Language is at the core of human communication, and for centuries, we've endeavored to unravel how it is intricately woven into
our cognitive processes. Today, we are on a quest to advance further by introducing computers to the realm of human language
through Natural Language Processing (NLP). NLP is an artificial intelligence branch dedicated to harnessing machine learning
techniques for a wide range of language-related tasks.

In recent years, NLP has witnessed a remarkable evolution with the emergence of transformative techniques such as transformers, the encoder-decoder framework, and the attention mechanism. These innovations have reshaped the landscape of NLP. To gain a deep understanding of these developments, one of the most valuable resources is the book Natural Language Processing with Transformers, Building Language Applications with Hugging Face,which we are summarizing in this series of articles.

In this first article, we will introduce the core concepts that underlie the pervasiveness of transformers and put them into context.

## The encoder-decoder framework:

The concept of an encoder-decoder, also known as a sequence-to-sequence architecture, made its debut in 2014 with the publication of the "Sequence to Sequence Learning with Neural Networks" paper by a Google research team. This architecture finds its forte in scenarios where both the input and output entail sequences of arbitrary lengths.
Comprising two core components, The encoder processes the input sentence word-by-word using recurrent models (RNN/LSTM/GRU), produces a representation of the entire sentence in a hidden space, and then the decoder retrieves this hidden state (representation) and generates an output.

However, a notable weakness of this architecture lies in the final hidden state of the encoder, which imposes an information bottleneck. This state must encapsulate the meaning of the entire input sequence since it serves as the sole source of information for the decoder during output generation. This limitation becomes particularly pronounced when dealing with longer input sequences.

Check out the second slide, it provides an overview of the inner workings of the architecture!!!
1- Word embedding: each word is represented by a dense, low-dimensional vector.
2- The Encoder processes the input sentence sequentially using the RNN cells and produces the final hidden state == sentence representation.
3- At each step, the Decoder, using the hidden state and the embedding of the previous token, generates the next most probable token.

## The attention mechanism: