Machine Translation

SOUMMER EL Mehdi, Data Engineering Student at National Institute Of Statistics And Applied Economics

ABSTRACT

Machine translation is a branch of computational linguistics that studies the process of translating text or speech from one language to another. there is an increased need and demand for language translations owing to the fact that language is an effective medium of communication. This paper is brief introduction to Machine Translation and its modern approach.

Keywords: Machine Translation, Sequence To Sequence, Encoder, Decoder, Attention mechanism

1. INTRODUCTION

Translation is part of the digital world. It connects people and businesses globally. In today's world, people are increasingly connected and are able to communicate with each other no matter their language. Translation is helping people connect and communicate globally.

The concept of translation was based on word-to-word translation. On a primary level, Machine translation performs a simple change of words in one natural language for words in another, but that usually cannot produce a good translation of a text, because recognition of whole phrases and the context in the target language is needed.

2. TRANSLATION PROCESS

The translation process can be described as a Decoding the meaning of the source text and re-encoding this meaning in the target language. Behind this simple procedure lies a complex cognitive process. To decode the meaning of the source text in its entirety, the translator must interpret and analyze all the features of the text, a process that requires in-depth knowledge of the grammar, syntax, semantics, etc.., of the source language. The translator needs the same in-depth knowledge to re-encode the meaning in the target language.

3. MACHINE TRANSLATION APPROACHES

Machine translation can use rules based on the language's grammar. It translates words by replacing the ones from the source language with the most suitable ones.

3.1 Rule-based Approach

The Rule-based Machine Translation works on the morphology, syntax and semantic of both languages. So, we required the syntax analysis, semantic analysis of Source text and to generate the text in target language we need syntax generation and semantic generation. We also need the bilingual dictionary of source and target languages. There are three different types of rule-based machine translation systems:

Direct Systems (Dictionary Based Machine Translation) map input to output with basic rules. Transfer RBMT Systems (Transfer Based Machine Translation) employ morphological and syntactical analysis. Interlingual RBMT Systems (Interlingua) use an abstract meaning.

To get a French translation of this English sentence one needs a dictionary that will map each English word to an appropriate French word, rules representing regular English sentence structure, representing regular French sentence structure.

3.1.1 Interlingual Approach

Interlingual machine translation is one instance of rule-based machine-translation approaches. The source language is transformed into an interlingual language independent of any language, then the target language is generated out of the interlingua. One of the major advantages of this system is that the interlingua becomes more valuable as the number of target languages it can be turned into increases.

3.1.2 Transfer base Approach

Transfer-based machine translation creates a translation from an intermediate representation that simulates the meaning of the original sentence, it is then used to generate the target language text with help bilingual dictionary and grammar rules.

3.2 Statistical Approach

In statistical machine translation, translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. The statistical approach contrasts with the rule-based approaches to machine translation as well as with example-based machine translation. A document is translated according to the probability distribution $P(e \mid f)$ that a string e in the target language is the translation of a string f in the source language. statistical machine translation has a number of approaches for modeling the probability distribution $P(e \mid f)$. One approach is to apply Bayes Theorem $P(e \mid f) = P(f \mid e).P(e)$ where the translation model and $P(e \mid f)$ is the probability that the source string is the translation of the target string, and $P(e)$ is the probability of seeing that target language string. Finding the best translation is done by picking up the highest probability:

