International Journal on
Advanced Science
Engineering
Information Technology

Vol.8 (2018) No. 4-2
ISSN: 2088-5334

Hybrid Machine Translation with Multi-Source Encoder-Decoder Long Short-Term Memory in English-Malay Translation

Yin-Lai Yeong, *Tien-Ping Tan, Keng Hoon Gan, Siti Khaotijah Mohammad

School of Computer Sciences, Universiti Sains Malaysia, Penang, Mthe alaysia
E-mail: yyl14_com021@student.usm.my, *tienping@usm.my, khgan@usm.my, sitijah@usm.my

Abstract— **Statistical Machine Translation (SMT) and Neural Machine Translation (NMT) are the state-of-the-art approaches in machine translation (MT). The translation produced by an SMT is based on the statistical analysis of text corpora, while NMT uses the deep neural network to model and to generate a translation. SMT and NMT have their strength and weaknesses. SMT may produce a better translation with a small parallel text corpus compared to NMT. Nevertheless, when the amount of parallel text available is large, the quality of the translation produced by NMT is often higher than SMT. Besides that, study also shown that the translation produced by SMT is better than NMT in cases where there is a domain mismatch between training and testing. SMT also has an advantage in long sentences. In addition, when a translation produced by an NMT is wrong, it is very difficult to find the error. In this paper, we investigate a hybrid approach that combines SMT and NMT to perform English to Malay translation. The motivation for using a hybrid machine translation is to combine the strength of both approaches to produce a more accurate translation. Our approach uses the multi-source encoder-decoder long short-term memory (LSTM) architecture. The architecture uses two encoders, one to embed the sentence to be translated, and another encoder to embed the initial translation produced by SMT. The translation from the SMT can be viewed as a "suggestion translation" to the neural MT. Our experiments show that the hybrid MT increases the BLEU scores of our best baseline machine translation in the computer science domain and news domain from 21.21 and 48.35 to 35.97 and 61.81 respectively.**

Keywords— **hybrid machine translation; statistical machine translation; neural machine translation.**

I. Introduction

Machine Translation (MT) is a process of translating text from a source language (for example English) to a target language (for example Malay) using a software. The maturity of the technology allows it to be used by everyone in his or her daily life. Two state-of-the-art MT architectures are statistical machine translation (SMT) and neural machine translation (NMT). One of the most popular NMT uses an encoder-decoder architecture [1]. The translation produced by a SMT is based on the statistical analysis of text corpora.

On the other hand, NMT uses the deep neural network to model and to generate a translation. The elementary unit of translation in SMT is word/phrase, but in NMT it is a vector. NMT uses word embedding to convert word to vector before input to NMT. Both are data-driven approaches that learn from parallel text corpus to build a translation model, but SMT will require an additional text corpus in the target language to build a language model. SMT and NMT have their strength and weaknesses. SMT may produce a better translation with a small parallel text corpus compared to NMT.

Nevertheless, when the amount of parallel text available is large, the quality of the translation produced by NMT is often higher than SMT [2;3]. Besides that, a study [3] also shown that the translation produced by SMT is better than NMT in cases where there is a domain mismatch between training and testing. SMT also has an advantage in long sentences [3]. Also, when an NMT produces erroneous translation, it is tough to troubleshoot because the translation produced is models by an enormous number of parameters in the neural networks. However, on a SMT, the possible translations of a word/phrase can be retrieved from the bilingual phrase table.

There are attempts to combine both SMT and NMT to harvest the strength of both approaches. Since SMT and NMT have different strength in different areas, the idea is that combining both approaches will produce a better translation. Many works show that combining SMT and NMT results in a better MT system. Cho et al. [4] showed that the translation quality of SMT was improved when recurrent neural network (RNN) encoder-decoder was used to calculate the conditional probabilities of phrase pairs as an additional feature in the existing log-linear model. Sutskever et al. [5] used long-short-term memory (LSTM) networks to restore the 1000-best translations produced by an SMT, and this simple approach improves the BLEU score from 33.3 to

36.5. Another study [6] combined SMT and NMT by using SMT features and phrase-based models in NMT search improves MT BLEU score as much as 2.3.

On the other hand, Stahlberg et al. [7] proposed a hybrid SMT and NMT model by minimizing the Bayes-risk where the NMT score is combined with the Bayes-risk of the translation according to the SMT lattice. Besides that, Wang et al. [8] also proposed to incorporate the SMT model into NMT framework where SMT produces a hypothesis, which is used as a suggestion by NMT. Du et al. [9] proposed a cascaded hybrid framework to combine NMT and SMT to improve the translation quality. On the other hand, Dabre et al. [10] used concatenate the source sentences to form a single long multi-source input sentence in multiple languages, Zoph and Knight [11] built a multi-source machine translation model and train to maximize the probability of a target string and Zhang et al. [12] proposed to extend the original encoder-decoder framework to multiple encoders and decoders.

