenters on MT performance is consistent across two categories of CS sentences; those with and without MCS. However, when comparing their effectiveness on monolingual Arabic vs. CS sentences, we observe that the rankings between segmenters are not consistent. In the case of constrained data size settings (25% and 50% of data), we observe a clear pattern where MT$_{ATB}$ outperforms MT$_{BPE}$ on monolingual sentences, while MT$_{BPE}$ outperforms MT$_{ATB}$ on CS. In the case of using 100% of the training data, MT$_{ATB+BPE}$ outperforms MT$_{BPE}$ on CS sentences; however, MT$_{BPE}$ outperforms MT$_{ATB+BPE}$ on monolingual Arabic sentences. Since our test and dev sets are dominated by CS sentences (61.5% and 63.8%, respectively), we believe that the overall ranking is more greatly affected by the systems' performance on CS sentences, thus reflecting the same ranking on the overall evaluation set as that across CS sentences.

RQ3 - Is there a correlation between a more morphologically correct segmentation and MT performance? For unsupervised morphology-based segmenters, a segmenter with a better segmentation EMMA F1 score also scores better in the downstream MT task. However, we cannot hypothesize that a better segmentation score implies a better translation system, as counter examples exist. For example, while we notice that MDMR$_{ATB}$ gives the best segmentation in terms of EMMA F1 score, it does not outperform any of the top-performing MT systems. We hypothesize that despite MDMR$_{ATB}$'s capability of generating morphologically correct segmentations, it can generate infrequent morphemes due to the out-of-domain data which it is trained on. This may not only increase the sentence length which worsens MT performance as shown in Mager et al. (2022), but may also be one of the contributing factors to the 9.70% OOV percentage found in MT$_{ATB}$. On the contrary, BPE$_{joint}$ performs the worst in the segmentation task as we expect, since it is designated for agnostic-based segmentations; however, it surpasses the top-performing MT models. We believe this is due to its capability to generate semi-correct segmentation and to reduce OOV rates.

7 Conclusion and Future Work

In this paper, we study the impact of a comprehensive set of morphological and frequency-based segmentation methods on MT, where the source is a code-switched text. The experiments are performed on code-switched Arabic-English to English. We found that the supervised morphological segmenter achieved the best performance on the segmentation task, followed by unsupervised morphological methods, and finally, unsupervised frequency-based. Afterward, we train 18 different MT systems with different source and target side segmentations. We find that the rank of the segmenters is reversed, as BPE's could not be outperformed (significantly) by any morphological-inspired segmentation method. However, combining morphology-based and frequency-based segmentations has shown to give improvements, which are statistically significant in lower resource settings, where the training data size is reduced to 25% and 50%. For future work, we plan to apply our different MT setups on other low resource and code-switched language pairs. Specifically, we plan to explore languages with different typologies, to study whether or not the relation between the data size and choice of the segmentation setup (frequency-based, morphology-based, or a mix) is based on morphological features and data size rather than the language itself. Moreover, we plan to extend our annotated dataset, *ArzEnSEG*, by adding further details to allow evaluating different schemes.

Limitations

The first challenge we face in this work is the computational power needed to tune the Morfessor family segmenters. Therefore, in an attempt to overcome this challenge, for the Morfessor family, the choice of the optimal hyperparameters is dependent on the parent tool. For instance, the optimal hyperparameters for Morfessor are directly used in its FlatCat variant and the hyperparameters specific to FlatCat are then tuned. The same applies for LMVR which is a variant of FlatCat. Moreover, we cannot verify whether or not our results will hold for languages with different typologies, specifically those that are low resource and code-switched. Therefore, the results of this research must be seen in light of these limitations.

Ethics Statement

We could not find any potential harm that might derive from this work. However, we understand that translation as a whole can impact the cultural and social life of the people that use it. This has been used in the past in a harmful way, i.e., to spread colonial views (Mbuwayesango, 2018). Therefore, we call the final user to use this work ethically. Regarding the annotation process, all manual annotations were made by a subset of the authors of this paper. Therefore, no hiring of external workers was necessary.

