A SYSTEMATIC READING IN STATISTICAL TRANSLATION: FROM THE STATISTICAL MACHINE TRANSLATION TO THE NEURAL TRANSLATION MODELS.

Zakaria El Maazouzi, Badr Eddine El Mohajir & Mohammed Al Achhab
N2T Laboratory, National School of Applied Sciences
University Abdelmalek Essaadi, Morocco

z.elmaazouzi.ma@ieee.org; b.elmohajir@ieee.ma; alachhab@ieee.ma

ABSTRACT

Achieving high accuracy in automatic translation tasks has been one of the challenging goals for researchers in the area of machine translation since decades. Thus, the eagerness of exploring new possible ways to improve machine translation was always the matter for researchers in the field. Automatic translation as a key application in the natural language processing domain has developed many approaches, namely statistical machine translation and recently neural machine translation that improved largely the translation quality especially for Latin languages. They have even made it possible for the translation of some language pairs to approach human translation quality. In this paper, we present a survey of the state of the art of statistical translation, where we describe the different existing methodologies, and we overview the recent research studies while pointing out the main strengths and limitations of the different approaches.

Keywords: Neural networks, recurrent neural networks, natural language processing, neural language model.

INTRODUCTION

Automatic translation of natural languages or machine translation framework refers to making use of computing power to build sophisticated data models

Received: 3 July 2017 **Accepted:** 28 August 2017

to translate one source language into another. Since the beginning through the late 1960s, many researchers have focused on presenting new techniques and improving the performance of automatic translation systems. Through all the past years, the machine translation research community has proposed different approaches, namely the phrase-based statistical machine translation, the most widely used. The core of the SMT approach refers to using count-based probability models which estimates the probabilities of units based on the analysis of the monolingual and bilingual corpora. Statistical machine translation has exponentially improved since IBM pioneered its word-based model in the late 1980s and early 1990s (Brown et al., 1990, Brown et al., 1993, Berger et al., 1994). Also with the introduction of phrase-based translation (Och et al. 1999, Marcu and Wong 2002, Kohen et al., 2003, Ochand Ney, 2004).

Recently researchers have proposed a novel approach for automatic translation completely based on artificial neural networks known as neural machine translation or encoder-decoder models. Intuitively, neural machine translation conducts an end-to-end translation, using a source language encoder and a target language decoder. This technique showed interesting results and improvement in the state of the art results in the field of statistical translation (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015; Gulcehre et al., 2015).

To the best of our knowledge, no previous work presenting a survey of the novel methodologies of neural machine translation have been published lately. Moreover, surveys studies used to be a solid source of ideas and information for interested individuals with no prior knowledge in a given field. For all these purposes, in this endeavor, we capitalize on the findings of the relevant published studies addressing the neural machine translation to present a document that summarizes the different aspects and advances in this topic. To ensure a good understanding of the discussed concepts, we will go through a detailed reading in machine translation, starting with an extended introduction to statistical and neural machines translation. This is followed by a survey of a selection of empirical studies recently published and concerning particularly with neural machine translation. A discussion section to analyze the advances and point towards the open lines of research that still requires more work in the area of neural machine translation.

GENERAL FRAMEWORK OF STATISTICAL TRANSLATION

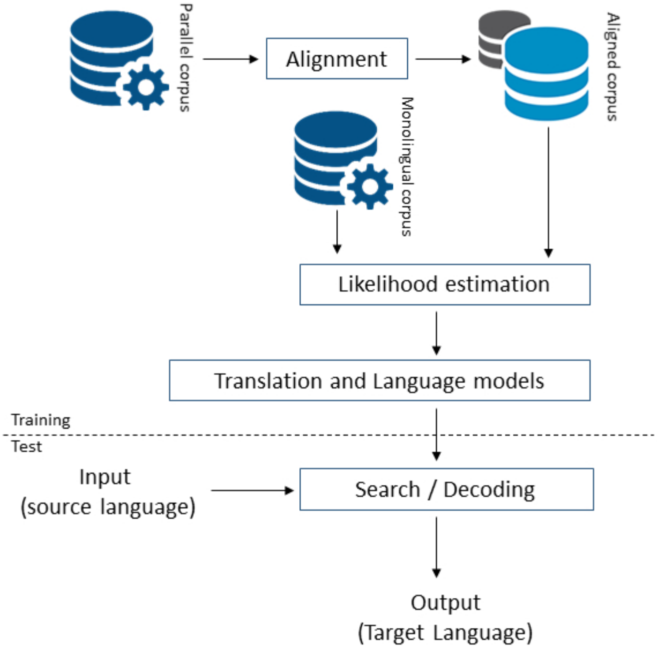


Figure 1. Simplified machine translation architecture.

The statistical machine translation in Figure 1 involves two distinct processes, generally following the machine learning workflow: training and decoding (or test), in order to train probabilistic models to search for the best translation. Training being a core part of the learning process, it refers to using count-based probability calculations on the dataset to train two statistical models; the first known as a translation model trained on a parallel corpus of the source and target datasets, and a language model extracted from training on a monolingual corpus. In statistical machine translation, a single source sentence could be mapped to many possible translations, i.e. the translation function is a conditional probability distribution $P(T \mid S)$ over a target sentence T, given the source sentence S. Thus the statistical machine translation core idea is to search for the translation that maximizes the former probability, i.e. the most appropriate for a given input sentence. Here the search problem might be addressed via two models used to score the candidate's translation, the noisy channel and the log-linear models. In what follows, we will give a definition of these two models and other important concepts of the statistical machine translation.

Noisy channel model. The noisy channel is a generative statistical model traditionally used in literature in statistical translation. It involves the product (3) of two features referring to the translation and language models $P(S|T)$ and P(T). The former, presents the likelihood that the source sentence S and the candidate translation T are translationally equivalent, i.e. a stochastic process that corrupts the target language to produce source language sentences while the second is a stochastic model trained on a monolingual corpus of the target language to generate target language sentences and asses the validity and the fluency of the candidate translation T in the target context. The main intuition behind the noisy channel model is to predict the target sentence equivalent for a given source sentence, via maximizing the probability $P(T|S)$:

$$E = \mathrm{argmax}_E\, P(T|S) \;\; (1) \qquad (1)$$

Then by applying the Bayes rule, we get:

$$E = \underset{E}{\mathrm{argmax}} \frac{P(T).P(S|T)}{P(S)}$$

As for a different E, P(S) is always constant we get:

$$E = \underset{E}{\mathrm{argmax}}\, P(T).P(S|T) \qquad (2)$$

Log-linear model. The log-linear model refers to the use of a discriminative model as a generalization of the noisy channel model, which brings more context into the modeling process. Log-linear is based on the maximization of the sum of several weighted features h_m, where the weights or model scaling factors λ_m are used to determine the contribution of a single feature to the overall model. Here as the model's name indicates, log probabilities are used to express the translational, language models, distortion and other features.

$$T = \underset{T}{\mathrm{argmax}} \sum_{m=1}^{M} \lambda_m \,.\, h_m(T,S) \qquad (3)$$

Translational model. The main purpose of the translation equivalence model is to estimate the lexical correspondence between the source and the target sentences. This model is trained to learn to model the translation of the input, the number of words in the output, the order of the translation within the output sentence, and the number of words to be generated. Features which are statistically referred to are :

- Lexical equivalence probability that relates to the likelihood that the source word *S* is translated into the target word T.
- Fertility models the probability that the word *S* generates a number of new words.
- Distortion $d(j \mid i, m, n)$ relates to the probability that the word in the *j* position generates a word in the *i* position given the length of the source and the target sentences *m* and *n* respectively.
- The probability that a target word is generated from ε.

