

Translation English to French

Mohamad Lakkis

November 8, 2024

Abstract

This document explains the model used to translate English text to French text, presenting the model's results and discussing its limitations. This study provides insights for understanding sequence-based NLP tasks, particularly in the context of translation using recurrent neural networks.

1 Introduction

This translation model was trained in two different settings: one using unidirectional and the other using bidirectional LSTM networks to convert English text to French. Although this basic LSTM architecture produces reasonably not bad results, we explore potential improvements using an attention mechanism, which we leave for future work. For now, we focus on LSTM networks with multiple layers, as will be discussed later.

The goal of these simulations is to enable the model to grasp the semantic meaning of phrases, rather than performing a simple word-by-word translation. To accomplish this, we selected an encoder-decoder architecture: the encoder captures the semantic structure and order of the words, while the decoder generates the French translation. In future work, we plan to enhance this model by introducing an attention mechanism and expanding the dataset.

2 Methodology

In this section, we describe the model's architecture and the training process. The model consists of an encoder and a decoder, both of which are LSTM networks. The encoder processes the input sequence, while the decoder generates the output sequence. The model is trained using the Adam optimizer and the categorical cross-entropy loss function.

2.1 Data Preprocessing

As a first step, we need an english and french vocabulary, so we tokenize the input sentences (in this case we used a tokenizer for each word, from NLTK library) and then formed the english vocabulary, from these distinct tokens. After that we start indexing each token in the input sequences. At this step each token is represented by an integer, so consequently each sentence is represented by a sequence of integers.

The same process is done for getting the french vocabulary and indexing the output sequences.

Additionally, we will also get the inverse mapping of the french and english vocabs, to be able to convert the output sequences from the model to the corresponding words.

Now in each of the two vocabs (english and french) we will add to them 4 special tokens:

- `<PAD>` token: to pad the sequences to have the same length.
- `<UNK>` token: to represent the unknown words.
- `<EOS>` token: to represent the end of the sentence.
- `<SOS>` token: to represent the start of the sentence.

Now we for the output sequences we will add the `<SOS>` token at the start of each sequence and the `<EOS>` token at the end of each sequence.

Note: we will not add these tokens to the input sequence.

Furthermore, we will pad the input sequences to have the same length (as maximum length of the input sequences) so any input sequence that is shorter than the maximum length will be padded with the `<PAD>` token. And any input sequence that is longer than the maximum length will be truncated.

Similar steps will be done for the output sequences.

Not that for the input sequence we will not be adding `<EOS>` and `<SOS>` tokens. We will use another method to process only the unpadded part of the sequence, and for that we will need the `src_length`, which will denote the actual length of the each sequence before padding.

After this step our inputs and expected outputs are ready to be fed to the model.

So to wrap up: Our inputs for the models will be the padded input sequences and the `src_length`. As for its output (during training) will be probability distribution over the french vocab, as for the output during inference it is just the class (the french token, that is most probable).

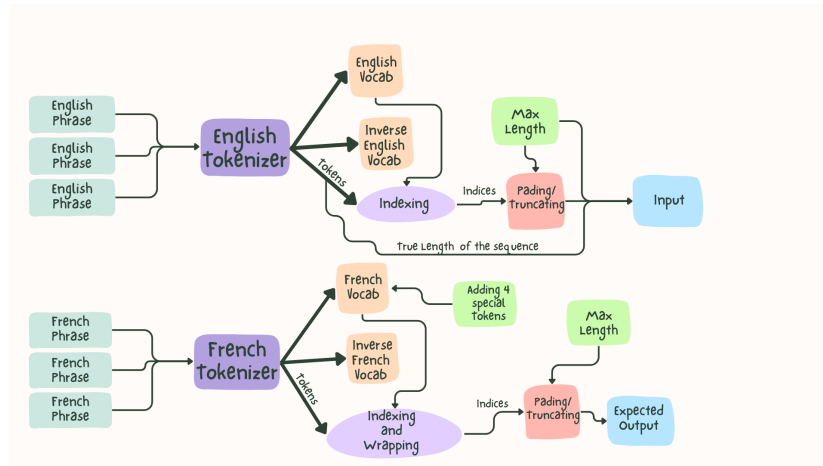


Figure 1: Data Preprocessing

Important Note: Both Inputs and Outputs are common for both models.