Acknowledgements

We want to thank all the anonymous reviewers for their helpful comments and suggestions. This project has benefited from financial support to Marwa Gaser and Manuel Mager by the DAAD (German Academic Exchange Service).

References

Abdulmohsen Al-Thubaity and Abdullah Al-Subaie. 2015. Effect of word segmentation on Arabic text classification. In *Proceedings of the International Conference on Asian Language Processing (IALP)*, pages 127–131.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

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 the Language Resources and Evaluation Conference (LREC)*, pages 3973–3977.

Tamali Banerjee and Pushpak Bhattacharyya. 2018. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the workshop on subword/character level models*, pages 55–60.

Leonard E Baum and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Raytheon BBN Technologies, Linguistic Data Consortium, and Sakhr Software. 2012. Arabic-Dialect/English parallel text – LDC2012T09. Web Download. Philadelphia: Linguistic Data Consortium.

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 the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. In *Proceedings of the Workshop on Computational Approaches to Code Switching*, pages 1–11.

Shuguang Chen, Gustavo Aguilar, Anirudh Srinivasan, Mona Diab, and Thamar Solorio. 2022. CALCS 2021 shared task: Machine translation for code-switched data. *arXiv preprint arXiv:2202.09625*.

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.

Mrinal Dhar, Vaibhav Kumar, and Manish Shrivastava. 2018. Enabling code-mixed translation: Parallel corpus creation and MT augmentation approach. In *Proceedings of the Workshop on Linguistic Resources for Natural Language Processing*, pages 131–140.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith L Klavans, and Smaranda Muresan. 2020. Morphagram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 7112–7122.

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.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Morfessor EM+Prune: Improved subword segmentation with expectation maximization and pruning. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 3944–3953.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor flatcat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of the International Conference on Computational Linguistics (COLING): Technical Papers*, pages 1177–1185.

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 the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China.

Nizar Habash, Muhammed AbuOdeh, Dima Taji, Reem Faraj, Jamila El Gizuli, and Omar Kallas. 2022. Camel treebank: An open multi-genre Arabic dependency treebank. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 2672–2681.

Nizar Habash, Mona Diab, and Owen Rambow. 2012a. Conventional Orthography for Dialectal Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 711–718, Istanbul, Turkey.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012b. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 1–9, Montréal, Canada.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Atlanta, Georgia.

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. Springer, Netherlands.

Nizar Y Habash. 2010. Introduction to Arabic natural language processing. *Synthesis lectures on human language technologies*, 3(1):1–187.

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 the International Conference on Language Resources and Evaluation (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 the Workshop for Arabic Natural Language Processing (WANLP)*.

Injy Hamed, Nizar Habash, Slim Abdennadher, and Ngoc Thang Vu. 2022c. Investigating lexical replacements for Arabic-English code-switched data augmentation. *arXiv preprint arXiv:2205.12649*.

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

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Conference on Machine Translation*, pages 478–494.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations*, pages 66–71, Brussels, Belgium.

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.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the International Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, Seth Kulick, Dalila Tabessi, and Sondos Krouna. 2012. Egyptian Arabic Treebank DF Parts 1-8 V2.0 - LDC catalog numbers LDC2012E93, LDC2012E98, LDC2012E89, LDC2012E99, LDC2012E107, LDC2012E125, LDC2013E12, LDC2013E21.

Manuel Mager, Özlem Çetinoğlu, and Katharina Kann. 2019. Subword-level language identification for intra-word code-switching. 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 2005–2011.

Manuel Mager, Arturo Oncevay, Elisabeth Maier, Katharina Kann, and Thang Vu. 2022. BPE vs. morphological segmentation: A case study on machine translation of four polysynthetic languages. In *Findings of*

the Association for Computational Linguistics (ACL), pages 961–971.