Language model. It is an assessment model that estimates how probable a sentence is correctly translated into the target language. A naïve estimation on a monolingual corpus with N sentences is extracted from the frequency calculation of a unit s (4):

$$P(S) = \frac{Ns}{Nsentences} \tag{4}$$

This formula is obviously underperforming when dealing with long sentences or unseen units in a corpus, as it is difficult to get a reliable estimation for sentences which are not occurring in the dataset, i.e. the use of naïve estimation in those cases would result in a zero probability. From this perspective, researchers have investigated the use of the n-gram approach to extract the language model estimation. So what is n-gram?

N-gram. The n-gram approach as derived from the Markov assumption. It emphasizes that only the previous $n - 1$ words matter to predict a word at a position *i*. Basically, n-gram refers to breaking a given sentence into smaller sequences to simplify the process. This technique helps in calculating the probability distribution of the full configuration as the product of the probability of each word w_i conditioned on the n previous words referring to a predefined window size e.g. unigram, bigram, trigram… (5).

$$P(w1, \dots, wn) = \prod_{i=1}^{n} P\left(w_i \middle| w_{i-(n-1)}, \dots., w_{i-1}\right) \tag{5}$$

To not discard infrequent units, a linear interpolation technique is implemented to smooth the conditional probabilities and prevent resulting in zero probability, e.g. for an input sentence ***"je voyagerai demain,"*** if we use a bigram model we get:

$$P(S) = P(je|\emptyset).P(voyagerai|je).P(demain|voyagerai)$$

Where for each factor:

$$P(voyagerai|je) = \frac{N_{(je\ voyagerai)}}{N_{(je)}}$$

By implementing the linear interpolation we get:

$$P(voyagerai|je) = \lambda_2 \frac{N_{(je,voyagerai)}}{N_{(je)}} + \lambda_1 \frac{N_{(je)}}{N_{(words)}} + \lambda_0$$

$$h_t = F_\theta(x_t, h_{t-1})(6)$$

Alignment in statistical machine translation. Alignment in statistical machine translation is proportional to the statistical translation model used. Here we point to two main types of models in statistical machine translation, which are the word and the phrase- based models.

Word- based IBM model. Word-based or IBM model (Brown et al., 1993), is the starting model of statistical machine translation, and it claims that the words in a sentence are multiplied, translated separately and then scrambled around to form the target sentence. The main idea behind the word-based model is the use of single words as translation units. Thus each word is translated to its equivalent in the target sentence. Then using alignment and reordering models, the words are rearranged to form a valid composition in the target context.

Phrase-based model. In some cases where a sequence of words is naturally translated as a single unit, a word-based model translation does not offer a reliable translation. The study of Och and Ney (2004) showed that using phrases as translational units can achieve a more reliable translation compared to the word-based. The reason is that this approach allows the use of more local and short- range local contexts to translate a group of words as a single unit to counter to the word-based. Note that a phrase is a sequence of words consistent with word alignment (Figure. 2), that has nothing to do with the linguistic element. $P(S|T)$ probability can always be obtained via the same process as for word- based with some modifications:

- Suggested source sentence into phrases.
- Translate each phrase into the target language.
- Reorder and align the output.

In a phrase-based model the calculated phrase's probabilities should be placed together with their phrase pairs in a so- called translation table. The latter is then used as a reference when searching for the best translation that maximizes the translation scores and for the rest of calculations.

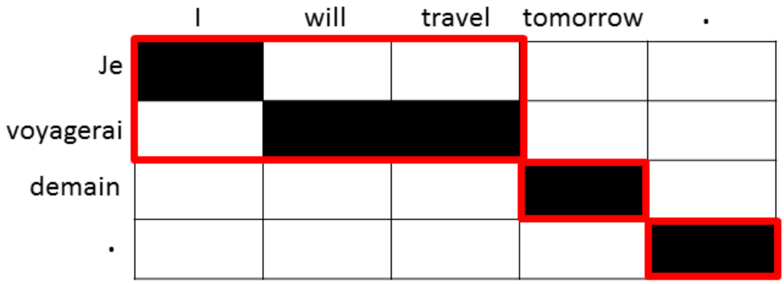


Figure 2. Example of phrase consistent with word alignment extraction.

Alignment. As corpora are generally not aligned at word level, the latter becomes tricky. Hence, the Expectation-Maximization algorithm (Figure. 3) is introduced to learn the alignment at word level. IBM models define alignment as asymmetric since each target word corresponds to only one source word but with the introduction of the fertility, the same cannot be said.

Search as a main problem of translation. The decoding phase (Figure. 4) is referred to as search algorithm. In statistical machine translation, the intuition behind using a search algorithm is to look for possible translations and score them: then maximize the obtained scores to get the most probable one given both the language and the translation models.

In theory, decoding involves generating the full set of possible translations for a given input string. The translation process is simply about matching phrases/ words from the input sentence against the translation model and, where available, retrieving their translations, ordering the generated substrings and concatenating them to produce a full translation hypothesis. Those hypotheses are used by the decoding algorithm to select the best one which would have the lower cost, i.e. maximizing the product of the language and translation models.

It is not mandatory to proceed decoding on a strictly left-to-right basis. As for some language pairs, e.g. Arabic/French, the left to right decoding would not be the logically appropriate method with decoder being based on probability calculations (1), selection of the best translation would always be made irrespectively of the nature of the source and target language.

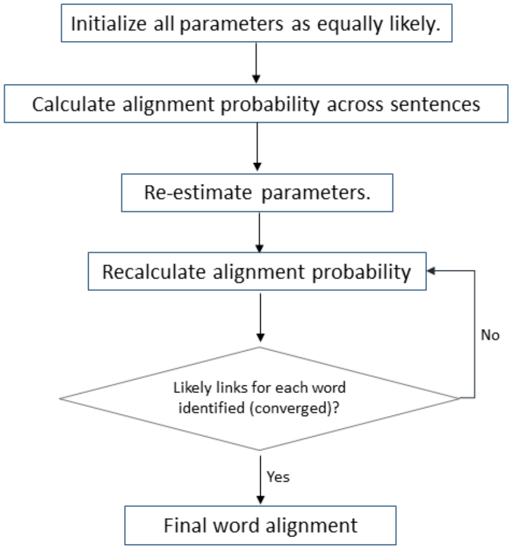

Initialize all parameters as equally likely.
Calculate alignment probability across sentences
Re-estimate parameters.
Recalculate alignment probability
No
Likely links for each word identified (converged)?
Yes
Final word alignment

Figure 3. Expectation maximization algorithm.

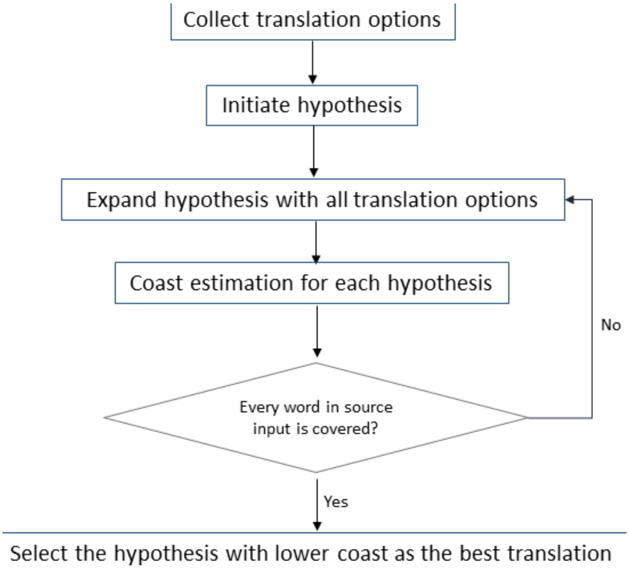
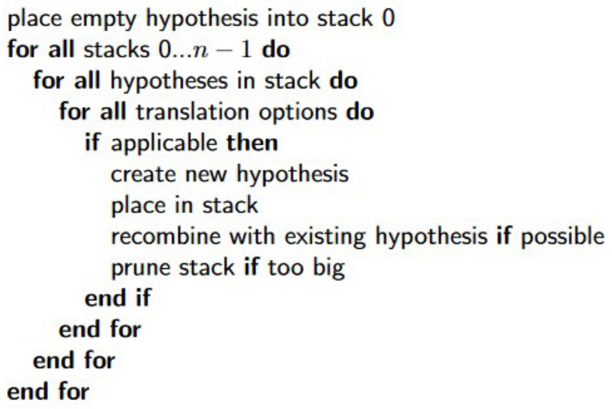

Collect translation options
Initiate hypothesis
Expand hypothesis with all translation options
Coast estimation for each hypothesis
No
Every word in source input is covered?
Yes
Select the hypothesis with lower coast as the best translation

Figure 4. Decoding core algorithm with exhaustive search.

