Arabic-English Neural Machine Translation

**Mohmmad Anas Koara**

Syrian Virtual university

Mohmmad\_anas\_65804@svuonline.org

**Dr. Nada Ghneim**

Syrian Virtual university

nada.ghneim@gmail.com

Abstract

Neural machine translation NMT is a modern approach for machine translation. Contrary to statistical machine translation, which consists of many components, NMT consists of single neural networks, which is more simple to train, NMT gained a lot of interest from researchers. In this paper wee experimented to apply NMT for Arabic↔English machine translation, we experimented basic encoder-decoder architecture and used attention mechanism and to improve translation quality, we also investigated the effect of preprocessing the of Arabic sentences before befoe applying it on the neural network. We get fairly good results from basic encoder-decoder architecture, and translation quality improved significantly when we used the Attention mechanism. But for our surprise using preprocessing on the Arabic sentence has worsened the result and did not give a good quality translation.

# Introduction

Neural machine translation NMT is a novel approach for machine translation, proposed by [1], [2]. Unlike the traditional phrase-based translation systems like [3]which consists of many small sub-components,need to be tuned to give the best translation, neural machine translation based on the idea of one building big neural networks that reads the source sentence and produce translation For build and train a single, large neural network that reads a sentence and outputs a correct translation, Most neural machine translation based Most of the neural machine translation models based on the encoder-decoder architecture s [2] [4], where the encoder read the source sentence and produce fixed-length vector representation to this sentence, and the decoder use this vector representation and produces translation to input sentence. The main advantage of encoder-decoder models that the encoder and decoder are trained together. Encode-decoder models are trained to maximize the probability of correct translation given source sentence. This architecture was very good results as showed by [1] [2]. But one drawback of this approach is that a neural network should fixed-length vector. This seems to be difficult especially if the sentence is very long, [4] showed the performance of the translation decreases rapidly as the length of an input sentence increases. To deal with this problem [5] Introduced new extension to encoder-decoder architecture, and followed by [6]. This extension is called Attention mechanism, which allows aligning words in translation to its corresponding word int the source sentence, the most important difference of this approach from the basic encoder-decoder is that it does not compress the input sentence into a single fixed-length vector. Studies have shown using attention mechanism has a huge improvement in NMT models output.

Another major issue with NMT is, that it operates of fixed vocabulary, and translation quality degrades significantly when input sentence contains rare or new words, To solve this problem [7] used Alignment function in attention-based models with the help of external dictionary to find unknown words in the output. [8] Introduced a new approach that works on a sub-word level, using Byte pair encoding [9] compression to segment word into smaller units.

In the Paper we exterminated These approaches on Arabic↔English Machine translation

# Neural Machine Translation

Neural Machine Translation NMT, since its introduction by (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014) has been proven to be the main choice for machine translation, it outperforms phrase-based statistical machine translation system.

In this paper, we Applied Neural machine translation for Arabic To English Translation and from English To Arabic.

One can consider the NMT System as a function to compute the conditional distribution

for target Sentence is translation for the sentence S=s1,...,sn

This conditional probability can be written as

NMT System computes later conditional probability, as multiplication for every position in the target sentence, the conditional probability of target word given the previous words ins the target and source sentence S.

early studies of neural machine translation suggested to use bilingual, parallel corpus(Cho et al., 2014; Sutskever et al.,2014), and looking in the details of these studies the relay on encoder-decoder architecture. [5] Introduced new Improvement by using the Attention mechanism to improve the quality of the translation we discuss two architecture Briefly in the flowing sections.

## Encoder-Decoder

In this architecture, the encoder reads the input sentence , and generate vector representation C for the input sentence. Usually, the encoder is made of one or multilayer RNN such as:

Where is RNN hidden state at time step t.