Sainik Kumar Mahata, Soumil Mandal, Dipankar Das, and Sivaji Bandyopadhyay. 2019. Code-mixed to monolingual translation framework. In *Proceedings of the Forum for Information Retrieval Evaluation*, pages 30–35.

Dora Rudo Mbuwayesango. 2018. The bible as tool of colonization. *Colonialism and the Bible: Contemporary Reflections from the Global South*, page 31.

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

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 7022–7032, Marseille, France.

John E Ortega, Richard Castro Mamani, and Kyunghyun Cho. 2020. Neural machine translation with a polysynthetic low resource language. *Machine Translation*, 34(4):325–346.

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 the Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Mai Oudah, Amjad Almahairi, and Nizar Habash. 2019. The impact of preprocessing on Arabic-English statistical and neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 214–221.

Yirong Pan, Xiao Li, Yating Yang, and Rui Dong. 2020. Morphological word segmentation on agglutinative languages for neural machine translation. *arXiv preprint arXiv:2001.01589*.

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

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 the International Conference on Language Resources and Evaluation (LREC)*, pages 1094–1101, Reykjavik, Iceland.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

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

Christian Roest, Lukas Edman, Gosse Minnema, Kevin Kelly, Jennifer Spenader, and Antonio Toral. 2020. Machine translation for English–Inuktitut with segmentation, data acquisition and pre-training. In *Proceedings of the Conference on Machine Translation*, pages 274–281.

Jonne Saleva and Constantine Lignos. 2021. The effectiveness of morphology-aware segmentation in low-resource neural machine translation. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL): Student Research Workshop*, pages 164–174.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 1715–1725, Berlin, Germany.

R. Mahesh K. Sinha and Anil Thakur. 2005. Machine translation of bi-lingual Hindi-English (Hinglish) text. In *Proceedings of Machine Translation Summit X: Papers*, pages 149–156, Phuket, Thailand.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 21–24, Gothenburg, Sweden.

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 (NAACL-HLT)*, pages 449–459.

Sebastian Spiegler and Christian Monson. 2010. EMMA: A novel evaluation metric for morphological analysis. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1029–1037.

Vivek Srivastava and Mayank Singh. 2020. PHINC: A parallel Hinglish social media code-mixed corpus for machine translation. In *Proceedings of the Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 41–49.

Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal dependencies for Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 166–176.

Ishan Tarunesh, Syamantak Kumar, and Preethi Jyothi. 2021. From machine translation to code-switching: Generating high-quality code-switched text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3154–3169.

Ahmed Tawfik, Mahitab Emam, Khaled Essam, Robert Nabil, and Hany Hassan. 2019. Morphology-aware word-segmentation in dialectal Arabic adaptation of neural machine translation. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 11–17.

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.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report, Aalto University, Finland.

Jitao Xu and François Yvon. 2021. Can you traducir this? machine translation for code-switched input. In *Proceedings of the Workshop on Computational Approaches to Linguistic Code-Switching*, pages 84–94.

Nasser Zalmout and Nizar Habash. 2017. Optimizing tokenization choice for machine translation across multiple target languages. *The Prague Bulletin of Mathematical Linguistics*, 108:257–270.

A Data Preprocessing

We use the same preprocessing pipeline for all the corpora, where we start by removing any corpus-related annotations. Afterward, we remove URLs and emoticons, through *tweet-preprocessor*,[6] remove trailing and leading spaces, and tokenize numbers. Finally, *Moses Tokenizer*[7] is applied for tokenization and empty lines are removed from the parallel corpora. For LDC2017T07 (Chen et al., 2017) and LDC2019T01 (Chen et al., 2019), some sentences have literal and intended translations for some words. Hence, we opt for one translation having all literal translations and another having all intended translations. Once all the preprocessing steps are done, we concatenate the nine corpora collectively and pass the resulting training corpus to MADAMIRA (Pasha et al., 2014) to obtain two different supervised morphological segmentations of the corpus, namely `ATB_BWFORM` and `D3_BWFORM` which we discuss in Section 4.2. Additionally, we obtain a raw training corpus by further tokenizing punctuation and removing emojis using MADAMIRA's D0 scheme (Zalmout and Habash, 2017). Nonetheless, we normalize the Arabic letters ى and أ to ي and ا respectively through CAMeL Tools (Obeid et al., 2020) since D0's output is not normalized.