Since the best translation is equivalent to the lowest cost hypothesis, the selection process should base its estimation on the cost of all the hypothesis to correctly select the lowest one. However, since the number of the hypothesis is exponential with the number of source words, decoding also becomes an optimization problem. In this context, several studies have proposed methods for optimal solutions. Berger et al. (1996) and Och et al. (2001) proposed such depth-first search methods as stack decoders. Also, Wang and Waibel (1997) and Tillmann and Ney (2003) developed breadth-first search methods, i.e. beam search. On the other hand, Germann (2001), and Watanabe and Sumita (2003) proposed greedy- type decoding methods.

The state of the art implementation of decoding for statistical machine translation is a beam search decoder (Kohen et al., 2007). A beam search is a comparison of the hypothesis that covers the same number of foreign words and a selection of the inferior hypotheses via the cost of each to prune them out. The pruning measure is not only about the cost, but also an estimation of the future cost. Beam size can be defined beforehand, so with a coast and future coast we can prune hypotheses that fall outside the beam; see stack decoding algorithm in Figure. 5.



```
place empty hypothesis into stack 0
for all stacks 0...$n - 1$ do
  for all hypotheses in stack do
    for all translation options do
      if applicable then
        create new hypothesis
        place in stack
        recombine with existing hypothesis if possible
        prune stack if too big
      end if
    end for
  end for
end for
```

Figure 5. Hypothesis expansion via stack decoding (Koehn 2007)

NEURAL MACHINE TRANSLATION

Recently, artificial neural networks have been successfully applied to many research endeavors that require a learning process, for instance, image processing and bioinformatics. Particularly, the marriage between natural language processing and neural networks has shown remarkable progress in the results of many study cases; one could be the neural machine translation. The latter is a newly proposed framework leading the way for machine translation research based purely on neural networks. Fresh results outlined by Hang Luong, Kyunghyun Cho, and Christopher Manning (2016), Kalchbrenner and Blunsom, (2013), Sutskever et al. (2014), and Kyunghyun Cho et al., (2014), have demonstrated a significant improvement in the translation of many language pairs. Compared to the state of the art results achieved by the classical translation models, mainly the phrase-based and the syntax-based translation systems, the neural MT outperforms those existing thechniques.

The core idea of neural machine translation involves an end-to-end trained model in both the learning and the decoding processes. In other words, via a single artificial network, a neural translation model designs a fully trainable model where every component is tuned based on training bilingual corpora to maximize its translation performance, i.e. it combines two neural networks (an encoder trained to compute a representation of each word in an input sentence, better known as word embedding, and a decoder which is a neural language model trained to generate one target word at a time t based on the source context and target history). Both jointly trained together to maximize the conditional probability $P(X|Y)$ of translating a source sentence, $x_1,...x_n$ to a target sentence, $y_1,...y_n$.

Artificial neural networks. They are limited imitations of how our brains work. The first created computational model for neural networks based on mathematics and algorithms was called threshold logic (Warren McCulloch and Walter Pitts, 1943). However, artificial neural nets have only had a big resurgence lately because of the advances in computer hardware, in particular after the introduction of the graphics processing units (GPUs), and with the appearance of deep learning. A neural network is an interconnected web of nodes and edges designed to perform complex tasks such as classification, regression and prediction. Neural nets are also highly structured networks and have three kinds of layers: input, output and so-called hidden layers, which refer to any layer between the input and the output layers and nodes which are intended to calculate different types of activation functions.

Moreover, neural networks count a large range of types for instance and non-exhaustively:

- Feed forward neural nets, the simplest and basic version. Conventional neural nets, widely applied in image processing domain.
- Recurrent neural nets, the one commonly used in NLP.

- Complex-valued neural nets based on complex numbers as entries, also successfully applied to embedding watermarks (Olanweraju et al, 2010).
- Multilevel self-organizing MAP, systematically evaluated for clustering applications by Shamsuddin et al., 2008.

As stated, the recurrent architecture is the one widely used in machine translation, due to its ability to handle and process sequential data, e.g. unfixed-length sentences. The intuition behind RNNs (Figure. 7) is their ability to maintain the internal state at a time *t* to be used in calculations at time *t* + 1. Each recurrent unit is classically composed from the input/output parts and a hidden unit connected via weighted connections, responsible for the calculation of an activation/transition function modeled mathematically as follows:

$$h_t = F_\theta(x_t, h_{t-1}) \tag{6}$$

where F_θ is a function parametrized by θ referring to a vector of weights and bias, and which takes as input the new element *xt* and the history h_{t-1} up to the $t-1$ th input element. There's a wide range of activation function types used to calculate the activation value at each node of the network given the weights, bias and input, namely sigmoid and hyperbolic tangent (Tanh) which are the ones usually used, especially in the case of a non-linear distribution. The Softmax function (13) is also used to estimate the conditional probabilities by modeling them as a multi-classification problem $p(y = c|x)$. RNNs are trained using the back propagation through time algorithm, which involves the calculation of loss function (LeCun, Chopra et al., 2006; Bengio et al., 2015) at the output level to perform updates on network parameters after each learned example using the stochastic gradient descent algorithm (Bottou, 2012). Despite that some studies have proposed better training technics such as Genetic Algorithm and Swarm Intelligence i.e. Artificial Fish Swarm optimization method (Shafaatunnur Hasan et al,. 2012) instead of backpropagation that tends to slow the convergence rate, the latter still the most common algorithm in the applications of neural networks in NLP.

Neural networks applied to language modeling. In NLP, a statistical language model is a conditional distribution on the identity of the ith word in a sequence, given the identities of all previous words. Language models also tend to estimate the distribution of natural language as accurately as possible.

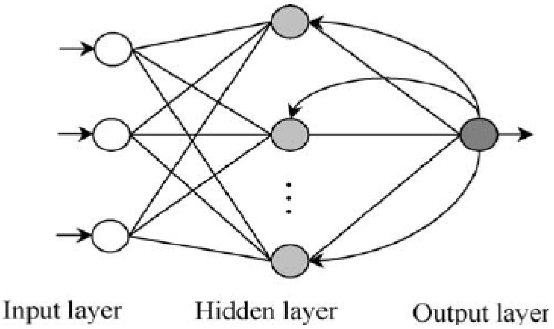


Figure 6. The architecture of a recurrent neural network.

Recently neural networks have been successfully applied to language modeling, via the so called neural language models that approach language modeling as a single neural network trained to maximize a probability distribution over a given unit of a sentence. In the case of neural machine translation, the decoder plays the role of a neural language model, as it learns to predict a word in a configuration at time *t* given a context that refers to the previously predicted word, history at time t-1 and a vector representation of the source sentence generated by the encoder. In addition the use of recurrent neural models in the translation task, forces the probability distribution of a sentence *p*(*X*) to be written as a recurrence, i.e. first it is rewritten as a probability of all the sequences composing it, $p(X) = p(x_1,..., x_p)$. Then by applying the definition of conditional probability $p(X|Y) = \frac{P(X,Y)}{P(Y)}$ on the integrity of the words we get: $p(X) = p(x_1)p(x_2| x_1,) p (x_3| x_1, x_2)...p (x_n| x_1,..., x_{n-1})(7)$. Consequently, a recursive formula could be formulated as with:

- s_t: A sequence at time *t*
- $s>_t$: Refers to all the sequences before time *t*.