In this paper, we propose to use multi-source encoder-decoder as a hybrid MT architecture in English-Malay translation. The idea of using the multi-source encoder-decoder NMT architecture as the hybrid MT is to use SMT to give translation suggestion to the multi-source encoder-decoder NMT. The multi-source encoder-decoder NMT will learn and model the translation suggestion and original sentence to be translated to produce a better translation. Many studies have shown that deep neural networks can learn and generalize given a vast amount of data.

II. Material and method

A. Statistical machine translation

Statistical Machine Translation (SMT) is one of the computer-based translation approaches that is based on statistical methods introduced since the mid of 20th century. The popularity of SMT is due to the translation architecture that is based on strong mathematical theories [13], good quality translation in many tests run, and the existence of toolkits, which can be used for building MT models within a short time.

SMT takes a source language sentence, S = $s_1, s_{2,} \ldots, s_n$, and generates a target sentence T = $t_1, t_2, \ldots t_n$ in the target language. In a probabilistic model, the best target language sentence, T* is the one whose probability P(T|S) is the highest. The equation will be decomposed by using Bayes theorem as follow:

$$T^* = argmax(P(T|S) \qquad (1)$$

$$T^* = argmax\left(\frac{P(S|T)\times P(T)}{P(S)}\right) \qquad (2)$$

$$T^* = argmax\big(P(S|T) \times P(T)\big) \qquad (3)$$

P(T) is the probability of the target language sentence, and it is evaluated using a language model. The language model for a language is built using a text corpus. The language model stores the statistics for word sequences. Usually, the probability for 3 words sequence, called 3-grams, is used. Hence, a language model functions as a grammar in the form of statistics for SMT.

On the other hand, P(S|T) is the probability of the source language sentence given a target language sentence. A translation model consists of a phrase translation table and a reordering table. The phrase translation table contains phrases and their translations. Each translation is assigned a probability. While the reordering table stores information, regarding the rearrangement of target phrases. A translation model should be built using parallel corpus.

P(T) is the probability of a target language sentence, which is modeled by a language model. The language model for the target language can be built with a target language text corpus. On the other hand, P(S|T) is the probability of a source sentence given the target sentence, which is modeled by a translation model. The model is built using a parallel corpus. Thus, the quality of a machine translation system largely depends on the availability of the large number of resources to build a robust language model and translation model. For low resourced language, the limited amount of these resources will proof building a reasonable good SMT system difficult.

As an example, Fig. 1 shows an English sentence “serve the rice hot” and the translation in Malay. Below each English word or phrase is given the possible “translation,” which are obtained through the alignment of parallel text. Referring to this example, the word “serve” has nine translations in Malay. The translation for each word/phrase and its probability/weight (not shown in the figure). Source and target word pair in the parallel corpus that is frequently aligned together will have a higher weight, P(s_i | t_j). Besides that, there is also a (Malay) n-gram language model weight. The word sequences that are frequently found in the text will have higher weights. For example, the word sequences “*menyajikan nasi*” and “*menghidangkan nasi*,” “*nasi panas*”, “*nasi hangat*” and others will have high (2-grams) weights because these 2 words sequences can be found a lot in Malay texts. On the other hand, 2-grams sequences such as “*bertugas nasi*”, “*padi marah*”, “*nasi seksi*” will have a weight equal to zero or very low because they are rarely found in the text corpus. There are many possible combinations of Malay words that can be formed, 882 in total. The SMT decoder will choose the word combination that the product is the highest when the weights from the translation model and language model are multiplied as the translation. Popular implementations of SMT are Pharaoh [13], which was succeeded by the open source Moses [14]. Phramer [15] is a Java implementation of a phrase-based statistical system and so on. GIZA++ [16] and MTTK [17] are tools for word/phrase alignment.

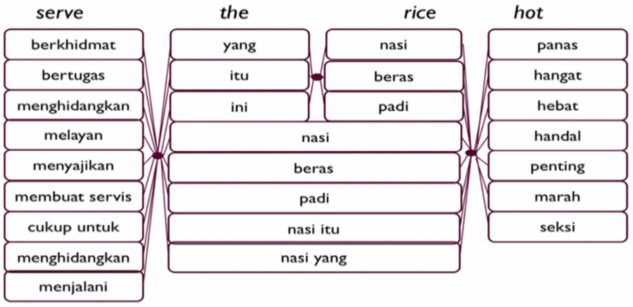

Fig 1. Translating the sentence “serve the rice hot” using SMT.

