A Survey on Neural Machine Translation

Prasanna Kumar R[1], Ippatapu Venkata Srisurya[2], Madisetty Tharun Kumar[2], Yash Upadhyay[2]

[1].Assistant Professor-SG Department of computer science and engineering, [2].B Tech-AI

Amrita School of engineering, Amrita Vishwa Vidhyapeetam.

[r\_prasannakumar@ch.amrita.edu,](mailto:r_prasannakumar@ch.amrita.edu,) [ch.en.u4aie20024, ch.en.u4aie20067, ch.en.u4aie20079, @ch.students.amrita.edu](mailto:ch.en.u4aie20024@ch.students.amrita.edu)

***Abstract* - Natural language processing is among the emerging fields in machine learning and deep learning. Neural machine translation is a subfield of Natural language processing that focuses on language translation. In this paper, the different methods in Neural machine translation were discussed along with its architecture. It starts from traditional Neural machine translation techniques that gives poor performance when it encounters long sentences and problems related to vocabulary . Attention-Based Neural machine translation can provide better performance for long sentences, but the problem of vocabulary remains the same. This can get solved by Attention-based Neural machine translation along with sub-word segmentation. At last, some of the essential models developed in recent times were shown.**

*Index Words -* Encoder-Decoder, Linguistic features, Long Short-term memory, Recurrent Neural Networks,

1. **Introduction**:

Warren Weaver invented neural machine translation in 1949. Until the 1980s, machine translation was primarily based on rule-based machine translation. The translation was done using dictionaries and grammar to examine linguistic information about both the source and target languages. However, advances in statistics and statistical models have developed a new methodology known as statistical machine translation. Ramon Neco and Mikel Forcada gave the idea of an Encoder and decoder structure for machine translation in 1997. And models based on neural networks were developed by Yoshua Bengio in 2003, which improved the data sparsity problem of traditional SMT models.[1]

Translation involves maximizing the conditional probability of y given x p(y|x), where y is the target phrase and x is the source phrase. There is an encoder and a decoder in machine translation. The Encoder's function is to encode the provided text into a vector representation, which the Decoder then decodes onto the target language. The whole process of this encoding and decoding is discussed in other sections. As a whole, this process is trained jointly to maximize the above probability.[2]

This probability distribution of the NMT is the most commonly used method.

1. **Neural Machine translation**:

**2**.**1 Recursive neural networks**:

In NMT, since both the source and the target sentences are variable in length, standard Artificial neural networks will not work. Recursive neural networks (RNN) were created in the 1980s, but in recent development in computation power along with the massive amounts of data available today, RNN came to use. This neural network is used for sequential data, which is an ordered data in which related things follow each other. Example Deoxyribose Nucleic Acid (DNA) sequence. In these networks, the decision is based on the current and also the previous state. Below is the figure representing a neural network.

