
Marwa Gaser,1 Manuel Mager,2∗ Injy Hamed,3,4 Nizar Habash,4 Slim Abdennadher,1 Ngoc Thang Vu3

1The German University in Cairo, 2AWS AI Labs
3University of Stuttgart, 4New 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 resource 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).

∗Work done while at the University of Stuttgart.
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-sense 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.

1http://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; Tawfiq 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.
(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.
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<sub>src</sub>, FC<sub>src</sub>, and LMVR<sub>src</sub> resemble Morfessor, FlatCat, and LMVR respectively, where the segmenters are trained on the source side. MORF<sub>tgt</sub>, FC<sub>tgt</sub>, and LMVR<sub>tgt</sub> 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<sub>joint</sub>.

**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<sub>ATB</sub> and MDMR<sub>D3</sub> respectively. Both schemes apply Alif/Ya normaliza-
### Supervised morphology-based segmentation

As shown in Table 3, both supervised morphology-based segmenters $\text{MDMR}_{AB}$ and $\text{MDMR}_{D3}$ 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 $\text{MDMR}_{AB}$ over $\text{MDMR}_{D3}$, which is also expected, as $\text{ArzEnSEG}$ is segmented following the ATB scheme.

### Unsupervised frequency-based segmentation

As expected, $\text{BPE}_{\text{joint}}$ 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 $\text{MorphA}_{tgt}$ outperforms $\text{MorphA}_{src}$ on Arabic words and $\text{FC}_{src}$ outperforms $\text{FC}_{tgt}$ 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 $\text{MorphA}_{src}$ 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 ‘the books’ is segmented to $\text{Al#ktb}$, which is considered valid in segmentation schemes like D3. However, since we use the ATB scheme in $\text{ArzEnSEG}$ annotation, the EMMA system penalizes the $\text{MorphA}_{src}$ segmenter and rewards the $\text{MorphA}_{tgt}$ 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. $\text{FC}_{tgt}$ is also shown to under segment around 17% more English words compared to $\text{FC}_{src}$ which can contribute to worse scores. We also observe that $\text{MORF}_{tgt}$ performs worse.

\(^2\)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.

---

### Table 3: EMMA F1 score calculated on $\text{ArzEnSEG}$ test set

<table>
<thead>
<tr>
<th>Segmenter</th>
<th>EGY</th>
<th>EN</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>0.806</td>
<td>0.953</td>
<td>0.838</td>
</tr>
<tr>
<td>$\text{MorphA}_{src}$</td>
<td>0.682</td>
<td>0.942</td>
<td>0.737</td>
</tr>
<tr>
<td>$\text{MORF}_{src}$</td>
<td>0.814</td>
<td>0.888</td>
<td>0.832</td>
</tr>
<tr>
<td>$\text{FC}_{src}$</td>
<td>0.821</td>
<td>0.961</td>
<td>0.851</td>
</tr>
<tr>
<td>$\text{LMVR}_{src}$</td>
<td>0.836</td>
<td>0.961</td>
<td>0.863</td>
</tr>
<tr>
<td>$\text{LMVR}_{src/egy}$</td>
<td>0.838</td>
<td>0.953</td>
<td>0.863</td>
</tr>
<tr>
<td>$\text{MorphA}_{tgt}$</td>
<td>0.806</td>
<td>0.953</td>
<td>0.838</td>
</tr>
<tr>
<td>$\text{MORF}_{tgt}$</td>
<td>0.814</td>
<td>0.951</td>
<td>0.837</td>
</tr>
<tr>
<td>$\text{FC}_{tgt}$</td>
<td>0.821</td>
<td>0.961</td>
<td>0.842</td>
</tr>
<tr>
<td>$\text{LMVR}_{tgt}$</td>
<td>0.836</td>
<td>0.961</td>
<td>0.842</td>
</tr>
<tr>
<td>$\text{LMVR}_{\text{joint}}$</td>
<td>0.841</td>
<td>0.963</td>
<td>0.868</td>
</tr>
<tr>
<td>$\text{BPE}_{\text{joint}}$</td>
<td>0.678</td>
<td>0.813</td>
<td>0.707</td>
</tr>
<tr>
<td>$\text{MDMR}_{\text{ATB}}$</td>
<td>0.935</td>
<td>0.953</td>
<td>0.939</td>
</tr>
<tr>
<td>$\text{MDMR}_{\text{D3}}$</td>
<td>0.868</td>
<td>0.953</td>
<td>0.887</td>
</tr>
</tbody>
</table>

---

4.4 Segmentation Results

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

### Unsupervised morphology-based segmentation

Results show that $\text{LMVR}$ 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 $\text{LMVR}$, given that it outperforms the other segmenters. We refer to these models as $\text{LMVR}_{\text{joint}}$ and $\text{LMVR}_{\text{src/egy}}$ respectively, as outlined in Figure 2. Results show that joint training provides best EMMA scores.
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). 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

$$\begin{array}{|l|l|l|l|}
\hline
\text{Segmentation} & \text{chrF2} \\
\hline
\text{Source} & \text{Target} & \text{EGY} & \text{EN} \\
\hline
\text{raw} & \text{raw} & \text{47.1} & \text{49.9} \\
\hline
\hline
\text{Unsupervised Morphology-based Segmenters} & & & \\
\hline
\text{MorphA$_{src}$} & \text{MorphA$_{tgt}$} & \text{47.0} & \text{49.7} \\
\text{MORF$_{src}$} & \text{MORF$_{tgt}$} & \text{47.4} & \text{50.8} \\
\text{FC$_{src}$} & \text{FC$_{tgt}$} & \text{47.2} & \text{50.6} \\
\text{LMVR$_{src}$} & \text{LMVR$_{tgt}$} & \text{48.3} & \text{51.7} \\
\text{LMVR$_{joint}$} & \text{LMVR$_{joint}$} & \text{48.8} & \text{52.5} \\
\hline
\text{Frequency-Based Segmenters} & & & \\
\hline
\text{BPE$_{joint}$} & \text{BPE$_{joint}$} & \text{50.1} & \text{53.7} \\
\text{BPE$_{joint}$} & \text{raw} & \text{47.4} & \text{50.8} \\
\text{raw} & \text{BPE$_{joint}$} & \text{44.3} & \text{46.9} \\
\hline
\text{Supervised Morphology-based Segmenters} & & & \\
\hline
\text{MDMR$_{ATB}$} & \text{raw} & \text{48.8} & \text{52.1} \\
\text{MDMR$_D3$} & \text{raw} & \text{47.9} & \text{51.1} \\
\hline
\text{Combination Segmenters} & & & \\
\hline
\text{MDMR$_{ATB}$+BPE$_{joint}$} & \text{BPE$_{joint}$} & \text{46.5} & \text{50.1} \\
\text{MDMR$_{ATB}$+BPE$_{joint}$} & \text{BPE$_{joint}$} & \text{50.2} & \text{53.8} \\
\text{MDMR$_D3$+BPE$_{joint}$} & \text{BPE$_{joint}$} & \text{46.9} & \text{50.7} \\
\text{MDMR$_D3$+BPE$_{joint}$} & \text{BPE$_{joint}$} & \text{49.8} & \text{53.3} \\
\hline
\end{array}$$

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.

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$_{joint}$.

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
segmenters (using LMVRsrc/egy for Arabic words and LMVRtgt for English words) outperforms training the segmenter jointly (LMVRjoint). This setup gives the best performing model, referred to as MTLMVR.

Amongst the supervised morphology-based segmenters, the setup with MDMRATB is the best, which we refer to as MTATB. The finding is consistent with Oudah et al. (2019)’s results.

For unsupervised frequency-based segmenters, using BPEjoint to segment both source and target sides outperforms MTLMVR by +0.8 chrF2 points and MTATB by +1.6 chrF2 points, which we refer to as MTBPE. 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 BPEjoint on top of MDMRATB, which we refer to as MTATB+BPE, slightly improves over MTBPE but without statistical significance. However, MTATB+BPE outperforms MTATB and MTLMVR with statistical significance.\footnote{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.} 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 (MTATB+BPE, MTBPE, MTLMVR, and MTATB). 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.