B. *Neural machine translation*

Artificial neural networks have been applied in many fields, for example, image recognition, face recognition, speech synthesis, muscle simulation [18], etc. The success is due to the introduction and advancement of approaches in convolutional neural networks (CNN) and recurrent neural networks (RNN) that models very complex patterns using deep layers of neural networks. CNN are special feed forward neural networks that allow deep neural networks (DNN) to be built and modeled using local receptive fields, pooling, and weight sharing [19]. CNN is an example that has shown exceptional accuracy in image classification [20]. On the other hand, in sequential pattern modeling such as sentences, weather, video and translation, RNN is the potential to produce excellent results.

Generally, a neural network consists of connected neurons. A basic neuron n_i with input x_j is multiplied with the weight w_{ij} and summed as z (eq 4). The value of z is normalized to a value between 0 and 1 using a logistic function (like tanh function and ReLU function) to get output h (eq 5).

$$z = \Sigma\, w_i x_i \tag{4}$$

$$h = \text{logistics}(z) \tag{5}$$

Typically, more than one neuron is used for modeling and prediction. Additionally, the output of a neuron can be input into the next neuron, and this form a feedforward neural network. See Fig. 2.

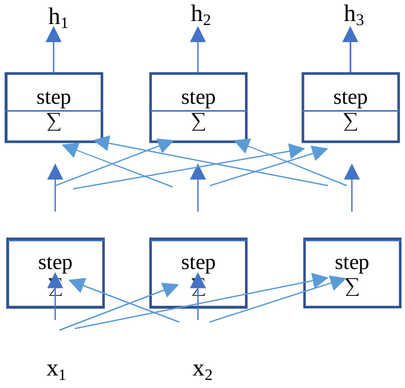

Fig. 2 A two-layered feedforward neural networks

If neural networks are used for classification purposes, output h will be input into a softmax output neuron layer. In general, softmax is a shared logistics function. The modeling of neural network parameters is done by using the back propagation algorithm. In the beginning, the neural network will be initialized with the appropriate values. Then, training data (for example x_1 and x_2) are inserted into the neural network from the outer neural layer, and the predictions (for example h_1, $h_{2,}$ and h_3) produced are compared to the expected values found in the training data. The difference between predictions and expectations in training data will be calculated (usually with mean square errors or cross-entropy) and backpropagated so that the neural network parameters are altered to reduce the difference between forecasts and expectations. This process is repeated until converged.

On the other hand, a recurrent neuron looks like a typical neuron, but it has an additional feedback loop to allow present information to be used for a subsequent neuron to make decisions. RNN can be considered as multiple copies of the same recurrent neurons that convey information to themselves as shown in Fig. 3. There are many variations of the recurrent neuron. Fig. 3 shows a simple recurrent neuron where i_t is inputted and produces output y_t. The output y_i is also feedback to the subsequent neuron. In some other type of recurrent neuron known as a recurrent cell, a state h_i instead of output y_i, which is the function of input x_t and previous state $h_{i\text{-}1}$ is feedback to the next neuron.

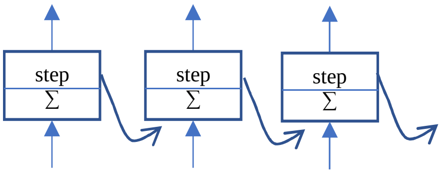

Fig. 3 A layer of recurrent neuron [21]

The RNN encoder-decoder architecture is a widely used architecture in neural machine translation [21]. Fig. 4 shows an example where it is used for decoding. The encoder-decoder architecture can be visualized as two RNN, namely the RNN encoder and RNN decoder. The RNN of the encoder looks like a typical RNN except that the output is ignored. A word-embedding module will convert a word to vector. The vectors are then inputted to the encoder. The RNN of the decoder converts the vector h received from the encoder to words. Notice that the vector of the tag <GO> will be entered as the first input, x_1 into the decoder to initiate the generation of a translation. During testing, the decoder will predict vector y_t. The output vector will be used as an input, x_{t+1} to the next cell in the decoder. This process repeats until the <EOS> tag is generated. On the other hand, during training, the output y_t is ignored and the actual reference r_t is used as input to the decoder instead.

