Hassan Sajjad Fahim Dalvi Nadir Durrani Ahmed Abdelali Yonatan Belinkov∗ Stephan Vogel
Qatar Computing Research Institute – HBKU, Doha, Qatar
{hsajjad, faimaduddin, ndurrani, aabdelali, svogel}@qf.org.qa
∗MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
belinkov@mit.edu

Abstract
Word segmentation plays a pivotal role in improving any Arabic NLP application. Therefore, a lot of research has been spent in improving its accuracy. Off-the-shelf tools, however, are: i) complicated to use and ii) domain/dialect dependent. We explore three language-independent alternatives to morphological segmentation using: i) data-driven sub-word units, ii) characters as a unit of learning, and iii) word embeddings learned using a character CNN (Convolution Neural Network). On the tasks of Machine Translation and POS tagging, we found these methods to achieve close to, and occasionally surpass state-of-the-art performance. In our analysis, we show that a neural machine translation system is sensitive to the ratio of source and target tokens, and a ratio close to 1 or greater, gives optimal performance.

1 Introduction
Arabic word segmentation has shown to significantly improve output quality in NLP tasks such as machine translation (Habash and Sadat, 2006; Almahairi et al., 2016), part-of-speech tagging (Diab et al., 2004; Habash and Rambow, 2005), and information retrieval (M. Aljlayl and Grossman, 2002). A considerable amount of research has therefore been spent on morphological segmentation in the past two decades, ranging from rule-based analyzers (Beesley, 1996; Durrani and Hussain, 2010) to state-of-the-art statistical segmenters (Pasha et al., 2014; Abdelali et al., 2016).

Morphological segmentation splits words into morphemes. For example, “wktAbnA” (gloss: and our book) is decomposed into its stem and affixes as: “w+ ktAb +nA”.

Despite the gains obtained from using morphological segmentation, there are several caveats to using these tools. Firstly, they make the training pipeline cumbersome, as they come with complicated pre-processing (and additional post-processing in the case of English-to-Arabic translation (El Kholy and Habash, 2012)). More importantly, these tools are dialect- and domain-specific. A segmenter trained for modern standard Arabic (MSA) performs significantly worse on dialectal Arabic (Habash et al., 2013), or when it is applied to a new domain.

In this work, we explore whether we can avoid the language-dependent pre/post-processing components and learn segmentation directly from the training data being used for a given task. We investigate data-driven alternatives to morphological segmentation using i) unsupervised sub-word units obtained using byte-pair encoding (Sennrich et al., 2016), ii) purely character-based segmentation (Ling et al., 2015), and iii) a convolutional neural network over characters (Kim et al., 2016).

We evaluate these techniques on the tasks of machine translation (MT) and part-of-speech (POS) tagging and compare them against morphological segmenters MADAMIRA (Pasha et al., 2014) and Farasa (Abdelali et al., 2016). On the MT task, byte-pair encoding (BPE) performs the best among the three methods, achieving very similar performance to morphological segmentation in the Arabic-to-English direction and slightly worse in the other direction. Character-based methods, in comparison, perform better on the task of POS tagging, reaching an accuracy of 95.9%, only 1.3% worse than morphological segmentation. We also analyze the effect of segmentation granularity of Arabic on the quality of MT. We observed that a neural MT (NMT) system is sensitive to source/target token ratio and performs best when this ratio is close to or greater than 1.
2 Segmentation Approaches

We experimented with three data-driven segmentation schemes: i) morphological segmentation, ii) sub-word segmentation based on BPE, and iii) two variants of character-based segmentation. We first map each source word to its corresponding segments (depending on the segmentation scheme), embed all segments of a word in vector space and feed them one-by-one to an encoder-decoder model. See Figure 1 for illustration.

2.1 Morphological Segmentation

There is a vast amount of work on statistical segmentation for Arabic. Here we use the state-of-the-art Arabic segmenter MADAMIRA and Farasa as our baselines. MADAMIRA involves a morphological analyzer that generates a list of possible word-level analyses (independent of context). The analyses are provided with the original text to a Feature Modeling component that applies an SVM and a language model to make predictions, which are scored by an Analysis Ranking component. Farasa on the other hand is a light weight segmenter, which ignores context and instead uses a variety of features and lexicons for segmentation.

