First Result on Arabic Neural Machine Translation

Amjad Almahairi
MILA, Université de Montréal
amjad.almahairi@umontreal.ca

Kyunghyun Cho
New York University
kyunghyun.cho@nyu.edu

Nizar Habash
New York University
nizar.habash@nyu.edu

Aaron Courville
MILA, Université de Montréal
aaron.courville@umontreal.ca

Abstract

Neural machine translation has become a major alternative to widely used phrase-based statistical machine translation. We notice however that much of research on neural machine translation has focused on European languages despite its language agnostic nature. In this paper, we apply neural machine translation to the task of Arabic translation (Ar↔En) and compare it against a standard phrase-based translation system. We run extensive comparison using various configurations in preprocessing Arabic script and show that the phrase-based and neural translation systems perform comparably to each other and that proper preprocessing of Arabic script has a similar effect on both of the systems. We however observe that the neural machine translation significantly outperform the phrase-based system on an out-of-domain test set, making it attractive for real-world deployment.

1 Introduction

Neural machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014) has become a major alternative to the widely used statistical phrase-based translation system (Koehn et al., 2003), evidenced by the successful entries in WMT'15 and WMT'16.

Previous work on using neural networks for Arabic translation has mainly focused on using neural networks to induce an additional feature for phrase-based statistical machine translation systems (see, e.g., (Devlin et al., 2014; Setiawan et al., 2015)). This hybrid approach has resulted in impressive improvement over other systems without any neural network, which raises a hope that a fully neural translation system may achieve a even higher translation quality. We however found no prior work on applying a fully neural translation system (i.e., neural machine translation) to Arabic translation.

In this paper, our aim is therefore to present the first result on the Arabic translation using neural machine translation. On both directions (Ar→En and En→Ar), we extensively compare a vanilla attention-based neural machine translation system (Bahdanau et al., 2015) against a vanilla phrase-based system (Moses, (Koehn et al., 2003)), while varying pre-/post-processing routines, including morphology-aware tokenization and orthographic normalization, which were found to be crucial in Arabic translation (Habash and Sadat, 2006; Badr et al., 2008; El Kholy and Habash, 2012).

The experiment reveals that neural machine translation performs comparably to the standard phrase-based system. We further observe that the tokenization and normalization routines, initially proposed for phrase-based systems, equally improve the translation quality of neural machine translation. Finally, on the En→Ar task, we find the neural translation system to be more robust to the domain shift compared to the phrase-based system.

2 Neural Machine Translation

A major workforce behind neural machine translation is an attention-based encoder-decoder model (Bahdanau et al., 2015; Cho et al., 2015). This attention-based encoder-decoder model consists of an encoder, decoder and attention mechanism. The encoder, which is often implemented as a bidirectional recurrent network, reads a source sentence $X = (x_1, \ldots, x_{T_x})$ and returns a set of context vectors $C = (\mathbf{h}_1, \ldots, \mathbf{h}_{T_x})$.

The decoder is a recurrent language model. At each time t', it computes the new hidden state by

$$\mathbf{z}_{t'} = \phi(\mathbf{z}_{t'-1}, \tilde{y}_{t'-1}, \mathbf{c}_{t'}),$$

where ϕ is a recurrent activation function, and $\mathbf{z}_{t'-1}$ and $\tilde{y}_{t'-1}$ are the previous hidden state and previously decoded target word respectively. $\mathbf{c}_{t'}$ is a time-dependent context vector and is a weighted sum of the context vectors returned by the encoder: $\mathbf{c}_{t'} = \sum_{t=1}^{T_x} \alpha_t \mathbf{h}_t$, where the attention weight α_t is computed by the attention mechanism f_{att}: $\alpha_t \propto \exp(f_{\text{att}}(\mathbf{z}_{t'-1}, \tilde{y}_{t'-1}, \mathbf{h}_t))$. In this paper, we use a feedforward network with a single $\tanh$ hidden layers to implement f_{att}.

Given a new decoder state $\mathbf{z}_{t'}$, the conditional distribution over the next target symbol is computed as

$$p(y_t = w | \tilde{y}_{<t}, X) \propto \exp(g_w(\mathbf{z}_{t'})),$$

where g_w returns a score for the word w, and V is a target vocabulary.