The decoder is trained to predict the next word , given the vector c and all previously predicted words

In this case, the decoder computes the probability of the translation Y as Multiplication of every word conditional probability

And every conditionl probability can be represented

|c,

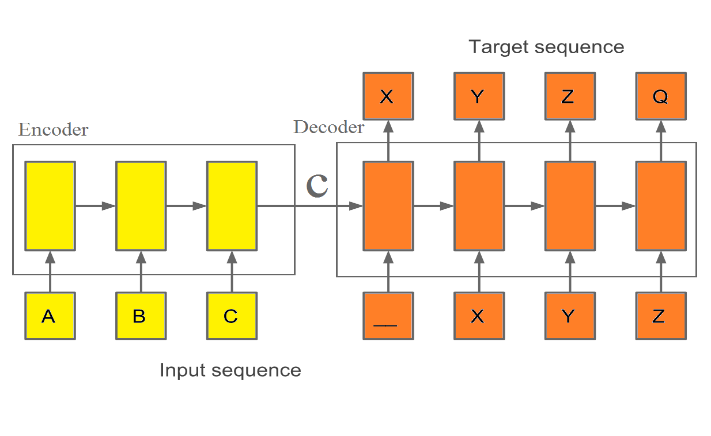
Where g is multi-layered RNN, is the hidden state of the RNN.

Figure 1 Encoder-Decocer NMT that translates {A,B,C } to {X,Y,Z,Q}

## Attention-based model

[5] notice the translation quality degrade significantly with the increase of the input sentence length, to solve this issue [5] add new extension to existing encoder-decoder architecture, the main advantages for this approach that it does not compress the input sentence in single vector, when the model generate word in the translation it search in the input sentence where most information is concentrated, and model generates word-based context vectors related to the source position and previously generated words.

### Encoder

In this model [5] used bidirectional RNN [10] For the encoder, BiRNN consists of forward and backward RNN. The forward RNN read input sequence normally from to, and calculates a sequence of forward hidden states

The forward RNN reads input sequence normally from to and calculates a sequence of backward hidden states.

And then the forward and backward hidden states are concatenated together to get annotation for each word

And so contains information about proceeding and following words around the word .

### Decoder

The decoder calculates the conditional probability for each word of the translation as:

Where is RNN hidden state at time , is the context vector for the target word . You should notice unlike encoder-decoder every word has distinct context vector . The context vector depends on input annotation which is a map of the input sequence, from the previous section we know, every annotation contains information about the whole sentence and concentrated around position

The context vector is computed from as weighed sum for input annotation

Where the weight for each input annotation is computed by

Where is alignment function and score how the input around position and the output around position matches.

This score is based on RNN hidden state and the input annotation .

Alignment function is modeled as feedforward neural network. and can be trained with other components of the model.

