e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

LANGUAGE TRANSLATION USING MACHINE LEARNING

Aman Sharma*1, Mr. Vibhor Sharma*2

*1,2Maharaja Agrasen Institute Of Technology, Delhi Guru Gobind Singh Indraprastha University, (GGSIPU), India.

ABSTRACT

Deep Neural Networks (DNNs) are powerful models that have achieved excellent performance in complex learning tasks. Currently, the mathematical language model and the neural language model still dominate research in the field of machine translation. Mathematical translation based on statistics today is the fastest but there is a lack of time accuracy. In contrast, a network-based network has high accuracy but has a very slow calculation process. In this study, comparisons between a neural network using the Recurrent Neural Network (RNN) and a mathematical-based network and an n-gram model of two French-English Machine Translation (MT) methods were developed. A quantitative analysis of both types of stress shows that the use of RNN yields a very positive effect. This paper reveals the details of the Deep Neural Network (DNN) and the concept of in-depth learning in the field of natural language processing i.e., machine translation. Now the DNN of the day plays a major role in leaning technologies. A repetitive neural network (RNN) is the best way to learn a machine. It is a combination of a recurrent neural network and a recurrent neural network (similar to the Recursive autoencoder). This paper outlines how to train a repetitive neural network for rearrangement for a source to identify. The default encoder helps to reconstruct the vectors of the target language. Therefore, powerful hardware (GPU) support is required. The GPU improves system performance by reducing training time.

Keywords: Neural Network (NN), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM).

I. INTRODUCTION

Digital translation (MT) is a process of translating text from one language to another using software by including both computer and language information. Initially, the MT system acquires text translation into the source language by simply associating the meaning of the words in the source language with the target language with the help of grammar. However, such methods did not produce good results due to their failure to capture the various sentence structures in the language. This process of translation is time-consuming and requires skilled craftsmen in both languages. Subsequently, integrated translation methods such as mathematical translation (SMT) and neural translation (NMT) technology have been introduced to address the challenges of legal-based approaches. MT has already shown promising results in bilingual translation. In contrast to SMT, which requires sub-components trained separately in translation, NMT uses one large neural network for training. This structure is made up of encoder and decoder networks where the encoder uses input sentences to produce vector representations, and then the decoder takes this vector and extracts In general, both coding and coding networks are designed using duplicate neural networks (RNN) or short-term memory (LSTM) unit (GRU) or RNN to two bids, which are more useful than RNN Or RNN, especially LSTM, theory p proven to handle long-term dependence on sentences translation of long sentences remains an unresolved issue. This problem is being addressed by a process called the attention mechanism. The working principle of the attention-grabbing approach is to look at different parts of source sentences, which contribute more to the prediction of each target word and translate, rather than taking the whole sentence and translating it. This helps to align the words from the source with the words of the language. However, the effectiveness of the NMT system depends largely on the size and quality of the corresponding corpus because unlike the SMT system, which has a different language model, NMT directly detects the target language of a particular basic language sentence. In-depth reading is a newly used method of machine translation. Unlike traditional machine translation, neural machine translation is a better choice for accurate translation and provides better performance. DNN can be used to upgrade traditional systems sequentially to make them work better. A variety of in-depth reading methods and libraries are needed to create a better machine translation system. RNNs, LSTMs, etc. They are used to train a program that will translate a sentence from one language into another. Converting appropriate networks and in-depth learning strategies is a good choice because it has slowed down

the process of increasing the accuracy of the translation system. Neural Machine Translation (NMT) is a method that enables a well-known machine translation process using a large network of neural networks to predict or detect the high sequence of words, basically modeling or creating all the sentences in one integrated way. It has gained acceptance in many major projects. NMT systems use continuous presentations that significantly reduce the problem of sparsity, and use more extreme cases, thus reducing the environmental problem. Many of the issues and shortcomings of traditional machine translation systems are eliminated by the new method, NMT. Deep Neural Machine Translation i.e., Deep NMT extension of NMT. They both use a large neural network with the only difference being that deep neural machine translation processes multiple layers of neural network in one place just as it incorporates advanced technology compared to NMT. A bidirectional recurrent neural network used as an encoder used by the neural network to insert a basic sentence and a second RNN known as a decoder used to predict words or sentences in the target language used.

I am building a deep neural network that acts as part of a machine translation pipeline. The pipe accepts certain language text as input and returns it to English text. The goal is to achieve the highest possible translation accuracy.