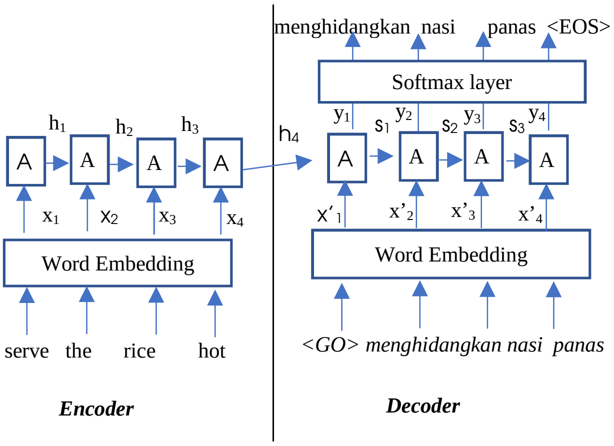

Fig. 4 RNN encoder-decoder architecture [22]

Many advancements have been introduced to this architecture. First, as with other neural network architectures, RNN of the encoders and decoders may consist of more than one layer. Adding more layers of neurons usually allows for better modeling. This can be achieved in two ways. First, an extra layer can be added at the encoder to process words in backward order. This approach is called bidirectional RNN. It allows the encoder to capture the information of a sentence from the front and back of a sentence. Besides that, additional RNN layers may also be piled on existing RNN layers on encoders and decoders. Second, an attention mechanism was introduced to allow for re-referencing to the input during decoding [22].

Additionally, more than one NMT models can be combined through ensemble modelling to produce a better translation. The main idea of this approach is that each NMT has different errors in translation modelling. By using several shared models, the translation error can be reduced thus further improving the quality of the translation produced.

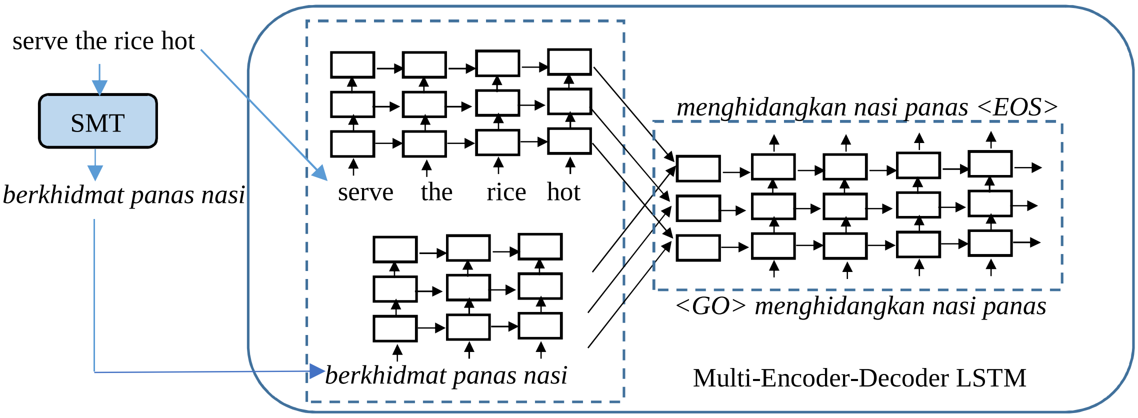
serve the rice hot
SMT
berkhidmat panas nasi
serve the rice hot
berkhidmat panas nasi
menghidangkan nasi panas <EOS>
<GO> menghidangkan nasi panas
Multi-Encoder-Decoder LSTM

Fig. 5 Multi-Source Encoder-Decoder LSTM [10]

C. Hybrid MT using multi-source encoder-decoder architecture

SMT and NMT have their strength in machine translation. In this paper, we combine the strength of an SMT into an NMT using a multi-encoder-decoder LSTM network for English-Malay translation. Fig. 5 shows the setup for English-Malay translation.

In general, we use a baseline SMT to translate a sentence in English to Malay. The idea is to use the translation produced by the SMT to assist the NMT in the translation. The translation produced by the SMT will be input into an encoder consist of LSTM network. At the same time, another encoder that also consists of LSTM network receives the input of the original sentence in English. Each encoder will embed the meaning of the sentence as a vector and send it to a decoder. The decoder then processes the vectors and emits the translation. For example, in Fig. 1, the SMT translate the English sentence "serve the rice hot" to "*berkhidmat panas nasi*." Both the English sentence and the translation produced by the SMT will be input to the multi-source NMT. The words are first converted to vectors through word embedding. The encoders can be imagined as a process that performs sentence embedding. The sentence vectors produced by the two encoders will be combined.

