A MACHINE TRANSLATION MODEL FOR FRENCH TO ENGLISH WITH BLEU SCORE

Dhruv Jain, Sunday Okechukwu

ABSTRACT

Machine translation is an essential technology for breaking language barriers and fostering cross-cultural communication. Loosely speaking Machine Translation refers to the automated process of translating text or speech from one language called (source language) to another language called (target language). For eg. frn-eng or spn-eng. Technically, it involves training machine learning models on large datasets of parallel text, where the same text is available in two or more languages In this paper, we present a machine translation model that translates French sentences into English using an encoder - decoder architecture with BLEU metric for evaluations. The encoder and decoder is built by stacking LSTMs. We use this type of layer because its structure allows the model to understand context and temporal dependencies of the sequences.

The model is trained on a large dataset of pre-processed French and English sentence pairs. The LSTM neural network model is used to learn the underlying patterns in the dataset and generate English translations for new French sentences. The model is trained to optimize the BLEU score, a widely-used metric for evaluating the quality of machine translation systems.

***Index Terms*—** BLEU, LSTM, French language, Englisgh language

1. INTRODUCTION

Language is a system of communication used by a particular country or community. considering language as a barriers from one country to another country. English is the most popular language world wide and its user are billions. French is the fifth most popular language across the world and its users are around 275+ millions. Looking into this we are finding a solution that can help a lot of foreign treaty to solve this problem using the model provided by Dhruv and Sunday.

The basic idea is to convert the french words to English using Natural language process and implementing the different strategies to get the best performance evaluation for this project. French to English translation using BLEU model evaluation is an automated translation system that uses a Long Short-Term Memory (LSTM) neural network model to translate French sentences into English. The system calculates the Bilingual Evaluation Understudy (BLEU) score, a widely-used metric for evaluating the quality of machine translation, to measure the accuracy of the translations. The model is trained to predict the English translation of a given French sentence by learning the underlying patterns in the data set. Once trained, the model can be used to translate new French sentences into English. This system has potential applications in various domains, including language learning, cross-cultural communication, and international business.

Main stages of the Project

- Data Acquisition
- Pre-processing
- Feature Extraction
- Modelling
- Evaluations
- Summary and direction for future work

1.1. All about data set

The data Acquisition collected from https://www.statmt.org/europarl/.
This is a huge data set contains 15215 unique words in english and 29251 unique words in french. The data after loading looks like this:

```
for sample in range(7):
    print(f'small_vocab_en Line {sample + 1}:  {english_sentences[sample]}')
    print(f'small_vocab_fr Line {sample + 1}:  {french_sentences[sample]}')
```

```
small_vocab_en Line 1:  new jersey is sometimes quiet during autumn , and it is snowy in april .
small_vocab_fr Line 1:  new jersey est parfois calme pendant l' automne , et il est neigeux en avril .
small_vocab_en Line 2:  the united states is usually chilly during july , and it is usually freezing in november .
small_vocab_fr Line 2:  les états-unis est généralement froid en juillet , et il gèle habituellement en novembre .
small_vocab_en Line 3:  california is usually quiet during march , and it is usually hot in june .
small_vocab_fr Line 3:  california est généralement calme en mars , et il est généralement chaud en juin .
small_vocab_en Line 4:  the united states is sometimes mild during june , and it is cold in september .
small_vocab_fr Line 4:  les états-unis est parfois légère en juin , et il fait froid en septembre .
small_vocab_en Line 5:  your least liked fruit is the grape , but my least liked is the apple .
small_vocab_fr Line 5:  votre moins aimé fruit est le raisin , mais mon moins aimé est la pomme .
small_vocab_en Line 6:  his favorite fruit is the orange , but my favorite is the grape .
small_vocab_fr Line 6:  son fruit préféré est l'orange , mais mon préféré est le raisin .
```

Fig. 1. French and English sentence by sentence

1.2. Data Acquisition

We started with a small dataset and use some tricks to create more data. Randomly select words in a sentence and trying the model accuracy. Take a uni gram, bi gram, tri gram, four gram for evelauting model.Back translation was used but the model is giving us a decent accuracy using BLEU model. [7]

```
50263459 English words.
301951 unique English words.
10 Most common words in the English dataset:
"the" "of" "to" "and" "in" "that" "a" "is" "for" "I"

52562231 French words.
395651 unique French words.
10 Most common words in the French dataset:
"de" "la" "et" "le" "à" "les" "des" "que" "en" "du"
```

Fig. 2. French and English Word Frequencies: Most Common Words

1.3. Text cleaning

The data collected does not required any data cleaning. The data doesn't contain any NA/missing values plus the french translation has more words in the model.