$$\hat{e} = arg\,max_e P(f \mid e) = arg\,max_e P(f \mid e).P(e) \tag{1}$$

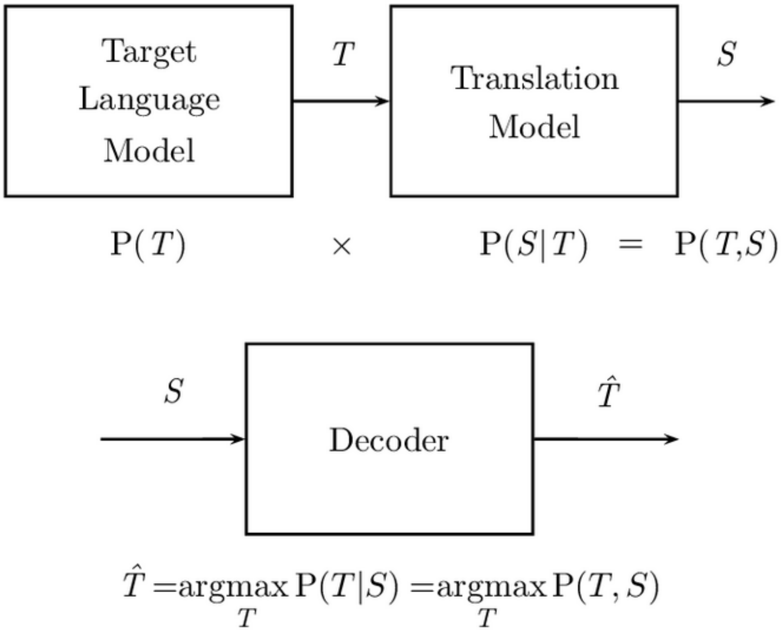
Target Language Model
T
Translation Model
S
$\mathrm{P}(T) \times \mathrm{P}(S|T) = \mathrm{P}(T,S)$
S
Decoder
$\hat{T}$
$\hat{T} = \underset{T}{\mathrm{argmax}}\, \mathrm{P}(T|S) = \underset{T}{\mathrm{argmax}}\, \mathrm{P}(T,S)$

Figure 1. Statistical Machine Translation.

3.3 Example-based Approach

Example-based machine translation is a memory-based translation based on the idea of analogy. it contains the point to point mapping between the source language and target language sentences basic idea is if already translated sentence occur again it, the same translation is likely to be correct again.

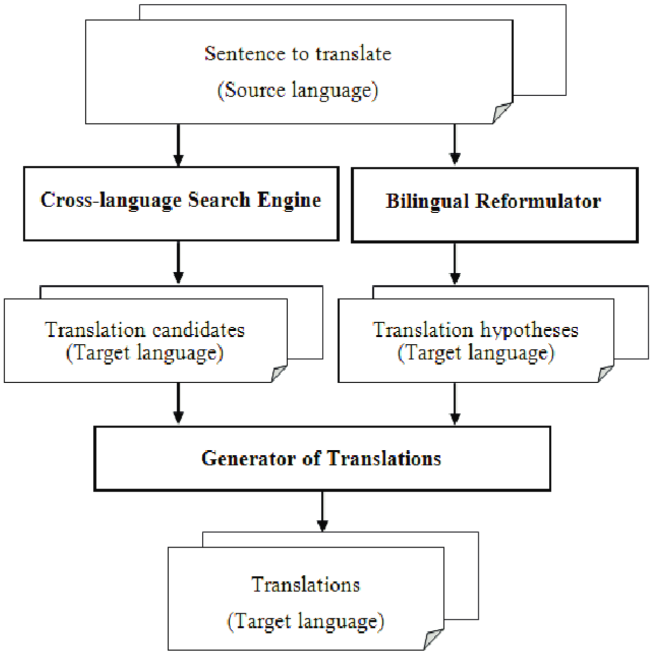

Figure 2. Architecture of the Example-based Machine Translation.

3.4 Neural Machine Translation

Neural machine translation is a Deep-learning approach that uses neural networks to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model. Neural machine translation has achieved great success and has become the new mainstream method in practical Machine Translation systems. As a data-driven approach to machine translation NMT try to estimate an unknown conditional distribution $P(y \mid x)$ given the dataset D, where x and y are random variables representing source input and target output, respectively. We attempt to answer the basic questions of how to design neural networks to model the conditional distribution? and how to generate a translation sentence from the NMT model with a given a source input? and finally how to effectively learn the parameters of NMT from data?