B Segmenters' Hyperparameters

Morfessor family Since all Morfessor family segmenters are morphology inspired, the hyperparameters are tuned on *ArzEnSEG*'s dev set. For LMVR$_{src}$ and LMVR$_{tgt}$ setting the vocabulary sizes to $64k$ and $16k$ respectively outperform $3k$, $5k$, $8k$, $16k$, $32k$, $100k$. For LMVR$_{joint}$ setting the vocabulary size to $32k$ outperforms $3k$, $5k$, $8k$, $16k$, $64k$, and $100k$. Meanwhile, For LMVR$_{src/egy}$ setting the vocabulary size to $64k$ outperforms $3k$, $5k$, $8k$, $16k$, $32k$, and $100k$.

Table 6 shows the possible values used during the optimal hyperparameter search for each Morfessor tool. For Morfessor, FlatCat, and LMVR 18, 360, and 7 different segmentation models are generated. These are a result of the combination of the possible hyperparameter values. The hyperparameter combination which yields the highest EMMA score on *ArzEnSEG*'s dev set for each Morfessor tool is used to segment the MT training data. The best combination values are reported in Table 7.

Segmenters Hyperparameters	
Hyperparameter	**Values Bound**
Morfessor	
-F	[0.003, 0.005, 0.007]
-d	[log, ones, none]
-a	[recursive, viterbi]
FlatCat	
-p	[50, 60, 70, 80, 90, 100, 200, 300]
–min-perplexity-length	[1, 2, 3, 4, 5]
–min-shift-remainder	[1, 2, 3]
–length-threshold	[2, 3, 4]
LMVR	
–lexicon-size	[3k, 5k, 8k, 16k, 32k, 64k, 100k]

Table 6: The values bound we use during the best hyperparameter combination search for the Morfessor tools.

MorphAGram Akin to the Morfessor family, we tune the hyperparameters on *ArzEnSEG*'s dev set and train two models: one on the source side and the other on the target side of the training parallel corpus which we refer to as MorphA$_{src}$ and MorphA$_{tgt}$, respectively (see Figure 2). Tuning results show that setting the vocabulary size to $3k$ for MorphA$_{src}$ outperforms $5k$, $8k$, $16k$, $32k$, and $50k$, while setting the vocabulary size to $50k$ for MorphA$_{tgt}$ outperforms $5k$, $3k$, $8k$, $16k$, and $32k$. Nevertheless, it is worth noting that the vocabulary size on the target side is $< 50k$ which shows that MorphA$_{tgt}$ performs the best when no segmentations are applied on the English words.

BPE Since BPE is a segmentation technique that is designated for agnostic segmentation for MT tasks, we tune the vocabulary size on *ArzEn*'s dev set in an NMT task. We apply a vocabulary size of $8k$, which outperforms $5k, 16k, 32k, 64k$.

C Segmenters Performance Analysis

Table 8 shows the error analysis we perform on the segmenters with regards to over segmentation, under segmentation, or generating the correct number of segmentations per word.

D MT Hyperparameters

The MT hyperparameters are shown in Table 9. We follow the FLORES hyperparameters for low-resource language pairs. The full train command can be found on FLORES GitHub.[8] The training

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

[7] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[8] https://github.com/facebookresearch/flores/blob/6641ec0e23d173906dd2e01551a430884b1dba31/floresv1/README.md#train-a-baseline-transformer-model

	Morfessor			FlatCat				LMVR
Data	**-F**	**-d**	**-a**	**-p**	**–min -perplexity -length**	**–min -shift -remainder**	**–length -threshold**	**–lexicon -size**
src	0.003	log	recursive	200	1	1	4	64k
tgt	0.003	log	recursive	100	4	2	4	16k
src/egy	0.007	log	recursive	300	1	1	2	64k
joint	0.007	log	recursive	300	4	2	4	32k

Table 7: The different hyperparameters used for each Morfessor family segmenter depending on whether the model is trained on the source (src), target (tgt), source without English, i.e., Egyptian, (src/egy), or source+target (joint) side(s).