By introducing an RNN language model $p(s_t| s>_t)$ at each time *t* we get:

$$\begin{cases} p(s_t|s_{<t}) = g_\theta(h_{t-1}) \ (9) \\ h_{t-1} = \emptyset_\theta(x_{t-1}, h_{t-2}) \ (10) \end{cases}$$

Given the input sequences and the previous history, the RNN predicts the next symbol via the calculation of the probability distribution conditioned on the whole history up to the $(t-1)^{\text{th}}$ symbol.

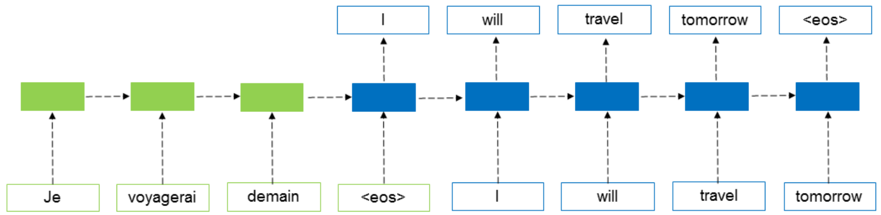


Figure 7. The architecture of the neural machine translation system.

The encoder-decoder translation system. Neural machine translation as described in the previous sections is based on an end- to- end encoder-decoder approach that translates a given source sentence into its equivalent target. The encoder and decoder being two recurrent neural networks jointly trained together to find the best translation, each of them has clear-cut missions. The encoder has been identified as a smart compressor of the input sentence into a fixed-length vector numerical representation, also known as a context vector or word embedding vector that captures every single detail of the source sentence in some way that the distance between two configurations (Figure 8) could be significant (Mikolov et al., 2013).

Once encoded, the resulting context vector is then conveyed to the decoder. As a recurrent neural language model, the decoder tends to generate a single target word at each sequence of time, given the representation of the full source context and the previously generated word and history as the target context. It should be noted that the input configuration is injected to the encoder-decoder system as one-hot aka 1-of-k vectors. This encoding technique is used to convert words of a sentence into a numerical format, which is more suitable as data for neural networks. The one- hot encoding is the simplest technique for encoding words in the vocabulary. In a nutshell, the one- hot vector of a given word is a vector filled with zeros except for the one at the position referring to its ID in the vocabulary.

In the rest of this section, we will outline both the encoder and decoder and the mathematic intuition behind them separately.

The encoder. In Figure 7 on the left side the encoding process is a straightforward application of the recurrent neural network. It takes the one-hot vector of each word in a source sentence as an input. Then it projects a word vector at time *t* to a matrix that will contain the whole phrase representation with predefined row size and the number of columns depending on the number of words in the source vocabulary. The resulting vector from the previous step is subjected to some calculations in the hidden state h_t (11) at each time t until getting the final internal state of the actual network. Note that during the training phase each element of the continuous vector is updated shortly and jointly with the rest of the parameters of the network to maximize the translation performance. $h_{i\epsilon[0,t]} = F_\theta (h_{i-1}, s_i)$ (11) with s_i the continuous vector representation at an iteration i.

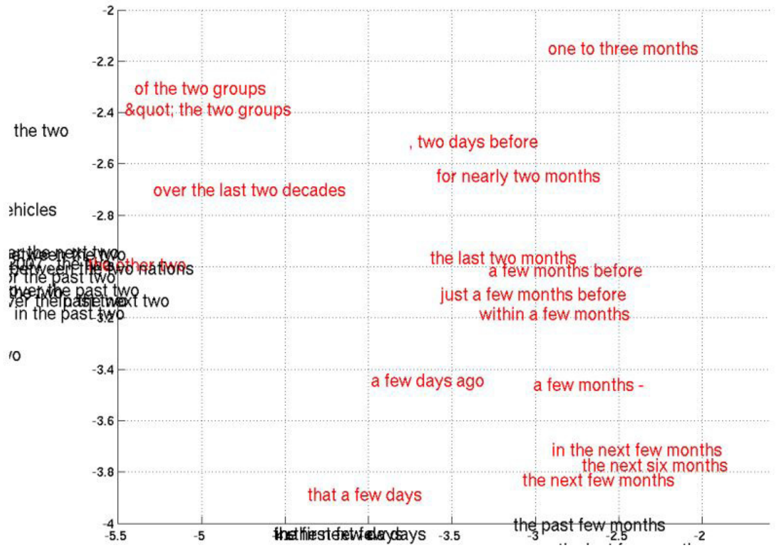


*Figure 8. T*he zoomed-in view of specific region (color–coded) 2–D embedding of the learned phrase representation (Kyunghyun Cho and al. 2014)

The decoder. In Figure 7 on the right side tends to generate a target word at each step, by involving a weighted distribution over the encoded source sentence vectors. This process refers to calculating the network's internal state z_i by using the context vector h_t obtained in the encoding process, the previous generated word and the previous internal state z_{i-1}. Then, based on the internal state z_i, the model scores (12) of each candidate target word is based on how likely it is to follow all the preceding translated words referring to the source sentence and by assigning a probability p_i to it.

$$e(k) = w_k^T z_i + b_k \qquad (12)$$

with w_k as the target word, z_i the RNN's calculated internal state and b_k the bias vector.

Once the words are scored, they are subjected to a softmax normalization function (13), in furtherance of squashing the scores' values into a [0, 1] interval, i.e. generating a probability distribution over the calculated scores. The latter is involved in selecting the best translation by using a simplification algorithm. We should mention that the word and the decoder's internal state are proportional, as the more correctly they align, higher the score gets, i.e. the product $w_k^T z_i$ gets large.

$$p\left(y_t = k \middle| y_1, y_2, \dots, y_{t-1}, c\right) = \frac{\exp(e(k))}{\sum_j \exp(e(j))} \qquad (13)$$

EMPIRICAL STUDIES IN NEURAL MACHINE TRANSLATION

In this section, we will survey the recent work relevant to the topic of neural machine translation, and conclude about the findings of those studies.

In the cutting-edge paper "Sequence to Sequence Learning with Neural Networks" by Sutskever, Ilya, Oriol and Quoc (2014), presented the results of their experiments conducted on neural networks for translation task. First, they proposed a general end-to-end approach to sequence learning based on a multilayered Long Short-Term Memory (LSTM) to map the input sequence to a vector of a fixed dimensionality, and another deep LSTM to decode the target sequence from the vector (Sutskever et.al 2014). In an other experiment, they used the LSTM to re-rank the 1000 hypotheses produced by a phrase-based SMT system which is also used for comparison with the proposed end-to-end approach. Finally, they also studied the effect of reversing the order of words in source sentences on the performance of LSTMs.

As a result, Sutskev's approach has achieved by far the best result of direct translation with large neural networks. According to the BLEU metric, the end-to-end translation experiment obtained 34.81 BLEU score on the WMT'14 English to French translation task, by directly extracting translations from an ensemble of 5 deep LSTMs (with 384M parameters and 8,000-dimensional state each) using a simple left-to-right beam search decoder. Compared to the SMT baseline used in the study, and which has trained on the same dataset, Sutskever's method showed an advancement of +1.51 BLEU score. Moreover, taking a look the results of rescored the publicly available 1000 best lists of the SMT baseline on the same task. The approach presented in this paper improved the baseline by +3.2 BLEU points. One last finding of this study was the fact that the LSTM did not suffer on very long sentences, which is a common issue of this kind of task. The authors gave credits of these results to the key contribution of reversing the order of words in the source sentence but not the target sentences in the training and test set.