The approach works by concatenating the two hidden states from two source encoders. An LSTM variant combines the two hidden states and cells. The cell states from each encoder have their own forget gates. The final cell state and hidden state are calculated as in a normal LSTM as follow:

$$i = signmoid(W_1^i h_1 + W_2^i h_2) \qquad (6)$$

$$f = sigmoid\left(W_i^f\ h_i\right) \qquad (7)$$

$$o = signmoid\ (W_i^0 h_i + W_2^0 h_2) \qquad (8)$$

$$\mu = \tanh(W_i^u h_i + W_2^u h_2) \qquad (9)$$

$$c = i_f \odot u_f + f_1 \odot c_1 + f_2 \odot c_2 \qquad (10)$$

$$h = o_f \odot \tanh(c_f) \qquad (11)$$

Where i in equation 6 is an input gate of a typical LSTM cell. In equation 7, there are two forget gates indexed by the subscript i. In equation 8, o is the output gate of a normal LSTM, and ⊙ is element-wise multiplication.

During training, the actual reference/translation of a sentence is known. At the decoder, a special tag <GO> will be input, the first word of the translation, which is "*menghidangkan*" is expected. The weights of the networks will be adjusted using the backpropagation algorithm. The word will then be input to the decoder, and the second word, which is "*easy*" is expected. The process continues until a special tag <EOS> is produced.

D. The English-Malay parallel text corpus

In this section, we describe our work to collect our English-Malay parallel text corpus. This corpus consists of parallel text for training and testing. Parallel text for training was extracted from bilingual dictionaries, theses, and articles. The testing part was extracted from news articles and exam questions. A bilingual dictionary contains a lot of bilingual phrases and example sentences that can be a starting point for building a parallel text corpus. The other sources of English-Malay parallel sentences are theses and articles. In Malaysia, it is customary in many journals and theses to have the abstract to be written in Malay and English. Also, an abstract from an article also contains many recent terms from different domains that are useful in translation. Thus, the abstract of these documents provides another source for us to extract English-Malay parallel text.

The English-Malay bilingual dictionary contains many examples of translations for words, phrases and sentences that can be used as parallel text for the general domain. To extract the text, an OCR was first used to convert scanned document to text. The order and position of the source and target language sentences of interest were determined in the text manually, and regular expressions were used to extract these sentences. In some cases, language identification algorithm has to be used to separate the sentences. One way is by using n-gram language model. Spelling correction were also carried out. This were done using minimum edit distance and n-gram model.

The other source where English-Malay parallel text can be found is in the abstract of a document. The theses produced by Universiti Sains Malaysia were downloaded from an open access repository web site using web crawlers. A total of 2687 theses from 1981 to 2015 were downloaded. The same method was applied to articles from local journals. These theses are from various fields such as social science, humanities, business, computer science, engineering and so forth.

On the other hand, the journal is from the science and applied science domain. The title and abstract were identified using keywords. The text was segmented based on the sentence, and some pre-processing was performed such as separating the punctuations from words and converting uppercase letters to lowercase letters. The last step is to align the sentences in the English abstract file to the corresponding sentence in the Malay abstract file automatically using an alignment algorithm. This was done because the sentences in both languages are not necessarily in the same order. Also, not every sentence in the abstract was translated. The BleuAlign tool [23] was used for aligning the sentences. The BlueAlign approach uses a BLEU score for aligning sentences. The BLEU formula, in this case, is modified slightly to count until 2-grams only and not 4-grams. The BLEU score is a metric commonly used to evaluate the quality of a translation by comparing translation hypotheses/outputs with the reference translation. To align a source sentence to the right target sentence, a reference translation for the source sentence has to be created first. The reference translation was generated using an initial SMT. BleuAlign will then find the target language sentence that has the highest BLEU score for each reference translation.

For testing an MT, a different set of parallel text test was collected. The tests evaluate MTs in both the news domain and computer science domain. For news domain, the parallel text was extracted from Malaysiakini [24] news portal that produces news in English and Malay. Most of the news generated by this portal contains passages and sentences that are similar. For testing MT on a CS domain, we extracted English-Malay parallel sentences from the exam questions of the School of Computer Sciences, Universiti Sains Malaysia. These examination papers are a good source for building a parallel text corpus because the exam questions exist in bilingual since the year 2000.

III. Results and Discussion

Before the experiments carried out on our proposed hybrid MT is presented, we first discuss the performance of the baseline SMT and baseline NMT. The baseline English-Malay SMT was built using Moses toolkit [14]. We used the English-Malay parallel text corpus collected contains 478 thousand parallel sentences for training the MT models. For testing, two thousand English-Malay parallel sentences from computer science (CS) domain and news domain were used. GIZA++ was used to create the phrase translation model. A Malay text corpus [25] with about 870 MB was used to build 4-grams language models using SRILM [26].