The entire model, including the encoder, decoder and attention mechanism, is jointly tuned to maximize the conditional log-probability of a ground-truth translation given a source sentence using a training corpus of parallel sentence pairs. This learning process is efficiently done by stochastic gradient descent with backpropagation.

Subword Symbols Sennrich et al. (2015), Chung et al. (2016) and Luong and Manning (2016) showed that the attention-based neural translation model can perform well when source and target sentences are represented as sequences of subword symbols such as characters or frequent character n-grams. This use of subword symbols elegantly addresses the issue of large target vocabulary in neural networks (Jean et al., 2014), and has become a *de facto* standard in neural machine translation. Therefore, in our experiments, we use character n-grams selected by byte pair encoding (Sennrich et al., 2015).

3 Processing of Arabic for Translation

3.1 Characteristics of Arabic Language

Arabic exhibits a rich morphology. This makes Arabic challenging for natural language processing and machine translation. For instance, a single Arabic token ‘ولمركبته’ (‘and to his vehicle’ in English) is formed by prepending ‘و’ (‘and’) and ‘لـ’ (‘to’) to the base lexeme ‘مركبة’ (‘vehicle’), appending ‘ه’ (‘his’) and replacing the feminine suffix ‘ ة ’ (*ta marbuta*) of the base lexeme to ‘ ت ’. This feature of Arabic is challenging, as (1) it increases the number of out-of-vocabulary tokens, (2) it consequently worsens the issue of data sparsity [1], and (3) it complicates the word-level correspondence between Arabic and another language in translation. This is often worsened by the orthographic ambiguity found in Arabic scripts, such as the inconsistency in spelling certain letters.

Previous work has thus proposed morphology-aware tokenization and orthographic normalization as two crucial components for building a high quality phrase-based machine translation system (or its variants) for Arabic (Habash and Sadat, 2006; Badr et al., 2008; El Kholy and Habash, 2012). These techniques have been found very effective in alleviating the issue of data sparsity and improving the generalization to tokens not included in a training corpus (in their original forms.)

3.2 Morphology-Aware Tokenization

The goal of morphology-aware tokenization, or morpheme segmentation (Creutz and Lagus, 2005) is to split a word in its surface form into a sequence of linguistically sound sub-units. Contrary to simple string-based tokenization methods, morphology-aware tokenization relies on linguistic knowledge of a target language (Arabic in our case) and applies, for instance, various morphological or orthographic adjustments to the resulting sub-units.

In this paper, we investigate the tokenization scheme used in the Penn Arabic Treebank (ATB, (Maamouri et al., 2004)) which was found to work well with phrase-based translation system in (El Kholy and Habash, 2012). This tokenization separates all clitics other than definite articles.

When translating *to* Arabic, the decoded sequence of tokenized symbols must be *de-tokenized*. This de-tokenization step is not trivial, as it needs to undo any adjustment (implicitly) made by the tokenization algorithm. In this work, we follow the approach

[1] see Sec. 5.2.1 of (Cho, 2015) for detailed discussion.

proposed in (Badr et al., 2008; Salameh et al., 2015). This approach builds a lookup table from a training corpus and uses it for mapping a tokenized form back to its original form. When the tokenized form is missing in the lookup table, we back off to a number of hand-crafted de-tokenization rules.

3.3 Orthographic Normalization

Since the sources of major orthographic ambiguity are in the letters 'alif' and 'ya', we normalize these letters (and their inconsistent replacements.) Furthermore, we replace parentheses '(' and ')' with special tokens '–LRB–' and '–RRB–', and remove diacritics.

4 Experimental Settings

4.1 Data Preparation

Training Corpus We combine LDC2004T18, LDC2004T17 and LDC2007T08 to form a training parallel corpus. The combined corpus contains approximately 1.2M sentence pairs, with 33m tokens on the Arabic side. Most of the sentences are from news articles. We ignore sentence pairs which either side has more than 100 tokens.

In-Domain Evaluation Sets We use the evaluation sets from NIST 2004 (MT04) and 2005 (MT05) as development and test sets respectively. In Ar→En, we use all four English reference translations to measure the translation quality. We use only the first English sentence out of four as a source during En→Ar. Both of these sets are derived from news articles, just as the training corpus is.