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

OOV To further study the reason behind MTBPE and MTATB+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 MTATB+BPE and MTBPE, the OOV percentage is 0%. However, for MTLMVR and MTATB, 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 MTATB+BPE and MTBPE 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 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 MT\textsubscript{ATB+BPE} varies under constrained conditions. For monolingual Arabic sentences, when training the MT systems on 100% of data, we see that MT\textsubscript{ATB+BPE} is not statistically significant over MT\textsubscript{BPE} and MT\textsubscript{ATB}. Moreover, MT\textsubscript{ATB+BPE} was outperformed by MT\textsubscript{BPE}. However, when training with 25% and 50% of data, MT\textsubscript{ATB+BPE} outperforms MT\textsubscript{BPE} and MT\textsubscript{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\textsubscript{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\textsubscript{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, MT\textsubscript{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\textsubscript{BPE} improves slightly over MT\textsubscript{ATB+BPE} on monolingual Arabic sentences. Therefore, we create a system selection setup which uses both, MT\textsubscript{ATB+BPE} and MT\textsubscript{BPE}, to investigate if it would lead to further improvements. In this setup, the CS and monolingual English sentences are translated using MT\textsubscript{ATB+BPE}, while monolingual Arabic sentences are translated using MT\textsubscript{BPE}. Despite the hybrid system showing an overall improvement of +0.1 chrF2 points over MT\textsubscript{ATB+BPE}, the improvement is not statistically significant.

### 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).
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 morphologically-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 extend our annotated dataset, ArzEn’s dev set, 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**


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.


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 ﯧ and ﯦ to ﯧ and ﯦ 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 64\(k\) and 16\(k\) respectively outperform 3\(k\), 5\(k\), 8\(k\), 16\(k\), 32\(k\), and 100\(k\). For LMVR\(_{joint}\) setting the vocabulary size to 32\(k\) outperforms 3\(k\), 5\(k\), 8\(k\), 16\(k\), 64\(k\), and 100\(k\). Meanwhile, For LMVR\(_{src/egy}\) setting the vocabulary size to 64\(k\) outperforms 3\(k\), 5\(k\), 8\(k\), 16\(k\), 32\(k\), and 100\(k\).

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.

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

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 3\(k\) for MorphA\(_{src}\) outperforms 5\(k\), 8\(k\), 16\(k\), 32\(k\), and 50\(k\); while setting the vocabulary size to 50\(k\) for MorphA\(_{tgt}\) outperforms 5\(k\), 3\(k\), 8\(k\), 16\(k\), and 32\(k\). Nevertheless, it is worth noting that the vocabulary size on the target side is < 50\(k\) 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 8\(k\), which outperforms 5\(k\), 16\(k\), 32\(k\), and 64\(k\).

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/facebookresearch/flores/blob/6641ec0e23d173906dd2e01551a430884b1db31/floresv1/README.md#train-a-baseline-transformer-model
### 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).

<table>
<thead>
<tr>
<th>Segmenter</th>
<th>src</th>
<th>tgt</th>
<th>src/egy</th>
<th>joint</th>
</tr>
</thead>
<tbody>
<tr>
<td>hyperparam</td>
<td>-F</td>
<td>-d</td>
<td>-a</td>
<td>-p</td>
</tr>
<tr>
<td>morphology</td>
<td>0.003</td>
<td>0.003</td>
<td>0.007</td>
<td>0.007</td>
</tr>
<tr>
<td>perplexity</td>
<td>log</td>
<td>recursive</td>
<td>200</td>
<td>100</td>
</tr>
<tr>
<td>shift</td>
<td>-min</td>
<td>-length</td>
<td>shift</td>
<td>-threshold</td>
</tr>
<tr>
<td>threshold</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>size</td>
<td>64k</td>
<td>16k</td>
<td>64k</td>
<td>32k</td>
</tr>
</tbody>
</table>