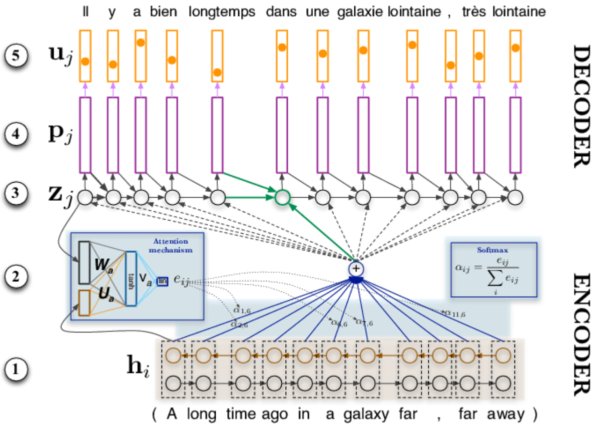
Il y a bien longtemps dans une galaxie lointaine , très lointaine
5
$\mathbf{u}_j$
4
$\mathbf{p}_j$
3
$\mathbf{z}_j$
2
1
$\mathbf{h}_i$
DECODER
ENCODER
Attention mechanism
W_a
U_a
tanh
v_a
e_{ij}
Softmax
$\alpha_{ij} = \frac{e_{ij}}{\sum_i e_{ij}}$
(A long time ago in a galaxy far , far away)

Figure 3. Architecture of the Neural Machine Translation system equipped with an attention mechanism.

3.4.1 Models

Auto-regressive model : Translation can be modeled at different levels, such as document, paragraph, and sentence-level. In this article, we focus on sentence-level translation. Besides, we also assume the input and output sentences are sequences. Thus the NMT model can be viewed as a sequence-to-sequence model. Assuming we are given a source sentence $x = \{x_1, ..., x_S\}$ and a target sentence $y = \{y_1, ..., y_T\}$ we can predict the output with the equation of the Left to right auto-regressive NMT :

$$P(Y = y \mid X = x) = \prod_{t=1}^{T} P(y_t \mid y_0, ..., y_{t-1}, x_0, ..., x_{t-1}) \tag{2}$$

RNN Encoder-Decoder model : Almost all neural machine translation models employ the encoder-decoder

Encoder : The task of the encoder is to maps the the source embeddings into hidden continuous representations "c". Then we process these words with a Recurrent Neural Network, the computation can be described as:

$$h_t = f(x_t, h_{t-1}) \tag{3}$$

$$c = q(h_1, , h_{T_x}) \tag{4}$$

where $\mathrm{h}_t \in \mathrm{R}^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. f and q are some nonlinear functions.

Decoder : The decoder is trained to predict the next word, given the context vector and all the previously predicted words. Mathematically the decoder defines a probability over the translation by decomposing the joint probability into the ordered conditionals

$$P(y) = \prod_{t=1}^{T} P(y_t \mid y_0, ..., y_{t-1}, x_0, ..., x_{t-1}) = \prod_{t=1}^{T} g(y_{t-1}, s_t, c) \tag{5}$$

where $\mathrm{h}_t \in \mathrm{R}^n$ is a hidden state at time t, and c is a vector generated from the sequence of the hidden states. g is a nonlinear function, potentially multi-layered, function that outputs the probability of the next word y_t and s_t is the hidden state of the RNN.

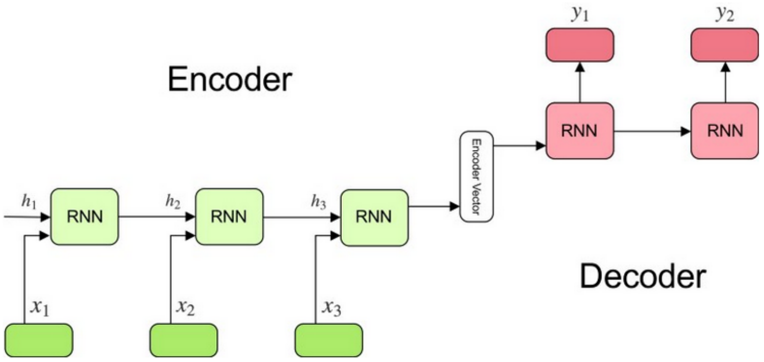

Figure 4. Encoder-Decoder Sequence to Sequence Model.