1.4. Pre-processing

We followed some some basic text pre processing: Lowering the text, puncuation, Padding. From sentences to word and assigning the words to a number for each sentence. From the below a mtrix is further created and the max number of lenght in sentences was 200 words.

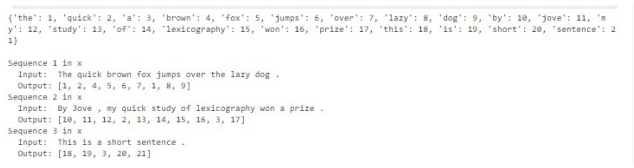

Fig. 3. Assigning each words a unique number

Since a majority of the pipeline is built with language specific tools, what will happen to our NLP pipeline, which is expecting English text? In such cases, language detection is performed as the first step in an NLP pipeline.Many people across the world speak more than one language in their day-to-day lives. Thus, it's not uncommon to see them using multiple languages in their social media posts, and a single post may contain many languages.

Input sequence length in machine translation, the length of the input sequence can vary from one sentence to another. However, neural networks are designed to work with fixed-length inputs. Padding is used to make all input sequences of the same length, which allows them to be processed in batches, making training more efficient.

Output sequence length similar to the input sequence length, the length of the output sequence can also vary. In order to generate an output sequence of the desired length, padding is used to ensure that all output sequences are of the same length.

Overall, padding is necessary in machine translation to ensure consistency in input representation, facilitate efficient computation, and enable the use of batch processing during training.

```
Sequence 1 in x
  Input:  [1 2 4 5 6 7 1 8 9]
  Output: [1 2 4 5 6 7 1 8 9 0]
Sequence 2 in x
  Input:  [10 11 12  2 13 14 15 16  3 17]
  Output: [10 11 12  2 13 14 15 16  3 17]
Sequence 3 in x
  Input:  [18 19  3 20 21]
  Output: [18 19  3 20 21  0  0  0  0  0]
```

Fig. 4. Padding

1.5. Feature engineering

Tokenization is the process of splitting a sentence or a piece of text into individual words, which are called tokens. In the context of language translation from French to English, word-level tokenization is an important pre-processing step that is used to convert the input text into a format that can be processed by a machine learning model. In tokenization, each word in the input text is treated as a separate token.Each of these words is considered a separate token, which can then be used as input to a machine learning model for language translation.

Here, instead of using pre-trained word embeddings such as GloVe. We shall train our own embeddings using Keras embedding layer. Word embeddings can be thought of as an alternate to one-hot encoding along with dimensionality reduction.

Embedding layer enables us to convert each word into a fixed length vector of defined size.

There are three parameters to the embedding layer.

•Input dim : Size of the French vocabulary

•Output dim : shape of the embedding vector

•Input length : Maximum length of the input sequence

1.6. Modeling

Data attribute Data consist of full length document.

Decision path Chose the right strategy for breaking the document into lower levels, Like pararaph, sentences or phases.

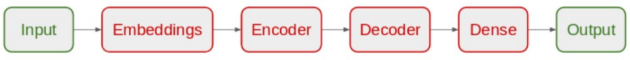

Fig. 5. Modle Architecture

The **encoder** takes the input sequence (in French) and encodes it into a fixed-length vector representation. This vector representation captures the meaning of the input sequence, which is then used as input to the decoder.

The **decoder** then generates the output sequence (in English) based on the encoded input representation. The de-

coder receives the encoded representation from the encoder and generates the output sequence one word at a time, starting with a special start-of-sentence token and ending with an end-of-sentence token.[3]

An encoder decoder structure allows for a different input and output sequence length. First, we use an Embedding layer to create a spatial representation of the word and feed it into a LSTM layer that outputs a hidden vector, because we just focus on the output of the last time step we use return sequences is equal to False.

This output vector needs to be repeated the same number of times as the number of time step in the decoder part, for that we use the Repeat Vector layer. The decoder will be built with LSTM layer and the parameter return sequences is equal to True, so each output of the time steps is used by the Dense layer.