Below is a summary of the various steps for preparation and modeling. Advanced steps include

- **Pre-installation:** upload and review data, cleaning, tokens, wrapping
- **Modeling**: building, training, and modeling
- **Predicting:** produce some English to French translations, and compare the translation of the results to a true translation of the world
- **Iteration:** iterate on the model, experimenting with different architectures

II. RELATED WORK

Before reporting the proposed model, we should look at recent activities that focus on the use of neural networks in SMTs to improve translation quality. Schwenk, in his paper, proposed using the feed-forward neural network to earn points in pairs. You have used a feed-forward neural network with a fixed Size input consisting of seven words, with zero paddings of short phrases. The program also had a fixed size output containing seven words for the output. But whenever we talk about real-world translations, the length of the phrase can vary greatly. Thus, the neural network model used must be able to handle phrases of varying lengths. Because for this purpose, we decided to use RNNs. Like Schwenk's paper, Devlin et al have also used a feed-forward neural network to produce translations, but they predict one word in a targeted sentence at a time. The system had an amazing performance than the aforementioned model. However, the use of feed-forward neural networks requires the use of phrases of limited size for optimal performance. Zou et al. it was proposed to study bilingual embedding of words/phrases, in which they used to calculate the distance between the phrase phrases and used it as an additional adjective to hit two pairs of SMT program phrases. In their paper, Chandar et al. trained the feed-forward neural network to read the input phrase map in the output sentence using the word bag method. This is closely related to the proposed model in Schwenk's paper, except that their representation of inserting the phrase is word-bag. Gao et al. suggested a similar way to use the word bag again. The same code embedding method used by the two RNNs was suggested by Socher et al, but their model was restricted to single-language placement. Recently, another model of encoder-decoder using RNN was proposed by Auli et al, in which the decoder was placed in the representation of the source sentence or source context. Kalchbrenner and Blunsom, in their paper, proposed a similar model using the concept of encoder and decoder. They used a convolutional n-gram (CGM) model for the coder component and a combination of the inverse CGM and RNN component of the decoder. The testing of their model was based on extracting the best list of phrases in the SMT word table.

III. METHODOLOGY

In this research work, Neural Machine Translation (NMT) is considered a method. NMT has been introduced as a new way of dealing with the many shortcomings of traditional machine translation systems. Specifically, e.g. The creation of input texts for the corresponding outgoing text. Its structure consists of two duplicate neural networks (RNNs), one used to find the sequence of text input i.e., coding, and the other used to produce translated translation text i.e., decoder.

As human beings, we read a complete sentence or text, understand its meaning, and offer a translation. Neural

Machine Translation (NMT) mimics that!

The encoder converts the input sentence to the "say" component that is decoded to provide translation.

Load & Examine Data

The data used for this project is from the language translation manythings.org. The dataset is a deu-eng that has a translation of the German language into English. This database contains 221533 different sentences in German and their translation into English.

```
English sample 1:  new jersey is sometimes quiet during autumn , and it is snowy in april .
French sample 1:  new jersey est parfois calme pendant l' automne , et il est neigeux en avril .

English sample 2:  the united states is usually chilly during july , and it is usually freezing in november .
French sample 2:  les états-unis est généralement froid en juillet , et il gèle habituellement en novembre .

English sample 3:  california is usually quiet during march , and it is usually hot in june .
French sample 3:  california est généralement calme en mars , et il est généralement chaud en juin .

English sample 4:  the united states is sometimes mild during june , and it is cold in september .
French sample 4:  les états-unis est parfois légère en juin , et il fait froid en septembre .

English sample 5:  your least liked fruit is the grape , but my least liked is the apple .
French sample 5:  votre moins aimé fruit est le raisin , mais mon moins aimé est la pomme .
```

Data Cleaning

In the first phase of the project, we will clean up the data. Take away the personality and change it to a lesser one. The most important step in any project, especially in NLP. The information we work with is often more than unplanned so there are some things we need to take care of before jumping into the model building component

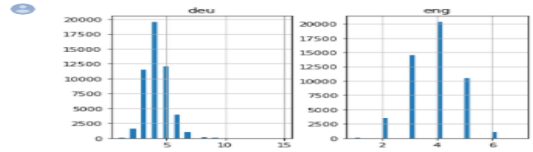
```
[ ] import string
    deu_eng[:,0] = [s.translate(str.maketrans('','',string.punctuation)).lower() for s in deu_eng[:,0]]
    deu_eng[:,1] = [s.translate(str.maketrans('','',string.punctuation)).lower() for s in deu_eng[:,1]]

    deu_eng

    array([['hi', 'hallo',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #380701 (cburgmer)'],
           ['hi', 'grüß gott',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #538123 (CM) & #659813 (Esperantostern)'],
           ['run', 'lauf',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #906328 (papabear) & #941078 (Fingerhut)'],
           ...,
           ['i wholeheartedly agree', 'ich stimme rückhaltlos zu',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #1488273 (Spamster) & #1693172 (al_ex_an_der)'],
           ['i will always love you', 'ich werde dich immer lieben',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #853146 (piksea) & #395302 (xtofu80)'],
           ['i will be back by nine', 'um neun bin ich wieder zurück',
            'CC-BY 2.0 (France) Attribution: tatoeba.org #72281 (CK) & #345033 (lilygilder)']],
          dtype='<U537')
```

Exploratory Data Analysis

Scatterplot And populate the lists with sentence lengths

The Seq2Seq model requires us to convert all inputs and output sentences into complete sequence of fixed lengths. Accurate - the maximum length of German sentences is 11 and that English phrases are 8