	EGY					EN				
Segmenter	**under**	**over**	**correct**	**seg.**	**unseg.**	**under**	**over**	**correct**	**seg.**	**unseg.**
raw	634	0	2,780	0	2,780	71	0	430	0	430
MorphA$_{src}$	249	855	2,310	385	1,925	70	45	386	1	385
MORF$_{src}$	466	299	2,649	148	2,501	15	103	383	42	341
FC$_{src}$	592	8	2,814	42	2,772	56	7	438	15	423
LMVR$_{src}$	520	47	2,847	111	2,736	43	7	451	28	423
MorphA$_{tgt}$	634	35	2,745	0	2,745	6	148	347	65	282
MORF$_{tgt}$	0	3,150	264	3	261	21	37	443	49	394
FC$_{tgt}$	634	0	2,780	0	2,780	66	8	427	5	422
LMVR$_{tgt}$	634	0	2,780	0	2,780	23	19	459	48	411
LMVR$_{joint}$	485	79	2,850	144	2,706	20	32	449	51	398
BPE$_{joint}$	338	368	2,708	230	2,478	28	132	341	30	311
MDMR$_{ATB}$	38	62	3,314	581	2,733	71	0	430	0	430
MDMR$_{D3}$	38	293	3,083	561	2,522	71	0	430	0	430

Table 8: The table shows the number of under segmented words (under), over segmented words (over), and the number of cases where the segmenter generates the correct count of morphemes (correct) for English (EN) and Arabic (EGY) words in *ArzEnSEG* test set. Additionally, out of the correct count of morphemes (correct), we report the words which originally require segmentation (seg.) and those which do not (unseg.).

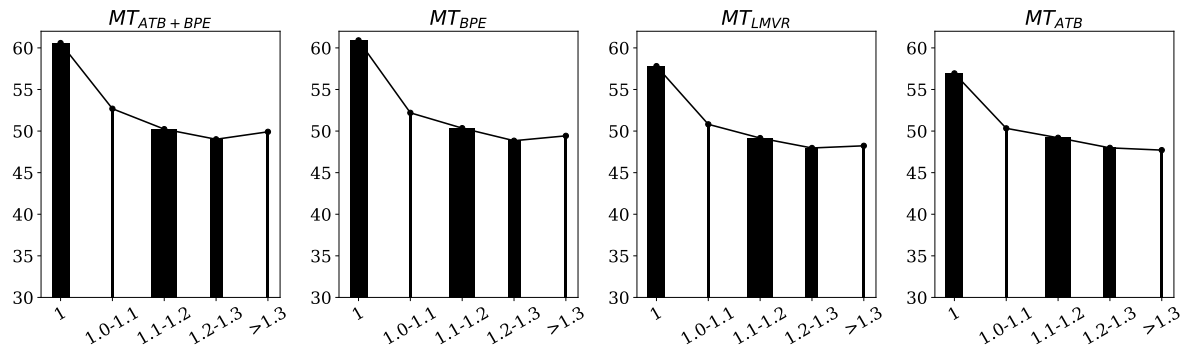
time for MT model the training time is exhibited in Table 10.