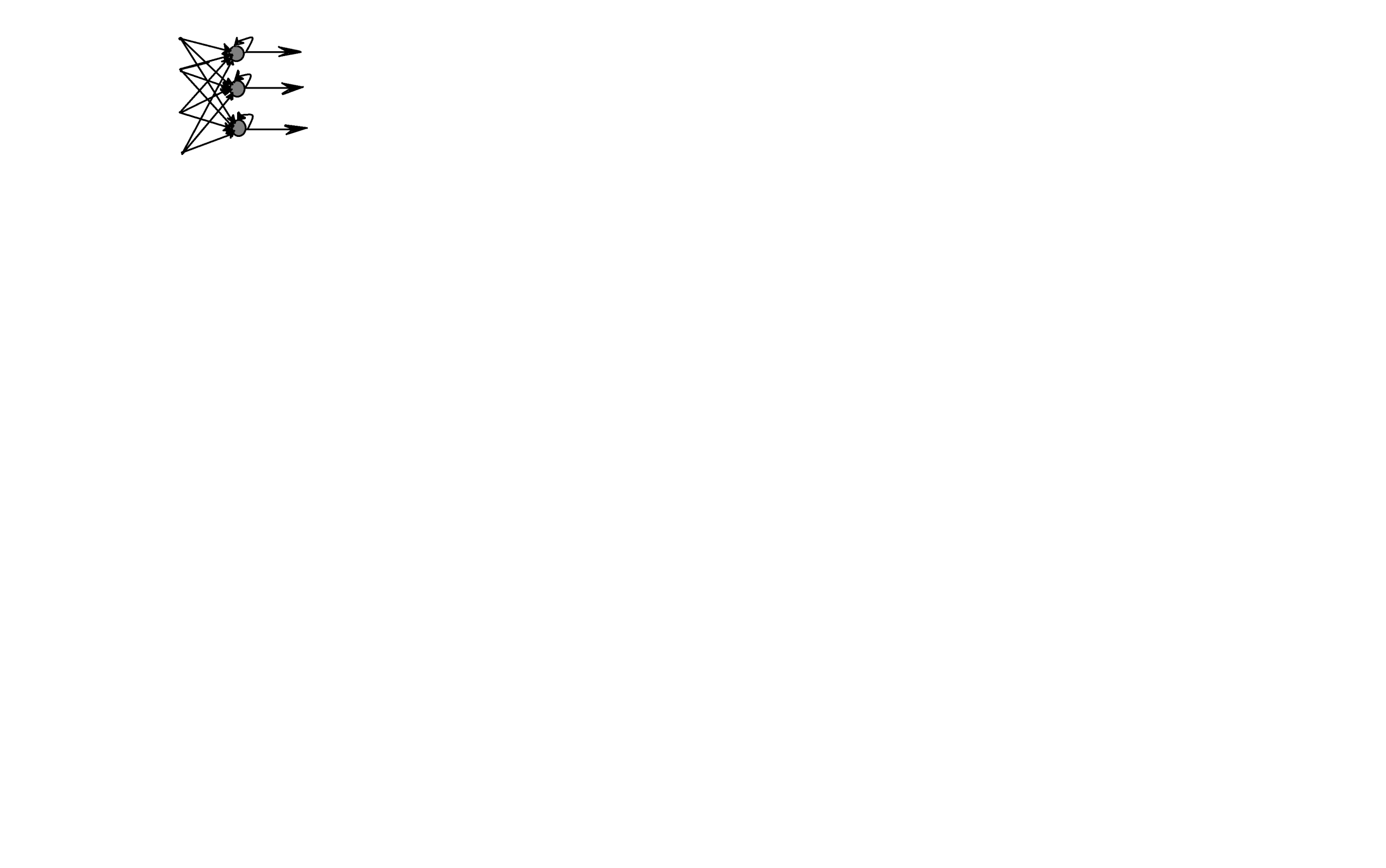


Fig-1 RNN

This neural network is required for machine translation because, to translate a word from one language to another, there is a necessity of understanding the phrase's meaning, which may be deduced from its context. For this, the idea about the previously used words becomes essential, which is possible with RNN. But the disadvantage with this RNN is, it will have a short-term memory. It can remember only a few before states, which is challenging in translating long sentences. So, a new method known as Long Short-term memory was introduced [4].

**2.2 Long Short-term memory (LSTM):**



Fig-2 LSTM

The above figure is an LSTM neuron in which there are two functions, sigmoid and tangenth functions, and gates, which makes this method intelligent by storing the words in long sentences. Notice that in this method, all the methods are not stored. Only those words are stored, which will have an impact in the future. The neuron will be able to detect them from the training phase.

In an LSTM, there will be three gates, memory cell, and candidate memory. Candidate memory records the information of the present state, and in the memory cell, the basic information was updated by adding necessary data and removing unnecessary data [4,5].

1. **Forget Gate:**

This gate ensures how much information needed to be erased. This gate will perform element-wise multiplication between the cell state and the previous state. This gate will have a sigmoid function that ensures that output values are continuous and lie between 0 and 1.

1. **Input Gate:**

Input gate uses two functional units. The first functional unit is a tangenth activation function that gives values in between -1 and 1 which decides the change of the cell state. The second function uses a sigmoid activation function. It is responsible for the magnitude of the change in cell state. After multiplying the results obtained in these two functional units, the input gate adds the result to the cell state, and thus updates the cell state.

1. **Output Gate:**

The range of cell memory lies between -1 and 1 using the function to compute the current hidden state. Output gate controls what to output in the hidden state at this point.