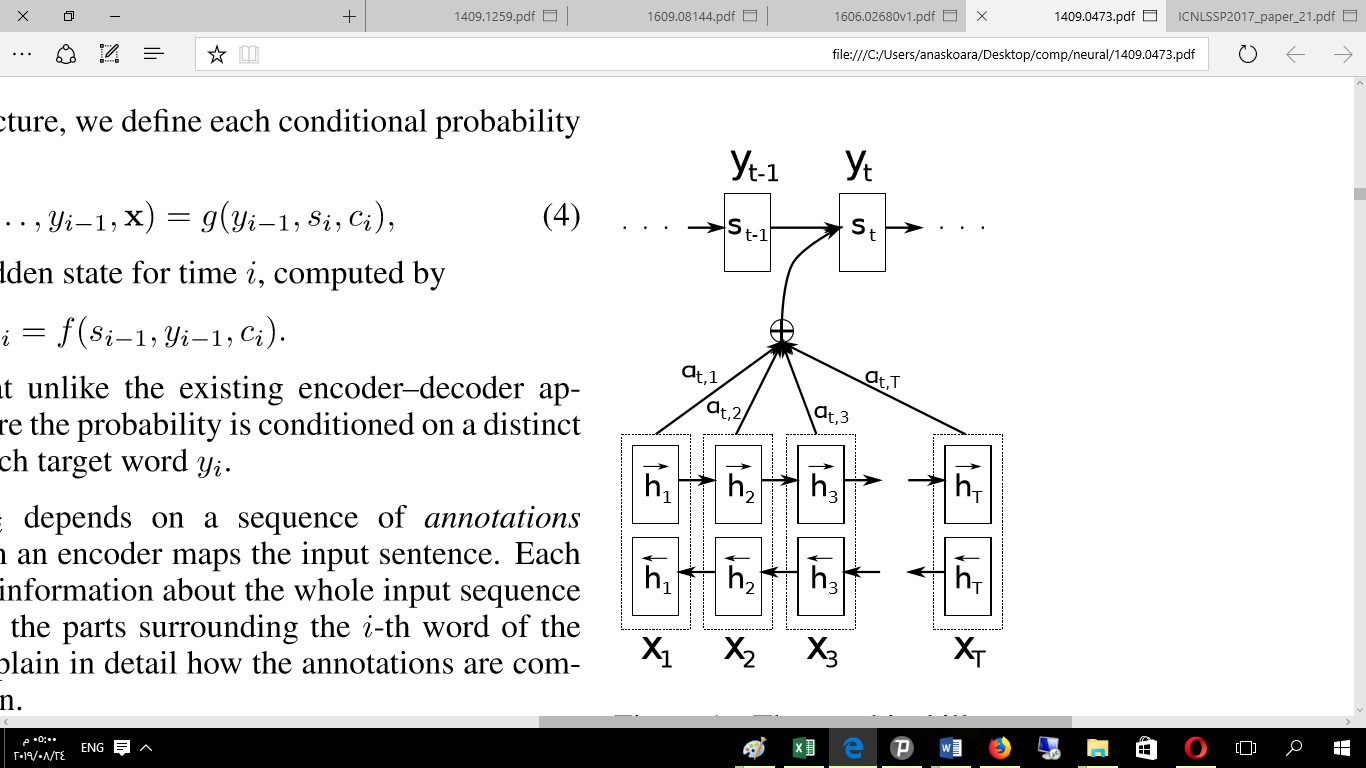


Figure 2 NMT with attention predicting the t-th word in the output [5]

# Experiment Settings

## Dataset

For all the models we build we used subset from Arabic-English United Nations Parallel Corpus UNCorpus, we randomly select 1M Pairs as a training set, 300k Pairs for validation set and 70K for the test set, we have chosen pairs that their sentences are 5-30 tokens long, pairs containing numbers were excluded.

We selected 50000 most frequently words as vocabulary for the Arabic language, 23k most frequently words as vocabulary for English, the least repetitive words were replaced by UNK.

## Models

We experimented three Different Models for Arabic->english, english->Arabic translation, these Models are:

* **Enc-Dec**: based Encoder-Decoder architecture as proposed by [1].
* **Attention**: Encoder-Decoder with attention architecture proposed by [2].
* **Attention\_Subword** Encoder-Decoder with attention on subword level proposed by [3]**.**

All the models we built has 2 bidirectional layers with 128 LSTM units for the encoder and four layers of 128 LSTM units [4] for the decoder.

For first two models we did not do and special processing on the data, and for the last model we used FARASA [12] for Arabic words segmentation, and then we Applied Byte Pair encoding [3] on both languages to limit our vocabulary to 30K subword symbol for both languages.

4.3 Handling unknown and rare words

To handle <unk> symbol on the output of the neural network. we simply left the symbol in the **Enc-Dec** model output without any special preprocessing.

And for the **Attention** model we used technique described by [5] that uses alignment function, to map the <unk> symbol in the output to its corresponding word to the output, and and get it translation with help of external dictionary . while our third model relied on the technique described by [3] to work on subword level to make model able to translate unknown and rare words

## Training

For all model weights were randomly initialized in the range [-0.05,0.05], using learning rate 0.5 and decreasing it gradually, models were trained using stochastic gradient descent and used Dropout [6] with rate 0.2 to for Regularization and prevent overfitting, training is done using Nvidia GTX 1050 ti with 4GB of Ram. Each model is trained for 1 million steps, enc-dec training took 5 days to complete, 7 days for attention model, while attention\_subword required 1o days to complete training. We used BLEU scores [7] to measure translation quality.

While using beam search decoder [8] can increase our translation models quality. we used greedy search decoder, because of our limited computational resources, and for the same reason, one can notice that our models are relatively small in comparing to other studies [1] [2].

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | **5-10** | **11-15** | **16-20** | **20-25** | **>25** | **All Dataset** |
| **Enc-Dec** | 22.13 | 21.18 | 20.17 | 18.14 | 17.52 | 20.19 |
| **Attention** | 28.88 | 28.47 | 27.49 | 28.45 | 27.66 | 27.45 |
| **Attention with unk replacement** | 28.45 | 29.55 | 28.97 | 29.87 | 29.45 | 28.41 |
| **Attention\_Subword** | 12.05 | 10.25 | 8.26 | 9.15 | 8.39 | 9.47 |

Table 1 Our model English To Arabic translation result for sentence length using BLEU Score

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | **5-10** | **11-15** | **16-20** | **20-25** | **>25** | **الكلي** |
| **Model** | 26.21 | 24.05 | 24.01 | 22.43 | 21.25 | 22.57 |
| **Enc-Dec** | 37.21 | 35.05 | 34.98 | 35.15 | 35.07 | 35.4 |
| **Attention** | 33.21 | 37.05 | 33.23 | 36.12 | 35.47 | 35.69 |
| **Attention with unk replacement** | 17.21 | 15.75 | 12.25 | 13.15 | 12.17 | 14.06 |

Table 2 Our model Arabic To English translation result for sentence length using BLEU Score

# Results and analysis

We tested our models on different sentence lengths as we can see from the tables 1&2 that translation quality degraded with the increase of sentence length.

For all models we notice the translation From Arabic to English has better results than translation from Arabic to English, this indicates the models cannot generalize well for the Arabic language, and this my reflects the complexity of Arabic sentences compared to Englis.

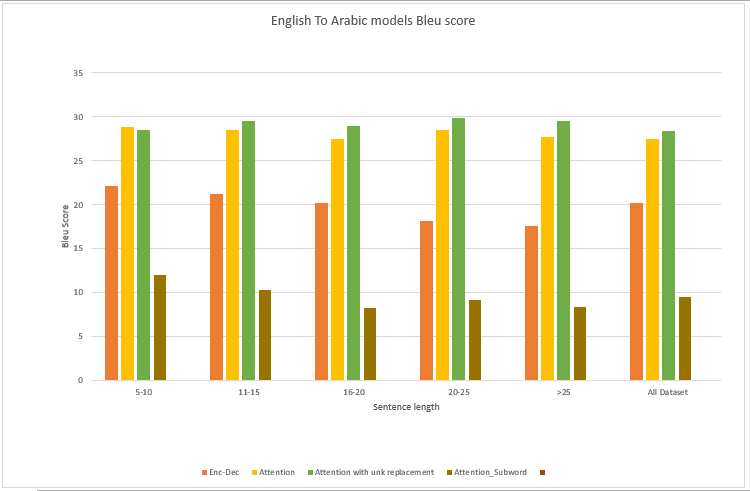
One can notice from the Attention model, the use of alignment function with help of external dictionary improved BLEU score by **0.65** from Arabic to English translation and **0.94** for English to Arabic translation.