Hyperparameter	Value
encoder-layers	5
decoder-layer	5
encoder-embed-dim	512
decoder-embed-dim	512
encoder-ffn-embed-dim	2
decoder-ffn-embed-dim	2
dropout	0.4
attention-dropout	0.2
relu-dropout	0.2
weight-decay	0.0001
label-smoothing	0.2
warmup-updates	4000
warmup-init-lr	1e-9

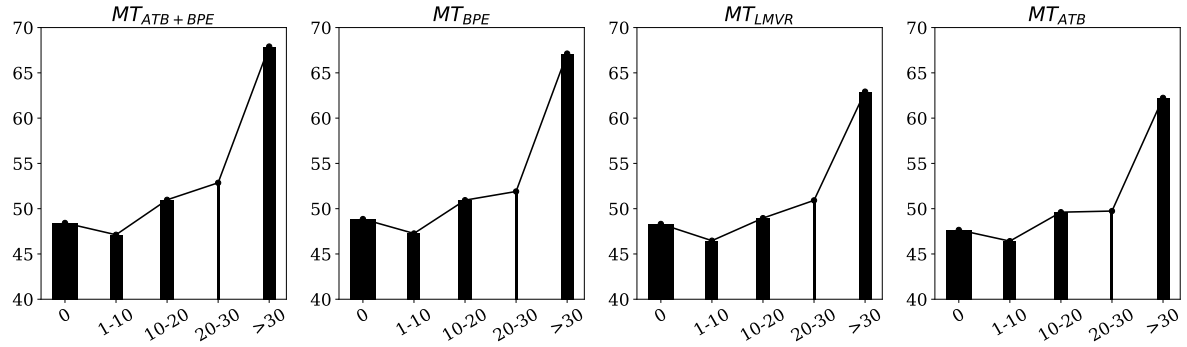
Table 9: FLORES hyperparameters for low-resource language pairs.

E Evaluating Systems Under Different Sentence Categories

Figure 5 shows the performance of the top MT systems from each segmentation setup across sentences of different morphological richness ratios and different percentages of English words in *ArzEn*'s dev set. Results show that there is a general decrease in performance as the morphological richness increases. However, as the percentage of English words in the sentences increases, the performance of the systems generally improves. It is also shown that MT$_{ATB+BPE}$ and MT$_{BPE}$ achieve overall comparable performances and outperform the other systems.



(a) Morphological Richness



(b) % English Words

Figure 5: The average chrF2 scores for the top performing MT systems from each segmentation setup across sentences with various (a) morphological richness ratios and (b) percentage of English words in *ArzEn*'s dev set. Morphological richness of a sentence is calculated as the quotient of the number of tokens in the segmented sentence and unsegmented original sentence. The bar width is indicative of the number of sentences in each bin.

Segmentation			**Training Time**
Source		**Target**	
EGY	**EN**	**EN**	***(seconds)***
raw		raw	13,522
Unsupervised Morphology-based Segmenters			
MorphA$_{src}$		MorphA$_{tgt}$	24,731
MORF$_{src}$		MORF$_{tgt}$	18,916
FC$_{src}$		FC$_{tgt}$	18,225
LMVR$_{src}$		LMVR$_{tgt}$	4,476
LMVR$_{joint}$		LMVR$_{joint}$	18,019
LMVR$_{src/egy}$	LMVR$_{tgt}$	LMVR$_{tgt}$	22,462
LMVR$_{src}$	LMVR$_{tgt}$	LMVR$_{tgt}$	4,181
LMVR$_{src/egy}$	LMVR$_{src}$	LMVR$_{tgt}$	4,526
Frequency-based Segmenters			
BPE$_{joint}$		BPE$_{joint}$	18,279
BPE$_{joint}$		raw	23,193
raw		BPE$_{joint}$	17,905
Supervised Morphology-based Segmenters			
MDMR$_{ATB}$	raw	raw	18,280
MDMR$_{D3}$	raw	raw	18,519
Combination Segmenters			
MDMR$_{ATB}$+BPE$_{joint}$	BPE$_{joint}$	raw	17,629
MDMR$_{ATB}$+BPE$_{joint}$	BPE$_{joint}$	BPE$_{joint}$	27,088
MDMR$_{D3}$+BPE$_{joint}$	BPE$_{joint}$	raw	24,256
MDMR$_{D3}$+BPE$_{joint}$	BPE$_{joint}$	BPE$_{joint}$	23,611

Table 10: The training time in seconds of our different NMT systems.