**2.3 Encoder:**

The Encoder is an application of RNN whose state changes each time the neuron hears a new word in a sentence and, at the end, returns the summary vector of the entire sentence. First, the Encoder takes the one word from the sentence and gives the one-hot vector of that word. Later this vector is represented on a low dimension space continuous vector representation denoted by si. This si can be seen as the si=E\*wi

E is the word embedding matrix where the columns' size is the same as the vocabulary size. And wi is the one-hot vector of the word. In the next step, the RNN has updated itself as a new word is seen in the sentence.

hi = f (si, hi-1) where f is a non-linear transformation function.

After processing the last word of the sentence, the vector generated is the summary vector of the whole sentence [6].

**2.4 Decoder:**

The Decoder is also an application of RNN that takes input as a summary vector which is the output of the encoder, previously generated target word, and the last hidden state. After processing the input, the probability distribution of the words over the target language vocabulary is obtained. Target words are then sampled from this probability distribution. The first internal state of the Decoder is calculated by the function zi=f ` (ht, zi-1, ui-1), where zi and zi-1 are the present and previous states of the Decoder. Ht is the summary vector, and ui-1 is the previously generated targeted vector.

Next, a score is calculated for the current state of the Decoder, and later this score is converted into probability. This score will be high if it aligns with the decoder well else low. From the probability distribution obtained from step 2, we sample the target word. The Decoder then repeats the preceding stages until no end-of-sentence is detected. As a result, a target sentence is created that corresponds to the input source sentence.

The diagram below depicts the Encoder and Decoder.

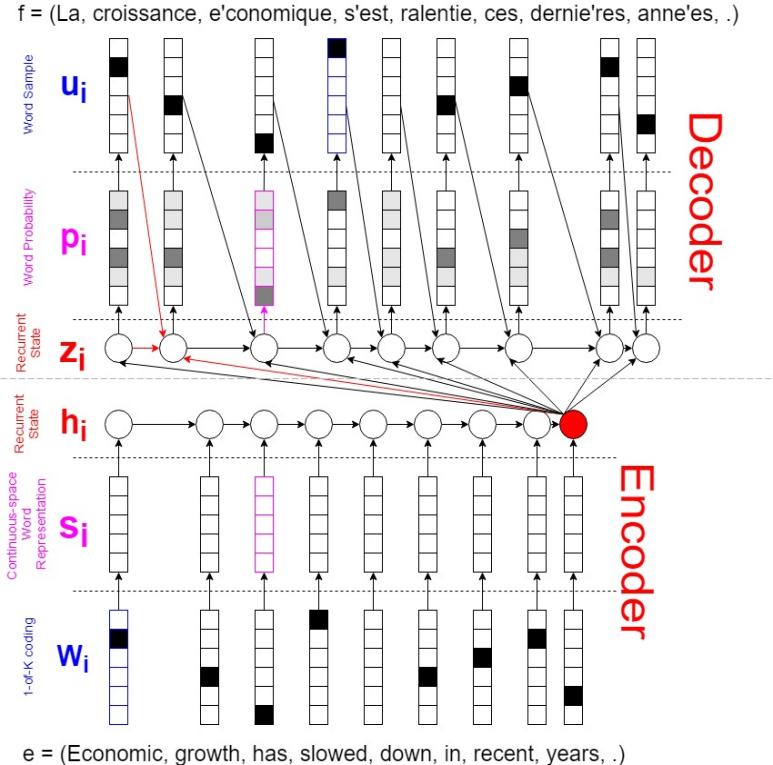


Fig-3 Encoder-Decoder

In recent years the attention-based Encoder gave better performance than traditional Encoder and Decoder. The difference in both of these is the encoder part. In primary Encoder, the gives the summary vector, but the attention-based Encoder gives the context vector. In this Encoder, each word is transformed into an annotation vector, and by taking convex combinations of these annotations, vectors give the context vectors. The coefficients in these combinations are called attention weights which are calculated during a training phase. To compute the attention weight, we need an alignment score defined as how relevant an I th source word to j the word [6].

1. **Subword NMT:**

All the above-discussed models are trained on limited size, but usually, different sets of words are not in the vocabulary in the training set. This problem is called off out of vocabulary problem. Our model should stop if a new word were seen in the testing time, so it is necessary to solve it. Most of these words are named entities like person names, compound words, cognates. To address each issue, we need different approaches. Named entities are probably the names of a person, animals, places, so direct copying the same from the source text to target text will work if both share common symbols. Else there is a need for some transliteration, but it is easy to sort this issue. Next is the Cognates, which are loan words; therefore, character level transliteration will get the work done. Translation of compound words was done by translating their morphemes separately [7].