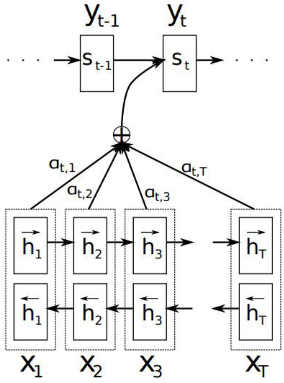


Figure 9. The graphical illustration of the proposed model trying to generate the tth target word y_t given a source sentence (x_1,…,x_T) Bahdanau et al. 2015

Bahdanau et al., 2014**.** One major drawback of the conventional neural machine translation is its underperformance when the length of an input sentence increases (Cho et al., 2014b). The reason is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This might cause some difficulties for the neural network to cope with long sentences, especially those that are longer than the samples in the training corpus. To resolve the problem, this article proposed an intention-based encoder-decoder system that learns jointly to align and translate. First, a source sentence is encoded into a sequence vector instead of a fixed length vector, using what is called bidirectional recurrent neural networks. The latter annotates each word by summarizing the preceding and following words, by reading the input sentence in order and calculates the forward hidden states. Then applies the same process in the backward order respectively. Finally, it concatenates the forward and backward hidden states to form the full representation of a given word $h_j = \left[\vec{h}_j^T, \overleftarrow{h_J^T}\right]$.

The approach of Bahdanu. for decoding a translation is a slightly different process compared to the classical one. They introduced a distinct context vector c_i depending on the source annotation and the attention score referring to a measurement of importance of a source word in respect to the previous target history to generate a target word at a given step. Mathematically, the proposal of Bahdanau conditioned the probability on this distinct context vector for each target word y_i as follows:

$$p(y_i|y_1, \dots, y_{i-1}, X) = g(y_{i-1}, s_i, c_i) \tag{14}$$

The context vector c_i is, then, computed as a weighted sum of the annotations h_j:

$$c_i = \sum_{j=1}^{Tx} \propto_{ij} h_j \tag{15}$$

$$\text{given } \propto_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{Tx} \exp(e_{ik})} \tag{16}$$

$$\text{and } e_{ij} = a(s_{i-1}, h_j) \tag{17}$$

Note that the alignment model directly computes a soft alignment e_{ij}, via a feedforward neural network which is jointly trained with all the other components of the proposed system.

The experiments conducted on the bilingual, parallel corpora provided by ACL WMT '14 were compared to an RNN Encoder-Decoder suggested by (Cho et al., 2014a). The proposed study demonstrated to be more robust to the length of the sentences as it performed well even with over 50 sentences in length. Also it outperformed the reference baseline used for comparison purpose according to the BLEU metric.

Effective Approaches to Attention-based Neural Machine Translation (Luong et al., 2015)**.** In this work, the authors extended the idea of attention-based neural machine translation from Bahdanau et al. (2014). They designed two novel types of attentional-based models, a global approach and a local one, each of which addressed the alignment process of the translation task differently. Similar to the existing attentional model, a simple concatenation layer was employed to combine information from the target hidden state and the source context vector into an attentional hidden state vector, which was then fed through the softmax layer to produce the predictive distribution. However, it differed in the global and local classes, i.e. the attention was placed on all source positions or only a few source positions.

Global attention: In this case, the context vector was computed as the weighted average over all the source hidden states. Instead of the concatenation of the forward and backward source hidden states in the bidirectional encoder as in Bahdanau's model. The hidden states at the top LSTM layers in both the encoder and the decoder were simply used for context computation.

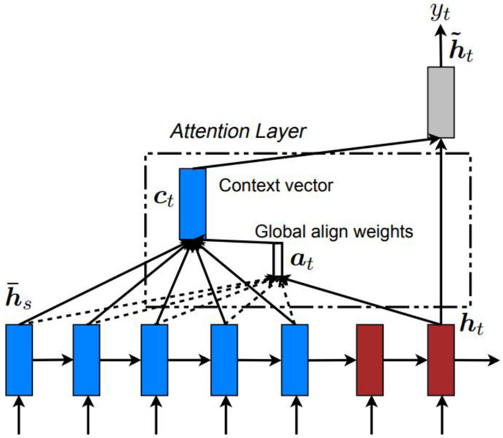


Figure 10. Global attention model illustration Luong et al. 2015.

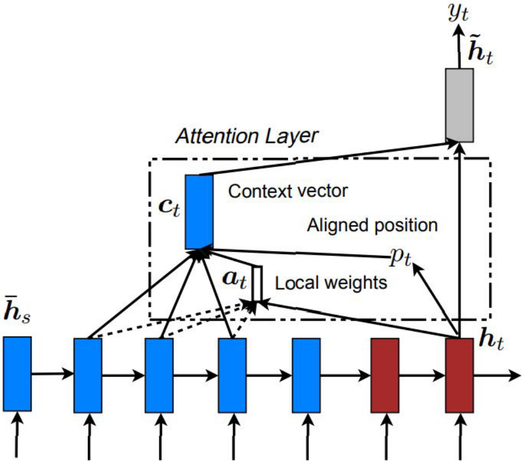


Figure 11. Local attention model illustration (Luong et al., 2015).

Local attention. It was inspired from the tradeoff between the soft and hard attentional models proposed by Xu et al. (2015) to tackle the image caption generation task. The local attention approach addressed some drawbacks of the proposed global attention approach. i.e. the global attention attends to all words on the source side for each target word, the thing that would be computationally expensive and potentially make it difficult for the translation task for long sentences. The model first generates an aligned position p_t for each target word at a time. Then it computes the context vector as a weighted average over the set of source hidden states within the selected window $[p_t-D, p_t+D]$. The model also considers two variants, namely Monotonic alignments which simply assume that the source and target sequences are roughly monotonically aligned; and a Predictive alignment that predicts the aligned position using a sigmoid function. The study demonstrated that both approaches are effective in the WMT translation tasks between English and German in both directions. The models boosted the BLEU score to 5.0 over non-attentional models. For English to German translation, the experimentations achieved new state-of-the-art results for both WMT'14 and WMT'15.

Character-Based Neural Machine Translation by Wang Ling et al., 2015. In this paper, the authors introduced a neural machine translation model that deals with character sequences rather than words. However, due to the importance of word-level information, they kept the use of the word-level information by composing representations of character sequences into representations of words. To implement their proposal, the authors adapted the attention-based neural translation model presented by Bahdanau et al. (2014), to operate over character sequences rather than word sequences while keeping the use of the latter too. For this purpose, they proposed a hierarchical architecture, which replaced the word lookup tables aka the one-hot encoding process and the word softmax with character-based alternatives for both the encoding and decoding parts of the system.

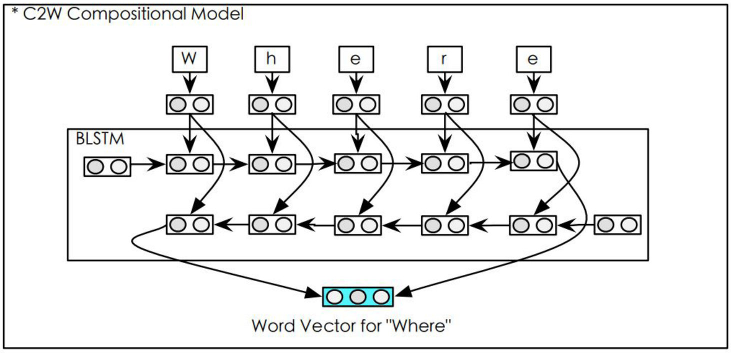


Figure 12. Illustration of the C2W model. Square boxes represent vectors of neuron activations Ling et al. 2015.

Character-based Word Representation. When given a word as an input, the model projects each character into a continuous dimensional vector using a character lookup table. Then, it builds a forward LSTM state sequence by reading the character vectors. Moreover, it reads the character vectors in the reverse order generating the backward states via a backward LSTM. At the final step, all the final states are combined to produce the representation of the word.