### 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.).

<table>
<thead>
<tr>
<th>Segmenter</th>
<th>EGY</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>634</td>
<td>71</td>
</tr>
<tr>
<td>MorphA$_{src}$</td>
<td>249</td>
<td>70</td>
</tr>
<tr>
<td>MORF$_{src}$</td>
<td>466</td>
<td>15</td>
</tr>
<tr>
<td>FC$_{src}$</td>
<td>592</td>
<td>56</td>
</tr>
<tr>
<td>LMVR$_{src}$</td>
<td>520</td>
<td>43</td>
</tr>
<tr>
<td>MorphA$_{tgt}$</td>
<td>634</td>
<td>6</td>
</tr>
<tr>
<td>MORF$_{tgt}$</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td>FC$_{tgt}$</td>
<td>634</td>
<td>66</td>
</tr>
<tr>
<td>LMVR$_{tgt}$</td>
<td>634</td>
<td>23</td>
</tr>
<tr>
<td>LMVR$_{joint}$</td>
<td>485</td>
<td>20</td>
</tr>
<tr>
<td>BPE$_{joint}$</td>
<td>338</td>
<td>28</td>
</tr>
<tr>
<td>MDMR$_{ATB}$</td>
<td>38</td>
<td>71</td>
</tr>
<tr>
<td>MDMR$_{DS}$</td>
<td>38</td>
<td>71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>encoder-layers</td>
<td>3</td>
</tr>
<tr>
<td>decoder-layer</td>
<td>5</td>
</tr>
<tr>
<td>encoder-embed-dim</td>
<td>512</td>
</tr>
<tr>
<td>decoder-embed-dim</td>
<td>512</td>
</tr>
<tr>
<td>encoder-ffn-embed-dim</td>
<td>2</td>
</tr>
<tr>
<td>decoder-ffn-embed-dim</td>
<td>2</td>
</tr>
<tr>
<td>dropout</td>
<td>0.4</td>
</tr>
<tr>
<td>attention-dropout</td>
<td>0.2</td>
</tr>
<tr>
<td>relu-dropout</td>
<td>0.2</td>
</tr>
<tr>
<td>weight-decay</td>
<td>0.0001</td>
</tr>
<tr>
<td>label-smoothing</td>
<td>0.2</td>
</tr>
<tr>
<td>warmup-updates</td>
<td>4000</td>
</tr>
<tr>
<td>warmup-init-lr</td>
<td>1e-9</td>
</tr>
</tbody>
</table>

### 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.
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 ArcEn’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.

<table>
<thead>
<tr>
<th>Segmentation</th>
<th>Training Source</th>
<th>Target</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGY</td>
<td>EN</td>
<td>EN</td>
<td>raw</td>
</tr>
<tr>
<td>raw</td>
<td>raw</td>
<td>13,522</td>
<td></td>
</tr>
<tr>
<td>MorphA&lt;sub&gt;src&lt;/sub&gt;</td>
<td>MorphA&lt;sub&gt;tgt&lt;/sub&gt;</td>
<td>24,731</td>
<td></td>
</tr>
<tr>
<td>MORF&lt;sub&gt;src&lt;/sub&gt;</td>
<td>MORF&lt;sub&gt;tgt&lt;/sub&gt;</td>
<td>18,916</td>
<td></td>
</tr>
<tr>
<td>FC&lt;sub&gt;src&lt;/sub&gt;</td>
<td>FC&lt;sub&gt;tgt&lt;/sub&gt;</td>
<td>18,225</td>
<td></td>
</tr>
<tr>
<td>LMVR&lt;sub&gt;src&lt;/sub&gt;</td>
<td>LMVR&lt;sub&gt;tgt&lt;/sub&gt;</td>
<td>4,476</td>
<td></td>
</tr>
<tr>
<td>LMVR&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>LMVR&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>18,019</td>
<td></td>
</tr>
<tr>
<td>LMVR&lt;sub&gt;src/egy&lt;/sub&gt;</td>
<td>LMVR&lt;sub&gt;src&lt;/sub&gt;</td>
<td>22,462</td>
<td></td>
</tr>
<tr>
<td>LMVR&lt;sub&gt;src&lt;/sub&gt;</td>
<td>LMVR&lt;sub&gt;tgt&lt;/sub&gt;</td>
<td>4,181</td>
<td></td>
</tr>
<tr>
<td>LMVR&lt;sub&gt;src/egy&lt;/sub&gt;</td>
<td>LMVR&lt;sub&gt;src&lt;/sub&gt;</td>
<td>4,526</td>
<td></td>
</tr>
<tr>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>18,279</td>
<td></td>
</tr>
<tr>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>raw</td>
<td>23,193</td>
<td></td>
</tr>
<tr>
<td>raw</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>17,905</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;ATB&lt;/sub&gt;</td>
<td>raw</td>
<td>18,280</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;D3&lt;/sub&gt;</td>
<td>raw</td>
<td>18,519</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;ATB&lt;/sub&gt;+BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>17,629</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;ATB&lt;/sub&gt;+BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>27,088</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;D3&lt;/sub&gt;+BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>24,256</td>
<td></td>
</tr>
<tr>
<td>MDMR&lt;sub&gt;D3&lt;/sub&gt;+BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>BPE&lt;sub&gt;joint&lt;/sub&gt;</td>
<td>23,611</td>
<td></td>
</tr>
</tbody>
</table>

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