On the other hand, our baseline English-Malay NTM used in the experiment was from a generic bidirectional LSTM encoder-decoder architecture with attention mechanism [27]. This NMT was based on Tensorflow 1.2, an open source software from Google. The encoder and decoder were configured as follow: one hidden layer, 512 states, no drop off and 60 thousand vocabularies. Additional out of vocabulary (OOV) words, words that do not occur in the training data, handling were added in the NMT. Two approaches were used. First, all numbers in digits and decimal numbers were normalized to <DIGIT> and <REAL> respectively using regular expression. Secondly, to overcome the problem of unknown English proper nouns that have the same surface form in Malay, we applied two approaches. The first approach simply replaced all <UNK> found in the hypothesis with OOV words extracted from the source language sentence in sequence.

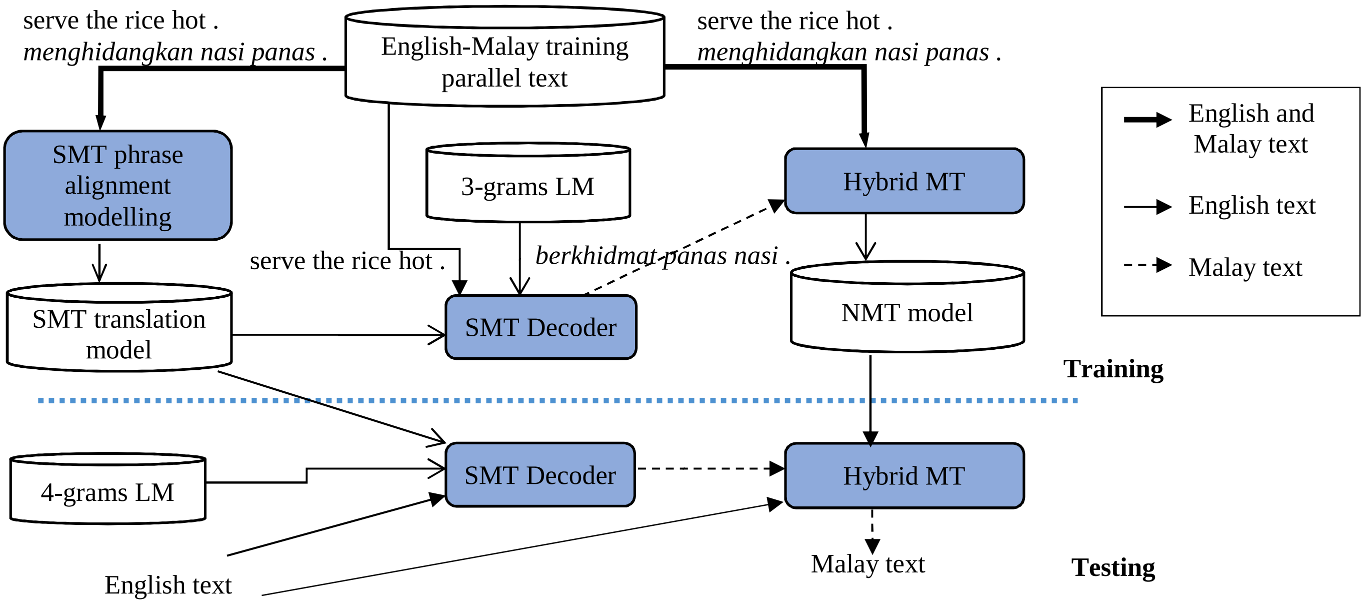

Fig. 6 Training and testing in hybrid MT

TABLE I
BASELINE ENGLISH-MALAY SMT AND BASELINE ENGLISH-MALAY NMT RESULTS

Approaches	**CS**	**News**
Baseline SMT	**21.13**	**48.35**
Baseline NMT	20.92	39.91
Baseline NMT + OOV	**21.21**	40.92

TABLE I shows the experimental results of baseline English-Malay SMT and baseline English-Malay NMT. The BLEU score for the baseline SMT was higher compared to the baseline NMT in the news domain. The baseline SMT obtained a BLEU score of 48.35, while the BLEU score for the baseline NMT is only 39.31. The usage of a language model in SMT was the reason why SMT obtained a very high BLEU score in the news domain. The 4-grams language model was trained using text extracted from online news articles. This result shows that SMT performs better in the in-domain test than NMT. In the CS domain, baseline SMT also produced a better result compared to baseline NMT. However, when additional OOV handling was carried out on NMT, the BLEU score improved to 40.92. The result on out-of-domain (CS) test showed that the performance of SMT and NMT are nearly the same. This result is different from the conclusion obtained in [3]. Probably the reason for the difference is because the data for the CS domain test was from exam questions, and most of the sentences still have a general structure even though there are CS OOV words in the sentences.