**3.1Byte-Pair encoding[BPE]**:

In Byte-Pair encoding, there will be two datasets, a training set, and a symbol set, in this method. Words which are in the training vocabulary are represented by a string of letters and a tag </w> which represents end of word. All of the characters have been added to the symbol vocabulary. Using the BPE technique, the most frequent symbol pair is identified, and all its occurrences are merged, producing a new symbol added to the vocabulary. Every merge operation produces a new symbol that represents a character n-gram. Recurring character n-grams (or whole words) are eventually merged into a single symbol. The final symbol vocabulary size is equal to the start vocabulary size plus the number of merging operations - the latter being the algorithm's lone hyperparameter. For example, let us consider a corpus.

‘l o w </w>’: 5

‘l o w e r </w>’ :2

‘n e w e s t </w>’ :6

First, the vocabulary has to be generated using a general regex expression. Let us consider

Vocab={</w>, d, e, I, l, n, o, r, s, t, w}

Now search for the most frequent character sequence in the corpus by the algorithm and add it to our vocabulary. For example, let's consider 'o w' is the most frequent word, these words get added to the vocabulary, and we ignore it in the corpus. The next iteration, 'l o w,' is the more frequent character sequence that gets appended to the vocabulary list. This algorithm is performed on an extensive training set; therefore, when a new word is introduced to the model like lowest, this can be matched to the word low since we have it in the vocabulary, which avoids zero probability [7].

1. **Linguistic features in NMT:**

**4.1 Models:**

The use of various linguistic features in our daily lives, which are lemma, POS tags, dependency parsing labels, and sub-word tags, can be included in NMT models, which improves the performance of NMT over rare words can also be the solution for out of vocabulary problem. We can estimate the unknown or the rare word which the model doesn't come through in its training phase by lemma. Subword tags help the model understand that these are not independent words but give extra information about the whole word. POS tags help in the situation when the same word has different meanings in different sentences.

To incorporate these features, we have to embed them onto the continuous space representation of the words [8].

**4.2 Lemma:**

This technique provides sharing of the information by the words having the exact base words. These types of words are represented as similar points in the word vector space. The help of this technique provides more efficiency when trained over large data sets. The problem of OOV can be solved to some extent. It is different from the Stemming algorithm. In stemming, some part of the word is chopped off. The number of words chopped off is decided by the algorithm but not knowing the meaning. But in the process of lemmenting, the algorithm will refer to the dictionary before reducing it to the base word.

1. **Priming Neural Machine Translation:**

Priming is the phenomenon in which the response to one stimulus influences the response of the other. The use of this phenomenon in machine translation can be explained with an example. I was in the bank yesterday. While translating it to the target sentence, the model needs to know whether the user was in the river bank or a financial body.

At this type of ambiguity type of sentence, the concept of primer will help. We will develop the primers for specific words in the training phase to decide the meaning based on the sentences he used before and after [13].

This can be done by creating a translation model incorporated for similar translation from translation memory, increasing the translation accuracy. This model has two main processes Similarity Computation and Priming schemes.

1. **Bidirectional interface:**

The section says that a primer can be developed at the training phase by reading the sentences before and next to the respective sentence. But in the traditional models will not achieve this. So unique models were proposed [14] called as Bidirectional interface in which the Decoder does the job from left to right and from right to left.

This will increase the performance of the model. They are 4 types in this—an agreement between L2R and R2L, Rescore with bidirectional decoding, asynchronous and synchronous decoding.

Theoretically, the probability of the context of the word in the L2R and R2L are the same. Based on these assumptions, a scheme in which one model's direction (L2R or R2L) regularizes the other. The type is rescoring the output of the L2R Model by R2L, using the output of the R2L to optimize the L2R model.