2.2 Data Driven Sub-word Units

A number of data-driven approaches have been proposed that learn to segment words into smaller units from data (Demberg, 2007; Sami Virpioja and Kurimo, 2013) and shown to improve phrase-based MT (Fishel and Kirik, 2010; Stallard et al., 2012). Recently, with the advent of neural MT, a few sub-word-based techniques have been proposed that segment words into smaller units to tackle the limited vocabulary and unknown word problems (Sennrich et al., 2016; Wu et al., 2016). In this work, we explore Byte-Pair Encoding (BPE), a data compression algorithm (Gage, 1994) as an alternative to morphological segmentation of Arabic. BPE splits words into symbols (a sequence of characters) and then iteratively replaces the most frequent symbols with their merged variants. In essence, frequent character n-gram sequences will be merged to form one symbol. The number of merge operations is controlled by a hyper-parameter $O_P$ which directly affects the granularity of segmentation: a high value of $O_P$ means coarse segmentation and a low value means fine-grained segmentation.

2.3 Character-level Encoding

Character-based models have been found to be effective in translating closely related language pairs (Durrani et al., 2010; Nakov and Tiedemann, 2012) and OOV words (Durrani et al., 2014). Ling et al. (2016) used character embeddings to address the OOV word problem. We explored them as an alternative to morphological segmentation. Their advantage is that character embeddings do not require any complicated pre- and post-processing step other than segmenting words into characters. The fully character-level encoder treats the source sentence as a sequence of letters, encoding each letter (including white-space) in the LSTM encoder (see Figure 1). The decoding may follow identical settings. We restricted the character-level representation to the Arabic side of the parallel corpus and use words for the English side.

Character-CNN Kim et al. (2016) presented a neural language model that takes character-level input and learns word embeddings using a CNN over characters. The embedding are then provided to the encoder as input. The intuition is that the character-based word embedding should be able to learn the morphological phenomena a word inherits. Compared to fully character-level encoding, the encoder gets word-level embeddings as in the case of unsegmented words (see Figure 1). However, the word embedding is intuitively richer than the embedding learned over unsegmented words because of the convolution over characters. The method was previously shown to help neural MT (Belinkov and Glass, 2016; Costa-jussà and Fonollosa, 2016). Belinkov et al. (2017) also showed character-based representations learned using a CNN to be superior, at learning word morphology, than their word-based counter-parts. However, they did not compare these against BPE-based segmentation. We use character-CNN to aid Arabic word segmentation.
### 3 Experiments

In the following, we describe the data and system settings and later present the results of machine translation and POS tagging.

**3.1 Settings**

**Data** The MT systems were trained on 1.2 Million sentences, a concatenation of TED corpus (Cettolo et al., 2012), LDC NEWS data, QED (Guzmán et al., 2013) and an MML-filtered (Axelrod et al., 2011) UN corpus.\(^1\) We used dev+test10 for tuning and tst11-14 for testing. For English-Arabic, outputs were detokenized using MADA detokenizer. Before scoring the output, we normalized them and reference translations using the QCRI normalizer (Sajjad et al., 2013).

**POS tagging** We used parts 2-3 (v3.1-2) of the Arabic Treebank (Mohamed Maamouri, 2010). The data consists of 18268 sentences (483,909 words). We used 80% for training, 5% for development and the remaining for test.

**Segmentation** MADAMIRA and Farasa normalize the data before segmentation. In order to have consistent data, we normalize it for all segmentation approaches. For BPE, we tuned the value of merge operations OP and found 30k and 90k to be optimal for Ar-to-En and En-to-Ar systems respectively. In case of no segmentation (UNSEG) and character-CNN (cCNN), we tokenized the Arabic with the standard Moses tokenizer, which separates punctuation marks. For character-level encoding (CHAR), we preserved word boundaries by replacing space with a special symbol and then separated every character with a space. English-side is tokenized/truecased using Moses scripts.

**Neural MT Settings** We used the seq2seq-attn (Kim, 2016) implementation, with 2 layers of LSTM in the (bidirectional) encoder and the decoder, with a size of 500. We limit the sentence length to 100 for MORPH, UNSEG, BPE, cCNN, and 500 for CHAR systems. The source and target vocabularies are limited to 50k each.

### 3.2 Machine Translation Results

Table 1 presents MT results using various segmentation strategies. Compared to the UNSEG system, the MORPH system\(^2\) improved translation quality by 4.6 and 1.6 BLEU points in Ar-to-En and En-to-Ar systems, respectively. The results also improved by up to 3 BLEU points for cCNN and CHAR systems in the Ar-to-En direction. However, the performance is lower by at least 0.6 BLEU points compared to the MORPH system.