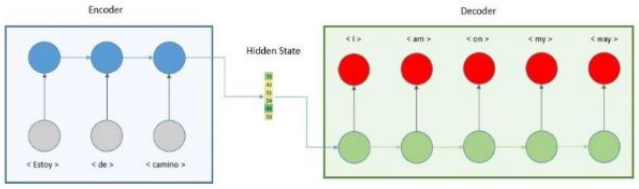

Fig. 6. Encoder and Decoder logic using LSTM

Embedding layer takes the input sequence and maps each word to a dense vector representation of units dimensions. This layer helps the model to better capture the meaning of words in the input sequence.

Bidirectional LSTM layer processes the embedded input sequence in both forward and backward directions and outputs a sequence of hidden states for each time step. [2]

Dropout layer randomly drops out some of the neurons to prevent overfitting in the model. Dense layer applies a **softmax activation** function to the output of the previous layer to generate the probability distribution over the output vocabulary.

1.7. Evaluation

Intrinsic evaluation focuses on intermediary objectives.
BLEU (Bilingual Evaluation Understudy) is a metric used for evaluating the quality of machine translation output against one or more reference translations. It calculates a score between 0 and 1, with higher scores indicating better translations. BLEU score is widely used in natural language processing (NLP) and machine translation evaluation. [5] It is used to measure the quality of text translated from a model The idea behind Bleu is to assign a single numerical score to a translation that tells us how good it is when compared to one or more human reference translations The approach that Bleu takes is to comapare the ngrams of the generated translations to the ngrams of the refrences Bleu is based on

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 111, 512)          15113728

 bidirectional (Bidirectiona  (None, 111, 1024)        4198400
 l)

 dropout (Dropout)           (None, 111, 1024)         0

 bidirectional_1 (Bidirectio  (None, 1024)             6295552
 nal)

 repeat_vector (RepeatVector  (None, 100, 1024)        0
 )

 bidirectional_2 (Bidirectio  (None, 100, 1024)        6295552
 nal)

 dropout_1 (Dropout)         (None, 100, 1024)         0

 dense (Dense)               (None, 100, 15251)        15632275

=================================================================
Total params: 47,535,507
Trainable params: 47,535,507
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7. Hyper parameter tuning model

ngram precision to measure translation quality.

One can observe that the train data set When we use it the Actual and predicted both are similar.

```
for i in range(10):
  print('Actual French Sentence: ', frn_train1[i])
  print('Actual English Sentence: ', eng_train1[i])
  print("Predicted English Sentence: ", logits_to_text(model.predict(frn_train[:10])[i], eng_tokenize
  print('\n')
```

```
Actual French Sentence:  dans ma chambre il y a une grande armoire
Actual English Sentence:  my room has a large closet
1/1 [==============================] - 5s 5s/step
Predicted English Sentence:  my large a a large large <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>


Actual French Sentence:  paris est parfois chaud au mois de novembre  et il est beau en septembre
Actual English Sentence:  paris is sometimes hot during november  and it is beautiful in september
1/1 [==============================] - 0s 44ms/step
Predicted English Sentence:  paris is sometimes hot during november and it is beautiful in september


Actual French Sentence:  tom est comme a
Actual English Sentence:  tom is like that
1/1 [==============================] - 0s 44ms/step
Predicted English Sentence:  tom is like that <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <
```

Fig. 8. Train score predictors

One can observe that the Test data set shows difference in actual and predicted values. For exaple actual english text is ”you are luck u didn’t die” but in predicted we observe ”you are luck u didn’t be.”

The output shows the BLEU scores for n-gram matches between the reference translation and the machine translation. The n-gram matches are computed for 1-grams (individual words), 1-2 grams (contiguous sequences of up to 2 words), 1-3 grams (contiguous sequences of up to 3 words), and 1-4 grams (contiguous sequences of up to 4 words).

In the output you provided, the BLEU scores are as follows: **1-grams: 0.8787 2-grams: 0.8546 3-grams: 0.8450 4-grams: 0.8179**

These scores indicate how closely the machine translation matches the reference translation in terms of the specified n-

```
Actual French Sentence:  vous avez de la chance de ne pas tre mortes
Actual English Sentence:  youre lucky you didnt die
1/1 [==============================] - 0s 42ms/step
Predicted English Sentence:  youre lucky you didnt be <PAD> <PAD> <PAD> <PAD> <PAI


Actual French Sentence:  je me suis cach derrire un rideau
Actual English Sentence:  i hid myself behind a curtain
1/1 [==============================] - 0s 44ms/step
Predicted English Sentence:  i hid myself in a <PAD> <PAD> <PAD> <PAD> <PAD> <PAD:


Actual French Sentence:  je veux prsenter mes excuses
Actual English Sentence:  i want to apologize
1/1 [==============================] - 0s 43ms/step
Predicted English Sentence:  i want to apologize <PAD> <PAD> <PAD> <PAD> <PAD> <PA


Actual French Sentence:  je peux avoir un clin
Actual English Sentence:  can i have a hug
1/1 [==============================] - 0s 43ms/step
Predicted English Sentence:  i can give a hug <PAD> <PAD> <PAD> <PAD> <PAD> <PAD>
```

Fig. 9. Test score predictors

grams. A higher score means that the machine translation contains more n-gram matches with the reference translation. [4] [6] The geometric mean for n-gram in machine translation evaluation is used to combine the individual precision scores for each n-gram (1-gram, 2-gram, 3-gram, etc.) into a single score that represents the overall translation quality.

The geometric mean is computed as the nth root of the product of the individual precision scores, where n is the number of n-gram types being evaluated. For example, if we are evaluating 1-gram, 2-gram, 3-gram precision and 4-gram precision, then the geometric mean would be computed as the cube root of the product of the three individual precision scores. The formula to add is blue4 = (p1*p2*p3*p4)(1/4). The GM is 0.8487 for overall n-grams used in the model. [1]

Regenerate response:

```
{'1-grams': 0.8787440241129639,
 '1-2-grams': 0.8546182128974017,
 '1-3-grams': 0.8449712336285509,
 '1-4-grams': 0.8178677566126499}
```

Fig. 10. BLEU score

1.8. Summary and Future work

Even though this model gives a decent result. We could improve it by increasing the number of LSTM layers in the model, instead of just three layers in the encoder and three layers in the decoder. We could also use a pre-trained embedding layer like word2vec or Glove. Finally, we could use the attention mechanism which is one of the major improvements in the natural language processing field.

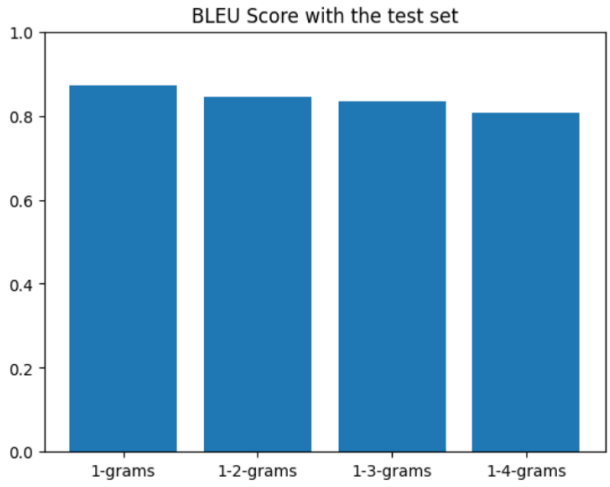

Fig. 11. BLEU score

2. REFERENCES

[1] KM Tahsin Hassan Rahit, Rashidul Hasan Nabil, and Md Hasibul Huq. Machine translation from natural language to code using long-short term memory. In *Proceedings of the Future Technologies Conference (FTC) 2019: Volume 1*, pages 56–63. Springer, 2020.

[2] Sarita G Rathod. Machine translation of natural language using different approaches. *International journal of computer applications*, 102(15), 2014.

[3] Middi Venkata Sai Rishita, Middi Appala Raju, and Tanvir Ahmed Harris. Machine translation using natural language processing. In *MATEC Web of Conferences*, volume 277, page 02004. EDP Sciences, 2019.

[4] Holger Schwenk, Marta R Costa-Jussa, and José AR Fonollosa. Smooth bilingual n-gram translation-gram translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 430–438, 2007.

[5] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[6] Lirong Yao and Yazhuo Guan. An improved lstm structure for natural language processing. In *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pages 565–569. IEEE, 2018.

[7] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. Bilingual word embeddings for phrase-

based machine translation. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1393–1398, 2013.