Encoder-Decoder model with attention : As a natural extension of the Encoder-Decoder model, Attention was proposed as a solution to the limitation of the Encoder-Decoder model encoding the input sequence to one fixed length vector from which to decode each output time step. This issue is believed to be more of a problem when decoding long sequences.

Encoder : as the previous model the encoder maps the the source embeddings into hidden continuous representations. Then we process these words with a Recurrent Neural Network, but to get the right context, a recurrent neural network that runs on the reverse, from the end of the sentence to the beginning, was built. Having two recurrent neural networks running on opposite directions is called a bidirectional recurrent neural network which means that the encoder consists of the embedding lookup for each input word, and the mapping that steps through the hidden states the computation can be described as (the +,- depend on the direction):

$$h_j = f(h_{j\pm1}, E_{x_j}). \tag{6}$$

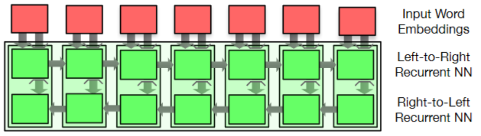

Figure 5. Input Encoder (bidirectional recurrent neural network)

Decoder : for the decoder unlike the RNN encoder–decoder approach the probability here is conditioned on a distinct context c for each target word y :

$$P(y_t \mid y_0, ..., y_{i-1}, x) = g(y_{i-1}, s_i, c) \tag{7}$$

where s_i is an RNN hidden state for time i :

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{8}$$

and the context c is computer as a weighted sum of h_i : $c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$ with

$$\alpha_{ij} = softmax(e)_{ij} = \frac{exp(e_{ij})}{\sum_j exp(e_{ij}))} \tag{9}$$

where $e_{ij} = a(s_{i-1}, h_j)$ is an alignment model which scores how well the inputs around position j and the output at position i match.

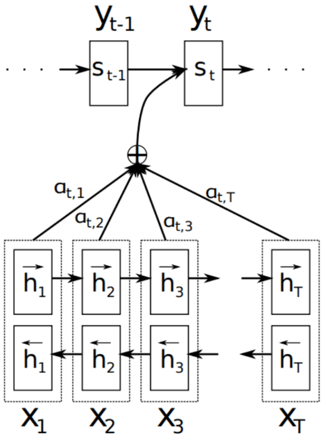

Figure 6. Output Decoder

Let α_{ij} be a probability that the target word y_i is aligned to, or translated from, a source word x_j . Then, the i-th context vector c is the expected annotation over all the annotations with probabilities α_{ij} . Intuitively, the decoder decides parts of the source sentence to pay attention to, realizing an attention mechanism.

4. ALTERNATE ARCHITECTURES

Most neural network research has focused on the use of recurrent neural networks with attention. But this is by no means the only architecture for neural networks. the Convolutional Neural Networks (resp. CNNs with attention) are also used, but nowadays State-of-the-art NMT models are primarily RNNs with attention mechanism and the Transformer which implement the self-attention mechanism

4.1 the Transformer Architecture

The Transformer use the classic encoder-decoder structure for his architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder.

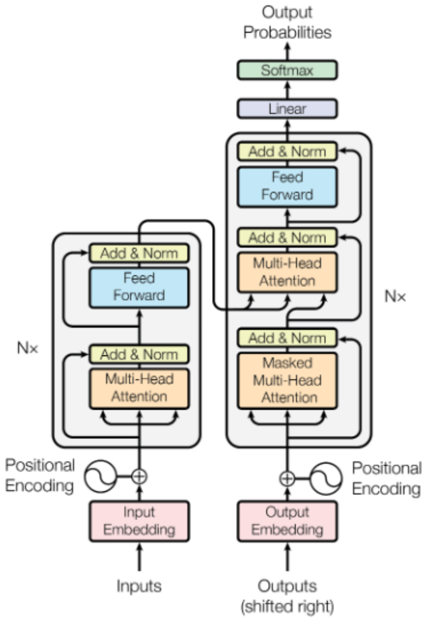

Figure 7. The Transformer model architecture.

Scaled dot-product attention: The transformer blocks are scaled dot-product attention units. When a sentence is passed into a transformer model, attention weights are calculated at the same time between every token. The attention unit produces embeddings for every token in context that includes information about the token itself along with a weighted combination of other related tokens each weighted by its attention weight.