In the En-to-Ar direction, where cCNN and CHAR are applied on the target side, the performance dropped significantly. In the case of CHAR, mapping one source word to many target characters makes it harder for NMT to learn a good model. This is in line with our finding on using a lower value of OP for BPE segmentation (see paragraph Analyzing the effect of OP). Surprisingly, the cCNN system results were inferior to the UNSEG system for En-to-Ar. A possible explanation is that the decoder’s predictions are still done at word level even when using the cCNN model (which encodes the target input during training but not the output). In practice, this can lead to generating unknown words. Indeed, in the Ar-to-En case cCNN significantly reduces the number of unknown words in the test sets, while in the En-to-Ar case the number of unknown words remains roughly the same between UNSEG and cCNN.

The BPE system outperformed all other systems in the Ar-to-En direction and is lower than MORPH by only 0.2 BLEU points in the opposite direction. This shows that machine translation involving the

<table>
<thead>
<tr>
<th># SEG</th>
<th>tst1</th>
<th>tst12</th>
<th>tst13</th>
<th>tst14</th>
<th>AVG.</th>
<th>tst1</th>
<th>tst12</th>
<th>tst13</th>
<th>tst14</th>
<th>AVG.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNSEG</td>
<td>25.7</td>
<td>28.2</td>
<td>27.3</td>
<td>23.9</td>
<td>26.3</td>
<td>15.8</td>
<td>17.1</td>
<td>18.1</td>
<td>15.5</td>
<td>16.6</td>
</tr>
<tr>
<td>MORPH</td>
<td>29.2</td>
<td>33.0</td>
<td>32.9</td>
<td>28.3</td>
<td>30.9</td>
<td>16.5</td>
<td>18.8</td>
<td>20.4</td>
<td>17.2</td>
<td>18.2</td>
</tr>
<tr>
<td>cCNN</td>
<td>29.0</td>
<td>32.0</td>
<td>32.5</td>
<td>28.0</td>
<td>30.3</td>
<td>14.3</td>
<td>12.8</td>
<td>13.6</td>
<td>12.6</td>
<td>13.3</td>
</tr>
<tr>
<td>CHAR</td>
<td>28.8</td>
<td>31.8</td>
<td>32.5</td>
<td>27.8</td>
<td>30.2</td>
<td>15.3</td>
<td>17.1</td>
<td>18.0</td>
<td>15.3</td>
<td>16.4</td>
</tr>
<tr>
<td>BPE</td>
<td>29.7</td>
<td>32.5</td>
<td>33.6</td>
<td>28.4</td>
<td>31.1</td>
<td>17.5</td>
<td>18.0</td>
<td>20.0</td>
<td>16.6</td>
<td>18.0</td>
</tr>
</tbody>
</table>

\(^1\)We used 3.75% as reported to be optimal filtering threshold in (Durrani et al., 2016).

\(^2\)Farasa performed better in the Ar-to-En experiments and MADAMIRA performed better in the En-to-Ar direction. We used best results as our baselines for comparison and call them MORPH.
Arabic language can achieve competitive results with data-driven segmentation. This comes with an additional benefit of language-independent pre-processing and post-processing pipeline. In an attempt to find, whether the gains obtained from data-driven segmentation techniques and morphological segmentation are additive, we applied BPE to morphological segmented data. We saw further improvement of up to 1 BLEU point by using the two segmentations in tandem.

**Analyzing the effect of OP:** The unsegmented training data consists of 23M Arabic tokens and 28M English tokens. The parameter OP decides the granularity of segmentation: a higher value of OP means fewer segments. For example, at \( OP=50k \), the number of Arabic tokens is greater by 7% compared to \( OP=90k \). We tested four different values of OP (15k, 30k, 50k, and 90k). Figure 2 summarizes our findings on test-2011 dataset, where x-axis presents the ratio of source to target language tokens and y-axis shows the BLEU score. The boundary values for segmentation are character-level segmentation \( (OP=0) \) and unsegmented text \( (OP=N) \). For both language directions, we observed that a source to target token ratio close to 1 and greater works best provided that the boundary conditions (unsegmented Arabic and character-level segmentation) are avoided. In the En-to-Ar direction, the system improves for coarse segmentation whereas in the Ar-to-En direction, a much finer-grained segmentation of Arabic performed better. This is in line with the ratio of tokens generated using the MORPH systems (Ar-to-En ratio = 1.02). Generalizing from the perspective of neural MT, the system learns better when total numbers of source and target tokens are close to each other. The system shows better tolerance towards modeling many source words to a few target words compared to the other way around.