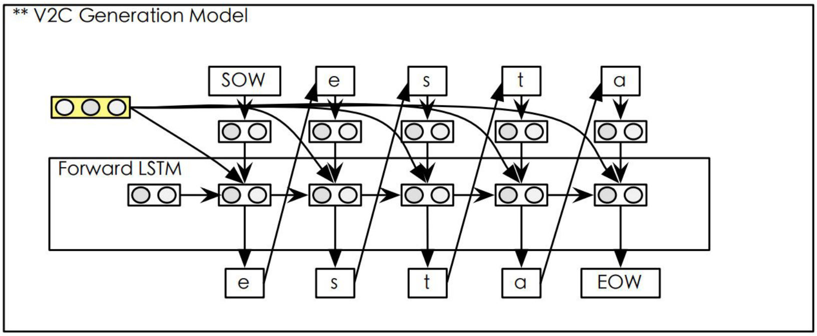


Figure 13. Illustration of the V2C model. Square boxes represent vectors of neuron activations (Ling et al., 2015).

Character-based Word Generation: The character-based word generation model addresses many problems at the word-level neural machine translation, namely, the traversing of the whole target vocabulary for each prediction during both the training and the testing phases, as well as the inability of word-level models to generalize unseen words in the training corpus. The main idea of the character- to- vector approach is that rather than learning to predict single words, the model predicts the character sequence of the output word. Each prediction is dependent on the aligned source word *a*, target word context , and on the previously generated characters. Thus the probability of a given word could be defined as

$$p(w_p|a, l^f_{p-1}) = \prod_{i\in[0,y]} p(w_{j,i}|w_{k,0},\ldots,w_{j,j-1},a,l^f_{p-1}) \qquad (18)$$

The current study showed some minimal improvement from a BLEU score perspective, and according to two experiences each on a different corpus. For the BTEC dataset, the character-based model enhanced the results of the word-based neural model with 0.07 BLEU points. While using the Europarl, the proposed model performed a gain of 0.2 BLEU points over the baseline word-based model. It was fairly noticeable that the differences were not significant, but according to the authors, they still represent a valuable finding compared to previous work.

On using very large target vocabulary for neural machine translation, Jean et al., 2015. This endeavor addressed the problem of handling large corpora in neural machine translation and dealing with learning and decoding complexity in this case. The proposal presented an approximate learning approach to a very large target vocabulary to allow its use of it with a constant training complexity relative to the size of the target vocabulary. Intuitively, the approach proposed to make use of only a small subset of the target vocabulary at each update to avoid the growing complexity of computing the normalization constant. This subset was defined prior to the learning process for each of the partitions obtained from partitioning the training corpus. In more detail, the pre-training process was repeated until reaching the end of the dataset, i.e. sequentially each target sentence in the corpus was examined before training began, and all the unique target words were accumulated until reaching a predefined threshold; then made use of the resulting vocabulary for the equivalent partition during training. By looking at the formula (19) for computing the next target word, we can tell that this approach can be seen as approximating the exact output probability:

$$p(y_t|\, y_{<t}, x) = \frac{\exp(w_t^T \emptyset(y_{t-1}, z_t, c_t) + b_t)}{\sum_{k:y_k \in V'} \exp(w_k^T \emptyset(y_{t-1}, z_t, c_t) + b_k)} \qquad (19)$$

Experiments on the proposed model to matched, and in some cases outperformed the baseline models used in the study. Also, a comparable performance of the state- of- the art results on both the English-German and the English-French translation tasks of WMT'14 was achieved based on BLEU scores using an ensemble of a few models with extensive target vocabularies.

Addressing the Rare Word Problem in Neural Machine Translation, Luong et al., 2015. This proposal addressed a significant weakness of neural machine translation systems, which was their inability to translate very rare words, also known as out-of-vocabulary words, correctly. The authors proposed a method of training a neural machine translation on data that was augmented by the output of a word alignment algorithm. This approach allows the translation model to retrieve the position of the corresponding source sequence for each out-of-vocabulary word in the target sentence by annotating the training corpus with specific alignment information. Then it used the relative position in the post-processing phase to retrieve the equivalent translation and replaced each unknown word in the system's output with a translation of its source word, using either a dictionary or the identity translation. Three strategies are presented in this article:

Copyable Model: In order to represent unknown words in the source and target language, this approach used multiple tokens and assigned repeating unknown words similar tokens,. i.e. the same unknown token has allocated to both the aligned source and the target unknown words, and words with no alignment or aligned to a known word used a special null token.

Positional All Model: This model used the universal token (UNK) and inserted a position token p_d after every word in the target side, where d indicates a relative position to denote the alignment of a target word j and a source word $i = j - d$.

Positional Unknown Model: The only difference to the Position All model was that an $UNKPOS_d$ was used to denote both the positon with respect to its equivalent source word and a word that is unknown.

Luong et al.'s proposition achieved competitive results on the WMT'14 English to French translation task with an improvement of up to 2.8 BLEU points over an identical neural machine translation system that did not use this technique. On the other hand, the Positional Unknown model had the best performance compared to the other methods.

Coverage-Based Neural Machine Translation, Zhaopeng Tu et al., 2015. This work addressed the problems of over-translation (some words were unnecessarily translated for multiple times) and under-translation (some words were wrongly untranslated), resulting from ignoring past alignment information in the standard neural machine translation. The authors proposed a coverage-based approach by introducing a coverage vector to keep track of the attention history in the attention model to guide it through by paying more attention to the untranslated source words. In the neural machine translation context, the coverage-based approach could be applied by intuitively injecting the coverage from the previous step to the attention model. This vector provides complementary information of how likely the source words were translated in the past. This technique would assess the current source sequence if it was heavily attended in the past, then push the attention to the less attended segments of the original sentence instead. The overall performance of linguistic coverage outperformed its NN-based counterpart on both the translation and the alignment tasks (according to the authors, as no results have been presented in the paper), indicating that explicitly linguistic regularities are essential to the attention model.

Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, Melvin Johnson et al., 2016. In this cutting-edge paper, Melvin Johnson et al. proposed a new solution for multilingual translation Figure 14 using a single neural network slightly following the basic neural machine translation architecture i.e. encoder, decoder, attention, and sharing a wordpiece vocabulary. Particularly, this model simply added an artificial token to the input sequence to indicate the required target language (<2es> refers to "translate to Spanish"). The authors focused on three main points, namely simplicity (the approach made use of a general neural machine translation model with no changes), low-resource language improvements (since the training was a cross-languages process where the learned parameters were implicitly shared among different learnt translation pairs) and Zero-shot translation (the training of the model on several language pairs were an implicit application of transfer learning where the model generalized to the translation of unseen language pairs).

On the other hand, the study was concerned with different configurations in order to train the monolingual model. Since the singularity/multiplicity of source/target languages could take various states, the authors assessed three important cases:

Many source languages to one target language: In this case, no additional source token was required since there was only a single target language. Three experiments have been conducted to assess the model for the many- to- one scenario, for instance, the one with WMT datasets, where the German-English and the French-English directions were combined to train the multilingual model. The model outperformed the baseline with +0.16 and +0.23 BLEU points with oversampling, and +0.11, +0.27 BLEU points without oversampling for the first and second directions respectively.

One source language to many target languages: For this scenario, an additional token to specify the target language was prepended to the input. Similar to the previous case, three experiments were conducted, each of which adopted two different translation directions. A significant gain of +0.90 and 0.23 BLEU points has been achieved through the English-Spanish and the English-Portuguese language pairs respectively. However, there were also some cases where the model underperformed the baseline results, namely English-French with oversampling, English-German and English-French without oversampling, and English-Korean.