1. **Conclusion:**

In this paper, a detailed description of neural machine translations was discussed. The concepts about RNN, encoder-decoder, Sub words, linguistic features in NLP were also discussed. Although much development in this field has occurred, some of the main hurdles that NMT models include are word vocabulary, long sentence translations, etc. Of course, Byte pair encoding solves the out-of-word vocabulary to some extent but not entirely. And for the long sentence translation, long short-term memory can handle up to some extent. There will be a word limit for translation. In recent years unsupervised techniques were introduced in NMT, which by which the efficiency of the model increases day by day.[15] In recent years the concept of transformers was introduced, which are very efficient in machine translation. With the help of this sequencing, sentences parallel can be possible, reducing so much of time [16].

**References:**

1. [“Neural Machine Translation: A Review of Methods, Resources, and Tools.” 2020.](http://paperpile.com/b/tCHFXB/ZnUp) *[AI Open](http://paperpile.com/b/tCHFXB/ZnUp)* [1 (January): 5–21.](http://paperpile.com/b/tCHFXB/ZnUp)
2. [“Neural Machine Translation by Jointly Learning to Align and Translate.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/Bx6w) <https://arxiv.org/pdf/1409.0473.pdf>[.](http://paperpile.com/b/tCHFXB/Bx6w)
3. [“On the Properties of Neural Machine Translation: Encoder–Decoder Approaches.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/klLv) <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.741.1291&rep=rep1&type=pdf>[.](http://paperpile.com/b/tCHFXB/klLv)
4. [“Neural Machine Translation of Rare Words with Subword Units.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/OzFh) <https://arxiv.org/pdf/1508.07909.pdf>[.](http://paperpile.com/b/tCHFXB/OzFh)
5. [“Linguistic Input Features Improve Neural Machine Translation.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/CMXh) <https://arxiv.org/pdf/1606.02892.pdf>[.](http://paperpile.com/b/tCHFXB/CMXh)
6. [“Unsupervised Neural Machine Translation.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/u08R) <https://arxiv.org/pdf/1710.11041.pdf>[.](http://paperpile.com/b/tCHFXB/u08R)
7. [“The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/Fs9q) <https://arxiv.org/pdf/1804.09849.pdf>[.](http://paperpile.com/b/tCHFXB/Fs9q)
8. [“Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network.” 2020.](http://paperpile.com/b/tCHFXB/QyqP) *[Physica D. Nonlinear Phenomena](http://paperpile.com/b/tCHFXB/QyqP)* [404 (March): 132306.](http://paperpile.com/b/tCHFXB/QyqP)
9. [Pham, Minh Quang, Jitao Xu, Josep M. Crego, François Yvon, and Jean Senellart. 2020. “Priming Neural Machine Translation.” In](http://paperpile.com/b/tCHFXB/Ia5Z) *[Proceedings of the Fifth Conference on Machine Translation](http://paperpile.com/b/tCHFXB/Ia5Z)*[, 516–27.](http://paperpile.com/b/tCHFXB/Ia5Z)
10. [“Long Short-term Memory RNN.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/S31i) <https://arxiv.org/pdf/2105.06756.pdf>[.](http://paperpile.com/b/tCHFXB/S31i)
11. [“Scalable Transformers for Neural Machine Translation.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/leHE) <https://arxiv.org/pdf/2106.02242.pdf>[.](http://paperpile.com/b/tCHFXB/leHE)
12. https://medium.com/syncedreview/history-and-frontier-of-the-neural-machine-translation-dc981d25422d
13. [“Long Short-Term Memory.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/bzOU) <https://ieeexplore.ieee.org/abstract/document/6795963>[.](http://paperpile.com/b/tCHFXB/bzOU)
14. [“Literature Survey: Neural Machine Translation.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/B2Al) <https://www.cfilt.iitb.ac.in/resources/surveys/NMT-survey-paper.pdf>[.](http://paperpile.com/b/tCHFXB/B2Al)
15. [“Handling Out-of-Vocabulary Words and Recognition Errors Based on Word Linguistic Context for Handwritten Sentence Recognition.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/xK1W) <https://ieeexplore.ieee.org/document/5277628>[.](http://paperpile.com/b/tCHFXB/xK1W)
16. [“Neural Machine Translation: A Review of Methods, Resources, and Tools.” n.d. Accessed July 12, 2021.](http://paperpile.com/b/tCHFXB/L2DR) <https://arxiv.org/pdf/2012.15515.pdf>[.](http://paperpile.com/b/tCHFXB/L2DR)