**Discussion:** Though BPE performed well for machine translation, there are a few reservations that we would like to discuss here. Since the main goal of the algorithm is to compress data and segmentation comes as a by-product, it often produces different segmentations of a root word when occurred in different morphological forms. For example, the words driven and driving are segmented as driven and driving respectively. This adds ambiguity to the data and may result in unexpected translation errors. Another limitation of BPE is that at test time, it may divide the unknown words to semantically different known sub-word units which can result in a semantically wrong translation. For example, the word “"شطر " is unknown to our vocabulary. BPE segmented it into known units which ended up being translated to courage. One possible solution to this problem is; at test time, BPE is applied to those words only which were known to the full vocabulary of the training corpus. In this way, the sub-word units created by BPE for the word are already seen in a similar context during training and the model has learned to translate them correctly. The downside of this method is that it limits BPE’s power to segment unknown words to their correct sub-word units and outputs them as UNK in translation.

### 3.3 Part of Speech Tagging

We also experimented with the aforementioned segmentation strategies for the task of Arabic POS tagging. Probabilistic taggers like HMM-based (Brants, 2000) and sequence learning models like CRF (Lafferty et al., 2001) consider previous words and/or tags to predict the tag of the current word. We mimic a similar setting in a sequence-to-sequence learning framework. Figure 3 describes a step by step procedure to train a neural encoder-decoder tagger. Consider an Arabic phrase “klm >SdqA\k b$rhm” (gloss: call your friends give them the good news), we want to learn the tag

---

\(^3\) \( N \) is the number of types in the unsegmented corpus.
of the word “بشرهم” using the context of the previous two words and their tags. First, we segment the phrase using a segmentation approach (step 1) and then add POS tags to context words (step 2). The entire sequence with the words and tags is fed to the sequence-to-sequence framework. The embeddings (for both words and tags) are learned jointly with other parameters in an end-to-end fashion, and optimized on the target tag sequence; for example, “NOUN PRON” in this case.

For a given word $w_i$ in a sentence $s = \{w_1, w_2, ..., w_M\}$ and its POS tag $t_i$, We formulate the neural TAGGER as follows:

$$\text{SEGMENTER}(\tau) : \forall w_i \mapsto S_i$$

$$\text{TAGGER} : S_{i-2} S_{i-1} S_i \mapsto t_i$$

where $S_i$ is the segmentation of word $w_i$. In case of UNSEG and cCNN, $S_i$ would be same as $w_i$. SEGMENTER here is identical to the one described in Figure 1. TAGGER is a NMT architecture that learns to predict a POS tag of a segmented/unsegmented word given previous two words.\(^4\)

Table 2 summarizes the results. The MORPH system performed best with an improvement of 5.3% over UNSEG. Among the data-driven methods, CHAR model performed best and was behind MORPH by only 0.3%. Even though BPE was inferior compared to other methods, it was still better than UNSEG by 4%.\(^5\)

**Analysis of POS outputs** We performed a comparative error analysis of predictions made through MORPH, CHAR and BPE based segmentations. MORPH and CHAR observed very similar error patterns, with most confusion between *Foreign* and *Particle* tags. In addition to this confusion, BPE had relatively scattered errors. It had lower precision in predicting nouns and had confused them with adverbs, foreign words and adjectives. This is expected, since most nouns are out-of-vocabulary terms, and therefore get segmented by BPE into smaller, possibly known fragments, which then get confused with other tags. However, since the accuracies are quite close, the overall errors are very few and similar between the various systems. We also analyzed the number of tags that are output by the sequence-to-sequence model using various segmentation schemes. In 99.95% of the cases, the system learned to output the correct number of tags, regardless of the number of source segments.

**4 Conclusion**

We explored several alternatives to language-dependent segmentation of Arabic and evaluated them on the tasks of machine translation and POS tagging. On the machine translation task, BPE segmentation produced the best results and even outperformed the state-of-the-art morphological segmentation in the Arabic-to-English direction. On the POS tagging task, character-based models got closest to using the state-of-the-art segmentation. Our results showed that data-driven segmentation schemes can serve as an alternative to heavily engineered language-dependent tools and achieve very competitive results. In our analysis we showed that NMT performs better when the source to target token ratio is close to one or greater.

**Acknowledgments**

We would like to thank the three anonymous reviewers for their useful suggestions. This research was carried out in collaboration between the HBKU Qatar Computing Research Institute (QCRI) and the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL).
References