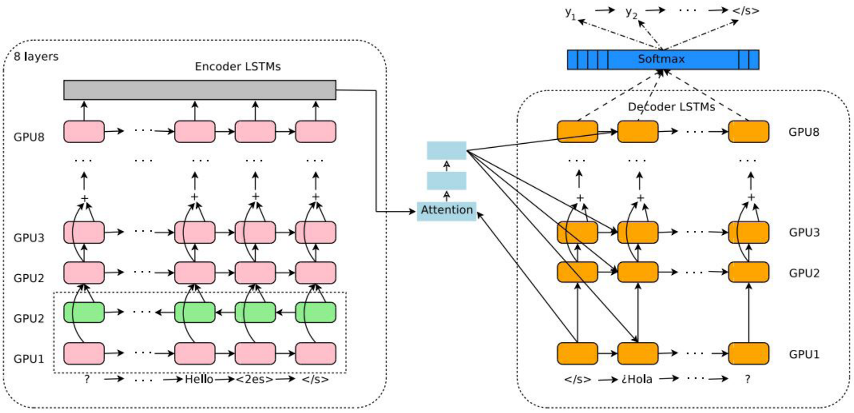


Figure 14. The model architecture of the multilingual GNMT system (Melvin Johnson et al., 2016).

Many source languages to many target languages: As for the one- to- many configurations, a prepended token to the input specifying the target language is needed. However, except the French-English with no oversampling and the English-Spanish pairs, the results of all the remaining directions involved in the experiment underperformed the single baseline models used in the study.
Stanford Neural Machine Translation Systems for Spoken Language Domains, Luong, and Manning, 2015. In this paper, Luong and Manning investigated the effectiveness of applying neural machine translation to spoken language domains. This work considered developing answers as to how to adapt existing NMT systems to a new domain and how to generalize NMT to low-resource language pairs. For this purpose, they trained the attention-based NMT model proposed by Luong et al., 2015, which included two types of attention; global and local.

NMT Adaptation to new domains: The idea of adaptation refers to reusing an existing state of the art system, in this case, an English-German system consisting of 8 individual models trained on WMT data with mostly formal texts, then further training it on a spoken language data (an English-German corpus provided by IWSLT 2015 in this case). The results of adaptation showed to be useful, giving a gain of +3.8 BLEU points compared to using an original model without further training.

Generalizing NMT for low-resource languages: One of the key requirements for training a neural machine translation model is the availability of large amounts of parallel data. However, for some language pairs, the acquisition of large parallel corpora is not handy. In this work, the authors considered applying NMT to the low-resource translation task from English to Vietnamese in IWSLT 2015. The idea was about maintaining a big number of words in the vocabulary, by limiting the preprocessing to a standard corpus tokenization with Moses tokenizer and preserving the casing for words, then replacing those whose frequencies were less than five by the keyword <UNK> (referring to unknown words). The proposal also focused on minimizing the parameters of the deep LSTM, e.g. opt for 2-layer LSTM models with 500-dimensional embedding and LSTM cells. As a result, the system performance was, unfortunately, behind the IWSLT'15 baseline. However, the approach showed to be promising as the gap was only about 0.6 BLEU points.

Applying neural machine translation to Arabic, Amjad Almahairi et al., 2016. Neural machine translation research has mainly focused on Latin languages. To the best of our knowledge, only one study (Amjad Almahairi et al., 2016) had addressed the application of NMT to Arabic language focusing on Arabic - English, and English – Arabic directions. This study reported an improvement by more than +2 BLEU points for the former direction, while for the other direction the proposed method achieved more than +4.98 BLEU upon the phrase-based bassline used in the study. The endeavor also investigated the importance of morphological-oriented and other linguistic-wise preprocessing methodologies for the Arabic language due to its ambiguity. Also by conducting such a preprocessing on the Arabic dataset used in the study, they were able to endorse their hypothesis as the results showed an improvement in comparison to the normal linguistic-less processing.

SUMMARY AND DISCUSSION

The evidence from this study points towards the idea that the neural machine translation approach has typically boosted translation quality, minimized translation errors (Luisa Bentivogli et al., 2016) and improved the translation grammaticality (Neubig et al., 2015) compared to the classical statistical machine translation. This considerable progress can be derived from the recently published results where authors compared their findings using neural models to the results of some statistical machine translation baselines using similar datasets. The comparison is naturally always based on the BLEU metric measurements (Workshop on Statistical Machine Translation'14, '15, '16). On the other hand, Table 1 summarizes some of the main strengths and limitations of both models and gives a clear vision of some advantages of the neural model face to the phrase-based model. One axiomatic advantage could be the fact that the statistical translation system is limited to a minimal local context, where the neural machine translation gives an extensive focus to the entire source sentence contextual information. From another perspective, Google also stated that Google neural machine translation reduced translation errors by up to 85% during tests using Wikipedia and news articles and also produced translations which are closer to human quality than phrase-based translations for different language pairs.

Table 1

A multiperspective comparison that summarizes the strengths and limitations of both statistical and neural machines translation

Neural machine translation	Statistical machine translation
- Generalization via continuous space representation. - Easy implementation of the decoder. - Conditioning the translation on full source context. - Make use of efficient computing strategies (GPU). - No need to store explicit phrase tables and language models.	- Log linear combination of many weak features. - Using only a small context limited to *n* units. - The use of phrase tables allows memorizing the translations of even infrequent words. - Highly intricate decoders.

Table2 presents a general comparison of the main models discussed in this paper, based on some positive and negative points that we were able to observe in the papers covered. We noticed that the number of work presented in the scope of neural machine translation is exhaustive, diverse and progressive. Moreover, each study tries to tackle a specific problem in neural translation and capitalizes on previous studies, giving more chances to improve the baseline systems and enhance the translation quality. The recent proposed approaches of recurrent, attention-based and character-based translation models have some similarities in the core concept of modeling the translation task as an end-to-end process, but differ in their architecture and global configurations, since they are usually parameterized according to the problems. Currently, the largely adopted architecture is the attention-based model due to its ability to handle long sentences, make use of bidirectional encoders instead of fixed-lentgh vectors, and its efficient soft alignment strategy.

Table 2

A General Comparison of the Cutting-Edge Methods in the Neural Machine Translations

Model	Pros	Cons
Recurrent neural machine translation (Sutskever et al., 2014).	-Direct end-to-end training. -Use of minimal domain knowledge. -Generalizing to new word phrases and sentences that do not occur in the training set.	-Inefficient generalization of rare words. -Making use of a fixed length vector to represent variable length sentences which may cause overfitting in some cases.
Attention-based neural machine translation (Bahdanau et al., 2014 ; Luong et al., 2015).	-Handling long sentences in the translation process. -Overtaking the use of the fixed-length vector in the encoding process. -Soft alignment strategy is easier to train than the hard attention approach.	-An extensive number of epochs is relevant to find the potential attention coefficients. -Overfitting for small datasets due to the absence of domain knowledge regarding word alignment.
Character-based Neural Machine Translation (Wang Ling et al., 2015).	-The capability of interpreting and generating unseen word forms. -Overtaking the preprocessing and tokenization of the source and target languages. -Reducing the source and target vocabulary size.	-The use of characters as atomic units tends to significantly complicate the training.

The last three studies covered in the previous section could be excellent examples of applying the neural translation model to various domains. Looking at both of them we can tell that despite that sequence-to-sequence models being standard, the choice of the neural architecture is always dependent on the context of the study, i.e., some languages would require a morphological-wise preprocessing as in the case of the Arabic language in the section above, and the new annotation concept to enable multilingual translation, where others would not.

Despite of the big success of applying neural networks to natural language processing, the use of artificial neural networks for translation ends, still raises many questions, especially, regarding the high efficient use of syntactic or semantic structures, making use of more appropriate monolingual data for domain adaptation, handling large vocabularies and dealing with low-resource language pairs. As a response to these issues, some studies have outlined novel solutions. For instance, a few studies proposed to combine neural machine translation with a separately trained language model (Gülçehre, Firat et al., 2015), a method in which the quality had improved up to +1.96 BLEU points on low resource language pairs like Turkish-English, and +0.39 and +0.47 BLEU points by extending the method to tremendous resource languages such as Czech-English and Dutch-English. Alternatively, other studies put forward new approaches to merge neural and statistical machine translation (Auli et al., 2013, Li, Liu, and Sun, 2013, Devlin et al., 2014). Others have suggested enhancing neural translation systems with useful statistical translation features (Wei He, Zhongjun He, Hua Wu, and Haifeng Wang, 2016).