Out-of-Domain Evaluation Set In the case of En→Ar, we evaluate both phrase-based and neural translation systems on MEDAR evaluation set (Hamon and Choukri, 2011). This set has four Arabic references per English sentence. It is derived from web pages discussing climate changes, significantly differing from the training corpus. This set was selected to highlight the robustness to domain mismatch between training and test sets.

We verify the domain mismatches of the evaluation sets relative to the training corpus by fitting a 5-gram language model on the training corpus and computing the likelihoods of the evaluation sets, on the Arabic side. As can be seen in Table 1, the domain of the MEDAR is significantly further away from the training corpus than the others are.

	MT04	MT05	MEDAR
Avg. Log-Prob.	-59.74	-55.97	-75.03

Table 1: Language model scores of the Arabic side. The language model was tuned on the training corpus.

Note on MT04 and MT05 We noticed that a significant portion of Arabic sentences in MT04 and MT05 are found verbatim in the training corpus (172 on MT04 and 26 on MT05). In order to accurately measure the generalization performance, we removed those duplicates from the evaluation sets.

4.2 Machine Translation Systems

Phrase-based Machine Translation We use Moses (Koehn et al., 2007) to build a standard phrase-based statistical machine translation system. Word alignment was extracted by GIZA++ (Och and Ney, 2003), and we used phrases up to 8 words to build a phrase table. We use the following options for alignment symmetrization and reordering model: *grow-diag-final-and* and *msd-bidirectional-fe*. KenLM (Heafield et al., 2013) is used as a language model and trained on the target side of the training corpus.

Neural machine translation We use a publicly available implementation of attention-based neural machine translation.[2] For both directions–En→Ar and Ar→En–, the encoder is a bidirectional recurrent network with two layers of 512×2 gated recurrent units (GRU, (Cho et al., 2014)), and the decoder a unidirectional recurrent network with 512 GRU's. Each model is trained for approximately seven days using Adadelta (Zeiler, 2012) until the cost on the development set stops improving. We regularize each model by applying dropout (Srivastava et al., 2014) to the output layer and penalizing the L2 norm of the parameters (coefficient 10^{-4}). We use beam search with width set to 12 for decoding.

4.3 Normalization and Tokenization

Arabic We test *simple tokenization* (**Tok**) based on the script from Moses, and orthographic *normalization* (**Norm**), and *morphology-aware tokenization* (**ATB**) using MADAMIRA (Pasha et al., 2014), . In the latter scenario, we reverse the tokenization before computing BLEU. Note that **ATB** includes

[2] https://github.com/nyu-dl/dl4mt-tutorial

	Arabic			English		En→Ar				Ar→En	
	Tok.	Norm.	ATB	Tok.	Lower	MT05		MEDAR		MT05	
PB-SMT	√			√		31.52	–	8.69	–	48.59	–
	√			√	√	33.03	(1.51)	9.78	(1.09)	49.44	(0.85)
	√	√		√		34.98	(3.46)	17.34	(8.65)	49.51	(0.92)
	√	√		√	√	35.63	(4.11)	17.75	(9.06)	49.91	(1.32)
	√	√	√	√		35.7	(4.18)	18.67	(9.98)	50.67	(2.08)
	√	√	√	√	√	35.98	(4.46)	19.27	(10.58)	51.19	(2.60)
Neural MT	√			√		28.64	–	11.09	–	47.12	–
	√			√	√	29.77	(1.13)	10.15	(-0.94)	47.63	(0.51)
	√	√		√		32.53	(3.89)	22.36	(11.27)	48.53	(1.41)
	√	√		√	√	32.95	(4.31)	22.79	(11.70)	47.53	(0.41)
	√	√	√	√		33.53	(4.89)	23.11	(12.02)	49.21	(2.09)
	√	√	√	√	√	33.62	(4.98)	24.46	(13.37)	49.7	(2.58)

Table 2: BLEU scores with the improvement over the tokenization-only models in the parentheses.

Norm, and both of them include simple tokenization performed by MADAMIRA.

English We test *simple tokenization* (**Tok**), lowercasing (**Lower**) for En→Ar and *truecasing* (**True**, (Lita et al., 2003)) for Ar→En.