The attention calculation for all tokens can be expressed as the softmax of The matrices Q , K and V are defined as the matrices where the ith rows are vectors $q_i = x_i W_Q$, $k_i = x_i W_K$ and $v_i = x_i W_V$ resp. where $W_{[Q,K,V]}$ are resp. the query, key and value weights matrices that the the transformer model learns.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{10}$$

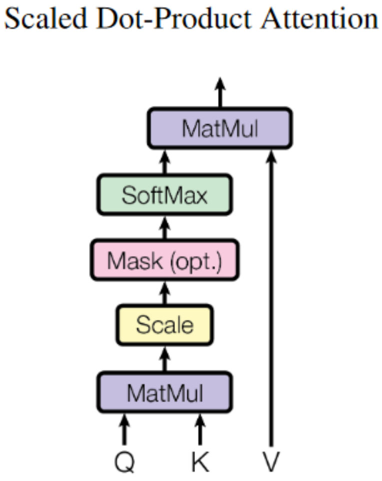

Figure 8. Scaled Dot-Product Attention.

Multi-Head Attention: the word came from the fact that on each layer in a transformer there is a set of the query, key and value weights matrices $W_{[Q,K,V]}$ which are called head. e computations for each attention head can be performed in parallel

$$MultiHead(Q,K,V) = Concat(head_1, ..., head_h)W^O \tag{11}$$

where $W^O \in R^{h.dim_{values} \times dim_{model}}$ and h is the number of parallel attention layers.
The outputs for the attention layer are concatenated to pass into the feed-forward neural network layers.

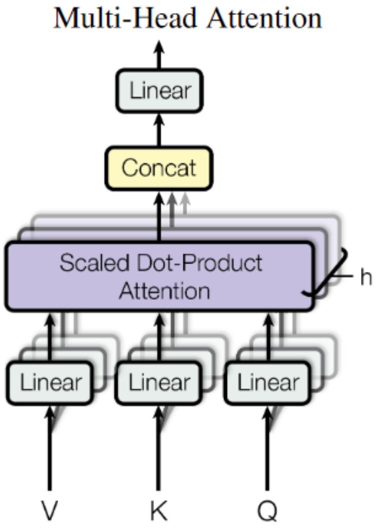

Figure 9. Scaled Dot-Product Attention.

Encoder: the encoder is composed of N = 6 identical layers,each layer have two sub-layers:
1- the multi-head attention mechanism accepts input encodings from the previous encoder and weighs their relevance to each other to generate output encodings.
2- the Feed Forward neural network further processes each output encoding individually.
The first encoder takes positional information and embeddings of the input sequence as its input, rather than encodings because no other part of the transformer makes use of this

Decoder: the encoder is composed of N = 6 identical layers,each layer have three sub-layers:
1- self-attention mechanism.
2- attention mechanism over the encodings.
3- feed-forward neural network.
The first decoder takes positional information and embeddings of the output sequence as its input, rather than encodings, the output sequence must be partially masked to prevent the transformer to use the current or future output to predict an output.
The last decoder is followed by a final linear transformation and softmax layer to produce the output probabilities.

5. CONCLUSION

Since the turn of the century, machine translation has experienced considerable growth, with several systems capable of automatically translating increasingly long texts in a matter of seconds. Years pass and new technologies take place to increase the quality of translation, starting from the normal approach to transformers architecture, and the accuracy augments with the complexity.

REFERENCES

Neural Machine Translation: A Review of Methods, Resources, and Tools, https://arxiv.org/pdf/2012.15515v1.pdf
Statistical Machine Translation, https://arxiv.org/pdf/1709.07809.pdf
Neural Machine Translation by JOINTLY LEARNING TO ALIGN AND TRANSLATE, https://arxiv.org/pdf/1409.0473.pdf
Attention Is All You Need, https://arxiv.org/pdf/1706.03762.pdf
https://towardsdatascience.com/neural-machine-translation-inner-workings-seq2seq-and-transformers-229faff5895b