CONCLUSION

Neural machine translation proved to be a cutting-edge improvement in the field of statistical translation. In this article, we have aimed to establish an overview of statistical translation starting from the classical statistical machine translation to the emerging approach of neural machine translation. Moreover, we surveyed some interesting endeavors that contributed generously and with remarkable results in developing this field. Through this study, we aimed to develop our understanding of the literature of statistical translation and paradigms of analysis that allow us to perceive the different aspects of the field. Moreover, moved forward with our future work that aligns with the scope of dealing with minimal corpora and limited resources language pairs in neural translation.

ACKNOWLEDGMENT

We would like to acknowledge the National Center for Scientific and Technical Research (CNRST) for supporting this study under the framework of Merit Scholarship Ref: L 02/13.

REFERENCES

Almahairi, A., Cho, K., Habash, N., & Courville, A. (2016). *First result on Arabic neural machine translation*. Retrieved from:https*://arXiv preprint arXiv/1*606.02680.

Auli, M., Galley, M., Quirk, C., & Zweig, G. (2013, October). Joint language and translation modeling with recurrent neural networks. In EMNLP, *3*(8), 1-10.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). *Neural machine translation by jointly learning to align and translate*. Retrieved from:https://arXiv preprint arXiv:1409.0473.

Bentivogli, L., Bisazza, A., Cettolo, M., & Federico, M. (2016). *Neural versus phrase-based machine translation quality: a case study*. Retrieved from:https://arXiv preprint arXiv:1608.04631.

Berger, A. L., Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Gillett, J. R., Lafferty, J. D., ... & Ureš, L. (1994, March). The Candide system for machine translation. In *Proceedings of the workshop on Human Language Technology,* 157-162. Association for Computational Linguistics.

Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, *22*(1), 39-71.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., ... & Soricut, R. (2014, June). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation* (pp. 12-58). Association for Computational Linguistics Baltimore, MD, USA.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., ... & Post, M. (2015). *Findings of the 2015 Workshop on Statistical Machine Translation*.

Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade* (pp. 421-436). Springer Berlin Heidelberg.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., ... & Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, *16*(2), 79-85.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, *19*(2), 263-311.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, *33*(2), 201-228.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). *On the properties of neural machine translation: Encoder-decoder approaches*. Retrieved from:https://arXiv preprint arXiv:1409.1259.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. Retrieved from:https://arXiv preprint arXiv:1406.1078.

Chung, J., Gülçehre, C., Cho, K., & Bengio, Y. (2015). *Gated feedback recurrent neural networks*. CoRR, abs/1502.02367.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R. M., & Makhoul, J. (2014, June). Fast and robust neural network joint models for statistical machine translation. In *ACL* (*1*), 1370-1380.

Germann, U., Jahr, M., Knight, K., Marcu, D., & Yamada, K. (2001, July). Fast decoding and optimal decoding for machine translation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics* (pp. 228-235). Association for Computational Linguistics.

Gulcehre, C., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H. C., ... & Bengio, Y. (2015). *On using monolingual corpora in neural machine translation*. Retrieved from: https://arXiv preprint arXiv:1503.03535.

Hang Luong, Kyunghyun Cho, & Christopher. (2016). *Manning Neural Machine Translation – Tutorial ACL 2016*

Hasan, S., Quo, T. S., & Shamsuddin, S. M. (2012). Artificial fish swarm optmization for multilayer network learning in classification problems. *Journal of Information & Communication Technology*, *11*, 37-53.

He, W., He, Z., Wu, H., & Wang, H. (2016, February). Improved neural machine translation with SMT features. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jean, S., Cho, K., Memisevic, R., & Bengio, Y. (2014). *On using very large target vocabulary for neural machine translation*. Retrieved from: https://arXiv preprint arXiv:1412.2007.

Jean, S., Firat, O., Cho, K., Memisevic, R., & Bengio, Y. (2015, September). Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, 134-140.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., ... & Hughes, M. (2016). *Google's multilingual neural machine translation system: Enabling zero-shot translation*. Retrieved from:https://arXiv preprint arXiv:1611.04558.

Kalchbrenner, N., & Blunsom, P. (2013, October). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1700–1709, Seattle, Washington, USA: Association for Computational Linguistics.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. (2015). *Show, attend and tell: Neural image caption generation with visual attention*. Retrieved from:https://arxiv.org/pdf/1502.03044.pdf

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., & Dyer, C. (2007, June). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions* (pp. 177-180). Association for Computational Linguistics.

Koehn, P., Franz Josef Och & Daniel, M. (2003). Statistical phrase-based translation. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*-Vol.1. Association for Computational Linguistics.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444.

Li, P., Liu, Y., & Sun, M. (2013, October). Recursive autoencoders for ITG-based translation. In *EMNLP* (pp. 567-577).

Ling, W., Trancoso, I., Dyer, C., & Black, A. W. (2015). Character-based neural machine translation. Retrieved from:https://*arXiv preprint arXiv*:1511.04586.

Luong, M. T., & Manning, C. D. (2015). Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation.*

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. Retrieved from:https://*arXiv preprint arXiv*:1508.04025.

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv*:1508.04025.

Luong, M. T., Sutskever, I., Le, Q. V., Vinyals, O., & Zaremba, W. (2014). Addressing the rare word problem in neural machine translation. Retrieved from:https://*arXiv preprint arXiv*:1410.8206.

Marcu, D., & Wong, W. (2002, July). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*-Volume 10 (pp. 133-139). Association for Computational Linguistics.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115-133.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems,* 3111-3119.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, *13*(6), 47-60.

Neubig, G., Morishita, M., & Nakamura, S. (2015). Neural reranking improves subjective quality of machine translation: NAIST at WAT2015. *arXiv preprint arXiv*:1510.05203.

Och, F. J., Ueffing, N., & Ney, H. (2001, July). An efficient A* search algorithm for statistical machine translation. In *Proceedings of the workshop on Data-driven methods in machine translation*-Volume 14 (pp. 1-8). Association for Computational Linguistics.

Och, F. J., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, *30*(4), 417-449.

Och, Franz Josef, & Hermann, N. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics*, 30 (4), 417-449.

Olanweraju, R. F., Aburas, A. A., Omran, O. K., & Abdalla, A. H. H. (2010). Damageless digital watermarking using complex valued artificial neural network. *Journal of Information and Communication Technology*, *9*, 111-137.

Philipp, K. (2004a). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proceedings of the Sixth Conference of the Association for Machine Translation in the Americas*, 115–124.

Ranzato, M. A., Poultney, C., Chopra, S., & LeCun, Y. (2006, December). Efficient learning of sparse representations with an energy-based model. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, 1137-1144. MIT Press.

Shamsuddin, S. M., Zainal, A., & Mohd Yusof, N. (2008). Multilevel kohonen network learning for clustering problems. *Journal of Information and Communication Technology*, *7*, 1-25.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems,* 3104-3112.

Tillmann, C., & Ney, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics*, *29*(1), 97-133.

Tu, Z., Lu, Z., Liu, Y., Liu, X., & Li, H. (2016). *Coverage-based neural machine translation*. Retrieved from:https://arxiv.org/pdf/1610.05150.pdf

Wang, Y. and A. Waibel. (1997). Decoding algorithm in statistical machine translation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the*

European Chapter of the Association for Computational Linguistics (A CL/EA CL ‘97), 366-372, Madrid, Spain. Watanabe, T., & Sumita, E. (2003, September). Example-based decoding for statistical machine translation. In *Machine Translation Summit IX*, 410-417.

Zhang, J., Utiyama, M., Sumita, E., & Zhao, H. (2014). Learning hierarchical translation spans. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (EMNLP), 183–188,