Byte pair encoding As mentioned earlier in Sec. 2, we use byte pair encoding (BPE) for neural machine translation. We apply BPE to the already-tokenized training corpus to extract a vocabulary of up to 20k subword symbols. We use the publicly available script released by Sennrich et al. (2015).

5 Result and Analysis

En→Ar From Table 2, we observe that the translation quality improves as a better preprocessing routine is used. By using the normalization as well as morphology-aware tokenization (Tok+Norm+ATB), the phrase-based and neural systems each achieve as much as +4.46 and +4.98 BLEU over the baselines, on MT05. The improvement is even more apparent on the MEDAR whose domain deviates from the training corpus, confirming that proper preprocessing of Arabic script indeed helps in handling word tokens that are not present in a training corpus.

We notice that the tested tokenization strategies have nearly identical effect on both the phrase-based and neural translation systems. The translation quality of either system is mostly effective by the tokenization strategy employed for Arabic, and is largely insensitive to whether source sentences in English were lowercased. This reflects well the complexity of Arabic scripts, compared to English, discussed earlier in Sec. 3.1.

Another important observation is that the neural translation system significantly outperforms the phrase-based one on the out-of-domain evaluation set (MEDAR), while they perform comparably to each other in the case of the in-domain evaluation set (MT05). We conjecture that this is due to the neural translation system's superior generalization capability based on its use of continuous space representations.

Ar→En In the last column of Table 2, we observe a trend similar to that from En→Ar. First, both phrase-based and neural machine translation benefit quite significantly from properly normalizing and tokenizing Arabic, while they are both less sensitive to truecasing English. The best translation quality using either model was achieved when all the tokenization methods were applied (Ar: Tok+Norm+ATB and En:Tok+True), improving upon the baseline by more than 2+ BLEU. Furthermore, we again observe that the phrase-based and neural translation systems perform comparably to each other.

6 Conclusion

We have provided first results on Arabic neural MT, and performed extensive experiments comparing it with a standard phrase-based system. We have concluded that neural MT benefits from morphology-based tokenization and is robust to domain change.

References

[Badr et al.2008] Ibrahim Badr, Rabih Zbib, and James Glass. 2008. Segmentation for english-to-arabic statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 153–156. Association for Computational Linguistics.

[Bahdanau et al.2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

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

[Cho et al.2015] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. *Multimedia, IEEE Transactions on*, 17(11):1875–1886.

[Cho2015] Kyunghyun Cho. 2015. Natural language understanding with distributed representation. *arXiv:1511.07916*.

[Chung et al.2016] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *ACL*.

[Creutz and Lagus2005] Mathias Creutz and Krista Lagus. 2005. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology.

[Devlin et al.2014] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *ACL*.

[El Kholy and Habash2012] Ahmed El Kholy and Nizar Habash. 2012. Orthographic and morphological processing for english–arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

[Habash and Sadat2006] Nizar Habash and Fatiha Sadat. 2006. Arabic preprocessing schemes for statistical machine translation.

[Hamon and Choukri2011] Olivier Hamon and Khalid Choukri. 2011. Evaluation methodology and results for english-to-arabic mt. *Proceedings of MT Summit XIII*, pages 480–487.

[Heafield et al.2013] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

[Jean et al.2014] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. In *ACL 2015*.

[Kalchbrenner and Blunsom2013] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

[Koehn et al.2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

[Lita et al.2003] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics.

[Luong and Manning2016] Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv:1604.00788*.

[Maamouri et al.2004] Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467.

[Och and Ney2003] Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

[Pasha et al.2014] Arfath Pasha, Mohamed Al-Badrashiny, Mona T 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 *LREC*, pages 1094–1101.

[Salameh et al.2015] Mohammad Salameh, Colin Cherry, and Grzegorz Kondrak. 2015. What matters most in morphologically segmented smt models. *Syntax, Semantics and Structure in Statistical Translation*, page 65.

[Sennrich et al.2015] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[Setiawan et al.2015] Hendra Setiawan, Zhongqiang Huang, Jacob Devlin, Thomas Lamar, Rabih Zbib, Richard Schwartz, and John Makhoul. 2015. Statistical machine translation features with multitask tensor networks. *arXiv:1506.00698*.

[Srivastava et al.2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

[Zeiler2012] Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.