Next, we presented the setup of our proposed hybrid MT. Fig. 6 shows the steps to train the hybrid MT. The SMT used in the proposed hybrid MT was trained slightly differently. From Fig. 6, one will notice that the same parallel text was used for training the SMT translation model and also training the hybrid MT model. The English text of the parallel text was decoded by the SMT. The translation in Malay produced will then be input to the hybrid NMT (multi-source encoder-decoder LSTM). Since we used the same data in SMT for training the SMT translation model and also for decoding, we do not want the English-Malay SMT translation model to overfit, or else the hybrid MT will ignore the other input source in English. To do that, we set the maximum size of a phrase to 2 when building the SMT translation model, and during decoding, a 3-grams language model was used. Note that during testing, a 4-grams language model will be used instead. The hybrid MT consists of two bi-directional encoders and a decoder of LSTM networks. The other configuration was as follow: one hidden layer, 512 states, no drop off and 60 thousand vocabularies.

TABLE II shows the BLEU scores obtained using the hybrid MT. The results are very encouraging. In all the tests, the hybrid MT performs significantly better than both SMT and NMT. The best baseline NMT that we obtained for CS domain was 21.17 and 48.35 for the news domain. On the other hand, the hybrid approach produced a translation with the BLEU score of 34.83 in the CS domain and 60.05 in the news domain. When OOV handling was carried out, the BLEU score improved to 35.97 and 61.81 respectively. The results show that the hybrid NMT uses the suggested translation from the SMT and improve it further.

TABLE II
BLEU SCORE RESULT FOR HYBRID MT.

Approaches	**CS**	**News**
Hybrid MT	34.83	60.05
Hybrid MT + OOV	35.97	61.81

Finally, we also investigate the effect of sentence length on SMT, NMT and hybrid MT. See Table III. The study [3] show that NMT has a lower BLEU score on long sentences than SMT. We separate the test set based on sentence length to 5 groups: sentence with less than or equal to 5 words, sentence with 6 to 10 words, sentence with 11 to 20 words and sentence with 20 to 30 words, and sentence with more than 30 words. We will compare the results for SMT and NMT on the CS domain since the test result on the domain is nearly the same (nevertheless, we also provide the results for news domain in the Table III below). On SMT, NMT and hybrid MT, all have better BLEU score when the sentence length increase. The reason longer sentence has a higher BLEU score is because the MT is able to use more contextual information to predict the translation. The results show that NMT performs worse than SMT on sentences that are short (6 to 10 words) and very long (more than 30 words). NMT performs the best on sentences that have an average length (20-30 words). Hybrid MT gives the higher quality translation compared to SMT and NMT in all cases (different sentence length).

TABLE III
DIFFERENT SENTENCE LENGTH AND BLEU SCORE RESULT FOR SMT, NMT AND HYBRID MT.

Num. of words	**BLEU Score**					
	SMT		NMT		Hybrid	
	CS	News	CS	News	CS	News
Ave.	**21.13**	**49.59**	**21.21**	**40.92**	**35.97**	**61.81**
<= 5	15.85	-	16.56	-	25.04	-
6-10	19.68	38.01	17.95	31.31	29.95	46.68
11- 20	19.87	**53.46**	21.00	43.01	33.76	61.92
20-30	22.14	51.58	**23.17**	**44.20**	**35.71**	**64.03**
>30	**23.88**	48.52	23.12	39.23	35.40	59.36

IV. CONCLUSIONS

In this research paper, we proved that the hybrid MT improves the quality of the translation of SMT and NMT. The hybrid MT uses the suggestion translation from the SMT and improves the translation that it produces. The BLEU score increases from 21.21, 48.35 to 35.97, and 61.81 for CS domain and news domain respectively. The results show that for in-domain (news) translation, SMT produces better result due to the language model. In term of sentence length and BLEU score, NMT perform worse than SMT when the sentence to translate is short and very long. Our experiments show that the hybrid MT using multi-source encoder-decoder long-short-term memory produce the translation with higher BLEU score compared to SMT and NMT in all cases.

ACKNOWLEDGMENT

Fundamental Research Grant (FRGS) 203.PKOMP.6711536 from Ministry of Higher Education Malaysia supports this work.

REFERENCES

[1] D. Bahdanau, K. Cho and Y. Bengio, “Neural machine translation by jointly learning to align and translate”. *CoRR abs*/1409.0473, 2014.

[2] T. Luong, H. Pham and D. C. Manning, “Effective approaches to attention-based neural machine translation,” in *Proc. Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal, pp. 1412–1421, Sept. 2015.