2.2 Model 1 Architecture: One-Directional LSTMs

Now that we understand the data preprocessing, we can move to the model architecture.

I think it is better now to look at the general hierarchy of the model through the following figure.

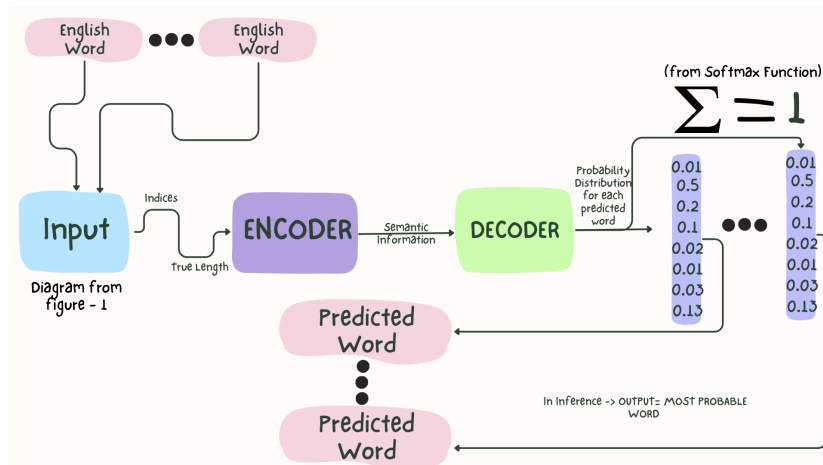
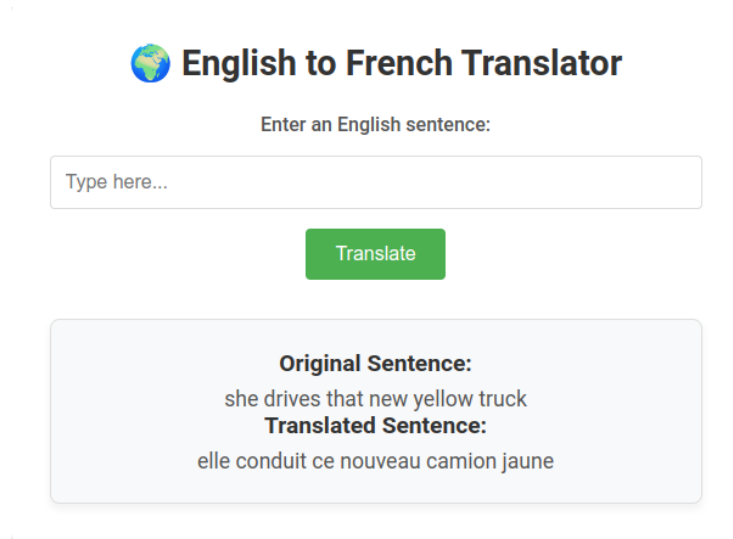


Figure 2: General Overview of the Model

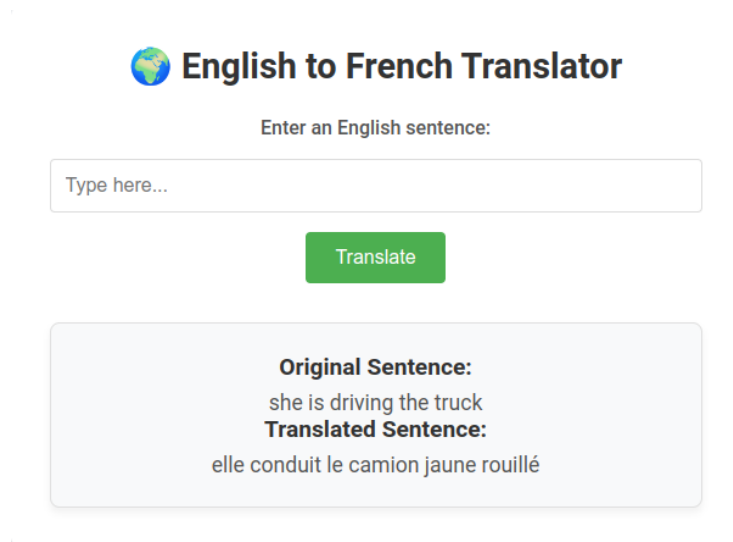
3 Importance of training and ways to prevent overfitting



The screenshot shows a web interface for an "English to French Translator". At the top, there is a globe icon followed by the title "English to French Translator". Below the title, it says "Enter an English sentence:". There is a text input field with the placeholder "Type here...". Below the input field is a green button labeled "Translate". Below the button, there is a light gray box containing the following text:

Original Sentence:
she drives that new yellow truck
Translated Sentence:
elle conduit ce nouveau camion jaune

Figure 3: Correct Output



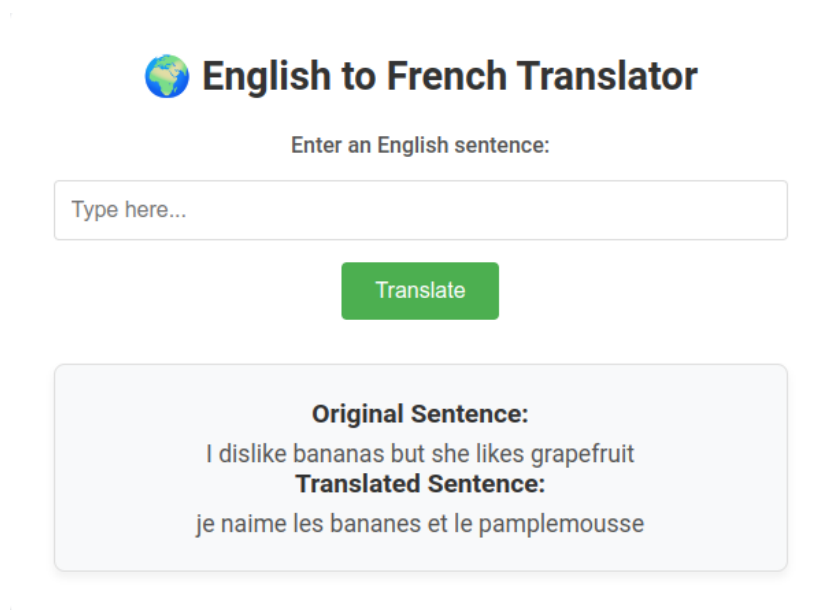
The screenshot shows the same web interface as Figure 3. The input field is empty. The "Translate" button is green. Below the button, there is a light gray box containing the following text:

Original Sentence:
she is driving the truck
Translated Sentence:
elle conduit le camion jaune rouillé

Figure 4: Overfitting and Association

From these two figures we can see how the model learned to associate the word truck always with color jaune, so even when the input is just truck the model will still output the "yellow truck" in french. This could be solved by adding more data, or by using additional layers with dropout, or also by using the attention mechanism which we will explore next.

4 Importance of Attention Mechanism



The screenshot shows a web interface for an "English to French Translator". It features a text input field with the placeholder "Type here...", a green "Translate" button, and a light blue output box. The output box displays the "Original Sentence: I dislike bananas but she likes grapefruit" and the "Translated Sentence: je naime les bananes et le pamplemousse". The word "dislike" is highlighted in blue in the original sentence, and "naime" is highlighted in blue in the translated sentence, illustrating a mistranslation where the model failed to capture the negative sentiment.

Figure 5: Attention Mechanism

this example clearly illustrates the importance of incorporating an attention mechanism in a Seq2Seq model. Without attention, the model processes the sentence sequentially, which can lead it to focus only on certain parts, especially the beginning or end, while losing context in the middle or when shifts in meaning occur (e.g., handling both "I dislike bananas" and "she likes grapefruit").

In this example, the model incorrectly translates "dislike" to "like" (in French, "je n'aime pas" would be used for "dislike" instead of "j'aime") and ignores the contrast between "I" and "she," leading to an incorrect translation.

The attention mechanism allows the model to dynamically focus on relevant parts of the input sequence as it generates each word in the output sequence. This enables the model to capture the context and meaning of the entire sentence, leading to more accurate translations.

The topic of attention will be explored in future work, as it is a crucial component

for improving the model's performance.

5 Conclusion

Your conclusion here.