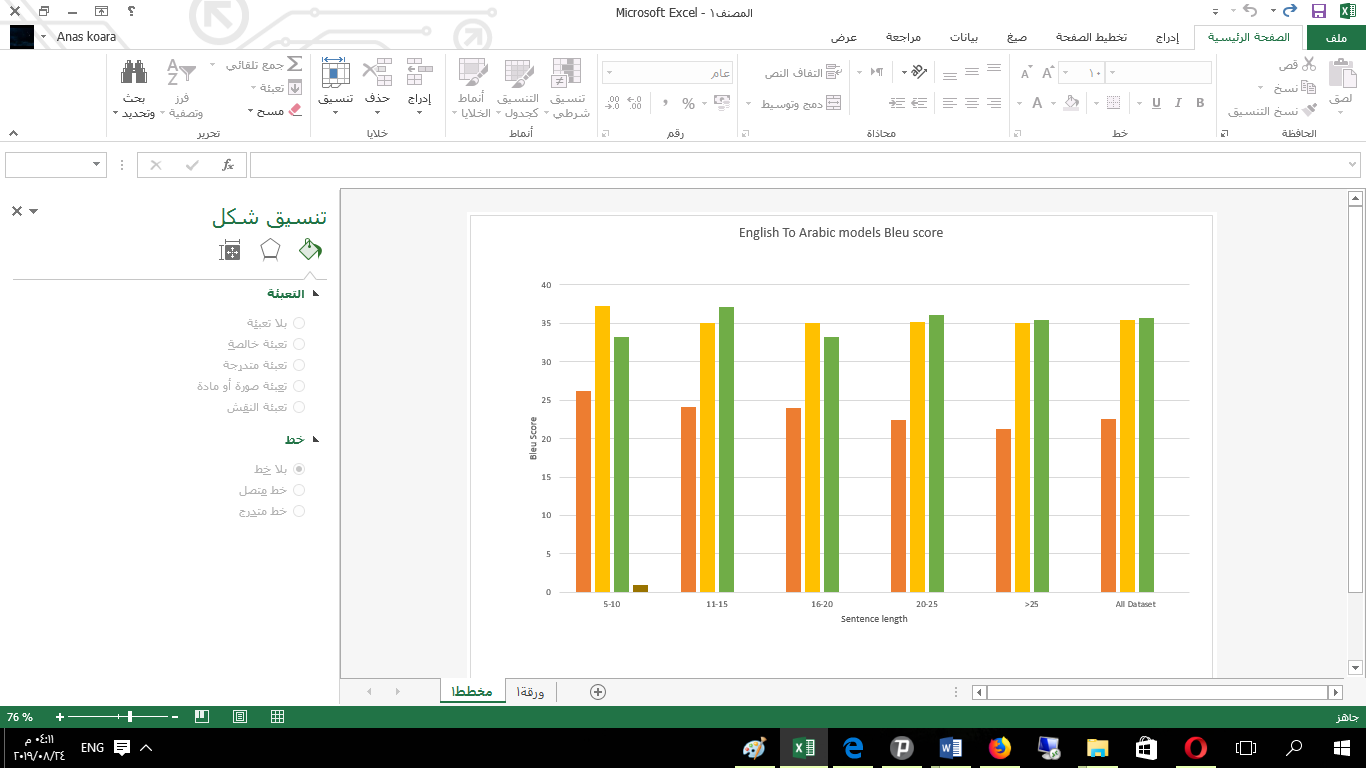
From model three we noticed the that used technique worsened the results and did not help at all which contradicts with other studies , further investigation showed that FARASA segmenter did not give good results, and another reason is because our models and training data is relatively small and model could not learn to generalize well, and this problem needs more investigation.Conclusion

Neural machine translation novel approach that uses single neural network and deep learning to translate sentence from source language to target language and it outperformed Standard statical machine translation, we used it on the problem of Arabic English translation and it showed that basic encoder-decoder can produce produced fairly good translation, and translation quality improved the using attention technique, we experimented different architecture and other approaches to improve translation quality.

These studies concluded the importance of doing more researched on Arabic machine translation, and the importance of building more efficient Arabic.

Our limited resources prevented us to do more experiment and evaluation.





# References

|  |  |  |
| --- | --- | --- |
| [1] |  | N. Kalchbrenner and P. Blunsom, "Recurrent Continuous Translation Models," in *EMNLP*, 2013. |
| [2] |  | I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014. |
| [3] |  | P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *HLT-NAACL*, 2003. |
| [4] |  | K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078,* 2014. |
| [5] |  | D. Bahdanau, K. Cho and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *ICLR,* 2015. |
| [6] |  | L. Minh-Thang, P. Hieu and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *arXiv preprint arXiv:1508.04025,* 2015. |
| [7] |  | T. Luong, I. Sutskever, Q. V. Le, O. Vinyals and W. Zaremba, "Addressing the rare word problem in neural machine translation," *ACL,* 2015. |
| [8] |  | R. Sennrich, B. Haddow and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909,* 2016. |
| [9] |  |  |
| [11] |  | S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation,* pp. 1735-1780, 1997. |
| [12] |  | A. Abdelali, K. Darwish, N. Durrani and H. Mubarak, "FARASA: A fast and furious segmenter for Arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2016. |
| [13] |  | N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overﬁtting," *The Journal of Machine Learning Research,* p. 1929–1958, 2014. |
| [14] |  | K. Papineni, S. Roukos, T. Ward and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002. |
| [15] |  | M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," *arXiv preprint arXiv:1702.01806,* 2017. |
| [16] |  | A. Almahairi, K. Cho, N. Habash and A. Courville, "First result on Arabic neural machine translation," *arXiv,* 2016. |