[3] P. Koehn and R. Knowles, “Six challenges for neural machine translation,” in *Proc. Workshop on Neural Machine Translation*, pp. 28–39, 2017.

[4] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Arxiv preprint arXiv:1406.1078,* 2014.

[5] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. International Conference on Neural Information Processing Systems*, pp. 3104-3112, Dec. 2014.

[6] L. Dahlmann, E. Matusov, P. Petrushkov and S. Khadivi, “Neural machine translation leveraging phrase-based models in a hybrid search,” in *Proc. Conference on Empirical Methods in Natural Language Processing*, Sept 2017.

[7] F. Stahlberg, A. de Gispert, E. Hasler and B. Byrne, “Neural machine translation by minimising the Bayes-risk with respect to syntactic translation lattices,” in *Proc. Conference of the European Chapter of the Association for Computational Linguistics*, vol 2, pp. 362–368, April 2017.

[8] X. Wang, Z. Lu, Z. Tu, H. Li, D. Xiong, and M. Zhang, “Neural machine translation advised by statistical machine translation,” in *Proc. AAAI Conference on Artificial Intelligence*, 2017.

[9] J. Du and A. Way, “Neural pre-translation for hybrid machine translation,” in *Proc. MT Summit XVI*, vol.1, pp. 27–40, Sept. 2017.

[10] R. Dabre, F. Cromieres and S. Kurohashi, “Enabling multi-source neural machine translation by concatenating source sentences in multiple languages”. arXiv preprint arXiv:1702.06135. 2017.

[11] B. Zoph and K. Knight, “Multi-source neural translation,” in *Proc. NAACL-HLT*, pp. 30-34, June 2016.

[12] J. Zhang, Q. Liu and J. Zhou, “ME-MD: An effective framework for neural machine translation with multiple encoders and decoders,” in *Proc. IJCAI*, pp. 3392-3398, Aug. 2017.

[13] P. Koehn, “Pharaoh: a beam search decoder for phrase-based statistical machine translation models,” in *Proc. AMTA*, pp. 115-124, Sept. 2004.

[14] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin and E. Herbst, “Moses: Open Source Toolkit for Statistical Machine Translation,” in *Proc. ACL 2007,* June 2007.

[15] M. Olteanu, C. Davis, I. Volosen and D. Moldovan, “Phramer – an open source statistical phrase-based translator,” in *Proc. Workshop on Statistical Machine Translation*, pp. 146-149, June. 2006.

[16] F. J. Och and H. Ney, “A systematic comparison of various statistical alignment models,” in *Proc. Computational Linguistics*, vol. 29, no.1, pp. 19–51, 2003.

[17] Y. Deng and W. Byrne, “MKKT: An alignment toolkit for statistical machine translation,” in *Proc. Human Language Technology Conference of the NAACL*, pp. 265-268, 2006.

[18] I. Mohd Yassin, R. Jailani, M. S. A. Megat Ali, R. Baharom, A. H. Abu Hassan and Z. I. Rizman, “Comparison between Cascade Forward and Multi-Layer Perceptron Neural Networks for NARX Functional Electrical Stimulation (FES)-Based Muscle Model,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 1, pp. 215-221, 2017.

[19] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[20] A. A. Amri, A. R. Ismail and A. Ahmad Zarir, “Convolutional neural networks and deep belief networks for analysing imbalanced class issue in handwritten dataset,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 7, no. 6, pp. 2302-2307, 2017.

[21] I. Sutskever, O. Vinyals and Q. V. Le, “Sequence to Sequence Learning with Neural Networks”. *Advances in Neural Information Processing Systems*, pp. 3104-3112, 2014.

[22] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. California, US: O'Reilly Media. 2017.

[23] R. Sennrich, M. Volk, “MT-based Sentence Alignment for OCR-generated Parallel Texts,” in *Proc. AMTA 2010*, 2010.

[24] (2018) The MalaysiaKini website. [Online]. Available: https://www.malaysiakini.com/

[25] T.-P. Tan, H. Li, E. K. Tang, X. Xiao and E. S. Chng, “MASS: A Malay Language LVCSR Corpus Resource,” in *Proc. Oriental Cocosda*, pp. 25-30, Aug. 2009.

[26] A. Stolcke, “SRILM – an extensible language modeling toolkit,” in *Proc. International Conference on Spoken Language Processing*, pp. 901–904, 2002.

[27] A. Bérard, O. Pietquin, L. Besacier and C. Servan, “Listen and translate: A Proof of Concept for End-to-End Speech-to-Text Translation,” in *Proc. NIPS*, pp. 1–5, 2016.