**2.4 Language Translation**

One of the earliest goals for computers was the automatic translation of text from one language to another. Machine translation is perhaps one of the most challenging artificial intelligence tasks given the fluidity of human language. Earliest, rule-based systems were used for this task, which were replaced in the 1990s with statistical methods.

More recently, neural network models have come to achieve state-of-the-art results in this field that is aptly named neural machine translation (NMT). Machine translation is the task of automatically converting source text in one language to text in another language. Let see how language translation work using; (a) Word-for-word translation. (b) Neural Machine translation (Neural networks)

**2.4.1 Word-for-word translation**

Word-for-Word translation involves matching every single word in the source language sentence and finding the corresponding word in the target language.

To do word-for-word translation, all we need is an accurate database with our targeted language and then match every word with it. For instance let say we are translating from English to French, for every English word, we simply look up the corresponding French word in the database. We repeat this process for every word in the sentence.

For example, we want to translate the sentence “**How old are you”** from English to French language. First we will first take every word in the English sentence for every word in the corresponding French word translation then split it out and repeat this foe every word in the sentence.

Word-word translation

How old are you?

Comment vieux sont vous

English Sentence

French Sentence

|  |  |
| --- | --- |
| How | Comment |
| Old | Vieux |
| Are | Sont |
| You | Vous |

This approach of translating is simple and easy to implement, but generally does not construct proper sentences, as seen above. Generally, Languages are composed of two important components:

1. **Token**: which is the smallest unit of a language
2. **Grammar:** which defines how this token should appear to make sense (ordering of tokens).

Every word in a sentence is a token. If languages were only constructed with tokens and grammar did not matter, then the word-for-word translation approach would be good enough and the problem of language translation would be easily implemented. In the example “How old are you” have four (4) words token.

Grammar is basically a guide or a set of rules that define the ordering for these words.

1. Adjectives follow adverbs
2. Nouns follow adjectives
3. Conjunctions can link two (2) ideas

Unfortunately, grammar is the key in making sense of sentences. Grammar must be incorporated into a translator’s logic. In order to incorporate grammar, there are things needed to consider:

1. Syntax Analysis: These are basic structure, it is basically asking the question, “Does this structure of the sentence look correct”
2. Semantics analysis: Semantic analysis basically deals with meaning. And it asks the question does the sentence make sense in context. If we don’t follow this we will just be outputting gibberish.

Instead of explicitly defining our grammar; we can use another approach (Using Neural Network) and let the machine neural network do it for us. Neural network are components that learn to solve problem by learning from hundreds and thousands of examples (Woodford, 2020).

## 2.4.2 Neural Machine Translation (Neural Networks)

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems mentioned above. It is a machine translation approach that applies a large artificial neural network toward predicting the likelihood of a sequence of words, often in the form of whole sentences. Unlike word-for-word translation, neural machine translation, NMT, trains its parts end-to-end to maximize performance.

Neural network learns to solve problems by looking a vast amount of examples. Thus, they can be used to define grammar for a translator. They are trained with language patterns and eventually are able to translate a given English sentence into French all on their own.

Training sample input

Training sample output

**Neural Network**

Comment vieux sont vous

How old are you

***Fig. 2.2: Simple neural network translator***

With our example of English to French translation of the sentence “**How old are you**”, a neural network takes an English sentence or sequence of words as an input and gives a French sentence or sequence as an output. In order for this input to be interpreted by a neural network, it will first need to be converted into a format it understands, i.e. a vector or matrix.

Vectors and matrices are collection of numbers representing data. This conversion from sentence to vector is called the Vector Mapper, and it is the first part of the network.

English input

**Neural Network**

0.93

0.16

0.61

0.59

0.46

Vector

***Fig. 2.3: Simple Vector Mapper***

Since translator deals with sequences of words or sentences, Recurrent Neural Network (RNN) can better up the translator. RNNs are networks that learn to solve problems that involve sentences (Amidi & Amidi, 2020). When our English sentence is translated into vector, it needs to be translated into a French sentence. This vector mapping is done with a second neural network. Since we are dealing with sentences, another RNN can be used. Together, these two neural networks make the basic foundation of a language translator which is called the Encoder-Decoder Architecture.

English

Input

Recurrent Neural

Network

**Vector**

0.93

0.61

0.59

0.46

Recurrent Neural

Network

French

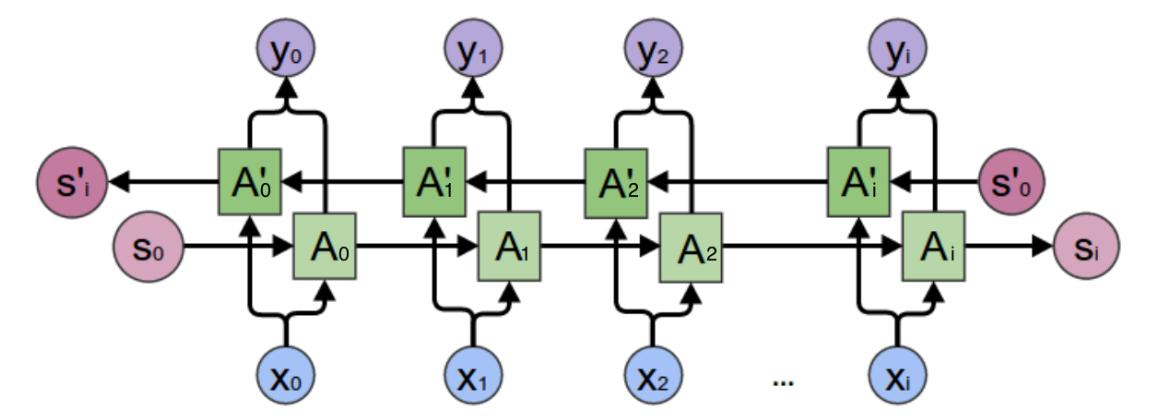
Output

***Fig. 2.4: Encoder-Decoder Architecture***

Recurrent Neural Network (RNNs) is designed to take sequences of text as inputs or return sequences of text as outputs, or both. They are called recurrent because the network’s hidden layers have a loop in which the output and cell state from each time step become inputs at the next time step. This recurrence serves as a form of memory. It allows contextual information to flow through the network so that relevant outputs from previous time steps can be applied to network operations at the current time step. These RNNs are called “Long Short-Term Memory Recurrent Neural Networks” (LSTM-RNN). LSTM networks are capable to deal with longer sentences fairly well.

The encoder-decoder architecture works well for medium-length sentences (around 15–20 words). Nevertheless, LSTM-RNN encoder-decoder structures do not fair as well with longer sentences. RNNs are not able to address the complexity of grammar in longer sentences. RNNs use persisted past information to make decisions about the present. This means that while translating the 8th English word to French, the RNN looks back to the previous 7 words that were translated to make a decision. However, in language, a word depends not only on the words that come before in a sentence but also the words after as well. In order to look in both directions, forwards and backward, a normal RNN is replaced with a bi-directional recurrent neural network.

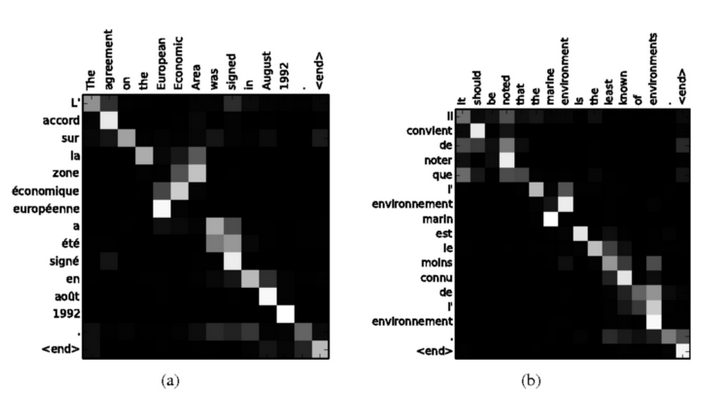
## 2.4.3 Bi-Directional Recurrent Neural Network

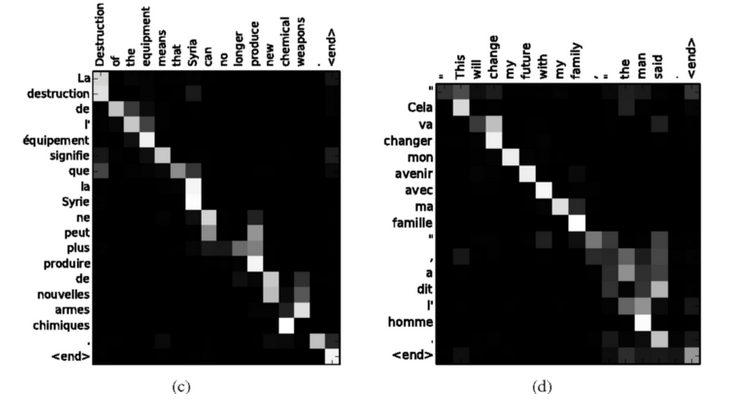


***Fig. 2.5: Bi-directional recurrent neural networks structure*** (Programmersought , 2016)

Bi-directional recurrent neural networks were introduced in 1993 but gained popularity recently with the emergence of deep learning. Since we are performing English to French translation, while joining some word in the French translation, we are looking at words that come before it and words that come after it. With a bi-directional network, we are able to do this. This solves a big problem but also brings up a new issue: is every word in a sentence pivotal to the structure of the previous and next word? Which words should we focus on more in a large sentence?

(Bahdanau, Cho, & Bengio, 2016) Introduced a method to figure this out by learning to jointly align and translate words. The alignment refers to the order of the words as well as an individual word’s weight in affecting previous and post words.





***Bahdanau et al., (2016)***

In the figure above the vertical axis are the French words and the horizontal axis are the English words. The squares shaded from black to white represent the weight of the alignment. White squares are the words that need to have more emphasis on and affect their surrounding word structure. This alignment is taught to an extra unit called an “Attention Mechanism” Bahdanau *et al., (2016).* It learns which English words to focus on while generating the words of the French translation. It sits between the encoder and decoder.

## 2.4.3.1 How Does Bi-Directional RNN model works

An English sentence is fed to an encoder. Then the encoder translates the sentence to a vector (numbers). After which the vector is sent to the Attention Mechanism (AM). The AM decides which French words will be generated by which English words. Then the decoder will then generate the French translation, one word at a time, focusing its attention on the words determined by the AM, producing the French sentence.

This model performs better than the original encoder/decoder architecture. Through using the bidirectional recurrent neural network, the incorporation of the various components of language is perfectly observed. We don’t just do word to word translation but also consider tokens and grammar (syntax and semantic analysis).