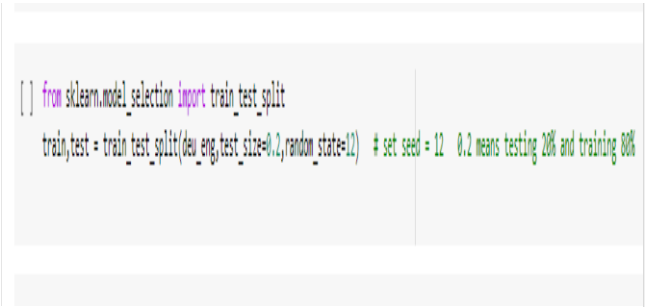
```
# empty lists
eng_l = []
deu_l = []

# populate the lists with sentence lengths
for i in deu_eng[:,0]:
    eng_l.append(len(i.split()))

for i in deu_eng[:,1]:
    deu_l.append(len(i.split()))
length_df = pd.DataFrame({'eng':eng_l, 'deu':deu_l})

length_df.hist(bins = 30)
plt.show()
```



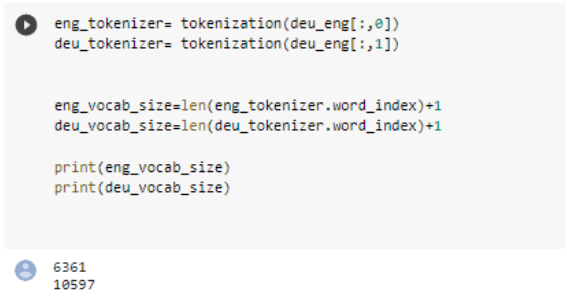
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

Vectorization

Next, vectorize our text data using the Camera's Tokenizer () section. It will turn our sentences into numerical sequences. After that, we can attach that sequence of eggs to make all the sequences of the same length. We will prepare the work to create the token. i.e., we will prepare tokens in both German and English sentences. Here the dataset is made by taking aggregate values to understand the data easily.



```
[ ] from sklearn.model_selection import train_test_split
    train,test = train_test_split(deu_eng,test_size=0.2,random_state=12)  # set seed = 12  0.2 means testing 20% and training 80%
```



```
eng_tokenizer= tokenization(deu_eng[:,0])
deu_tokenizer= tokenization(deu_eng[:,1])


eng_vocab_size=len(eng_tokenizer.word_index)+1
deu_vocab_size=len(deu_tokenizer.word_index)+1

print(eng_vocab_size)
print(deu_vocab_size)
```

```
6361
10597
```

IV. DEVELOPING A MODEL

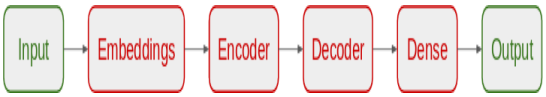
First, let's separate the RNN construction at a higher level. Referring to the diagram above, there are a few parts of the model that we should note:

- **Inputs**- The input sequence is set in the model with the same name at all times. Each word is coded as a separate whole number or one-hot encoded vector that maps in English database vocabulary.
- **Embedding Layers**- Embedding is used to convert each word into a vector. The size of the vector depends on the complexity of the vocabulary.
- **Recurrent Layers (Encoder)**- This is where the context from the word vector in the past steps is applied to the current word finder.
- **Dense Layers (Decoder)**- These are fully integrated layers that are used to specify input with the correct translation sequence.
- **Outputs**- Results are returned as an integer sequence or coded with one hot code that can be mapped to a French database vocabulary.

We will begin by describing our Seq2Seq type structure:

• In embedding, we will use the embedding layer and the LSTM layer

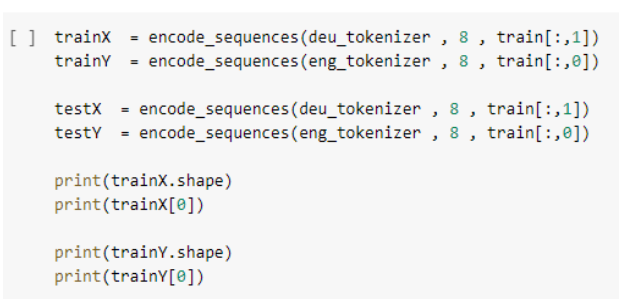
• In the decoder, we will use another layer of LSTM followed by a thick layer



Shuffle and Split Data

We will include **German sentences as input sequences and English sentences as target sequences.** This should be done in the train set and testing sets.

Prepare Training and Validation Data

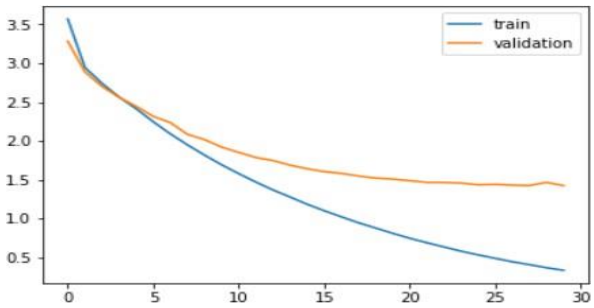


```
[ ]  trainX  = encode_sequences(deu_tokenizer , 8 , train[:,1])
     trainY  = encode_sequences(eng_tokenizer , 8 , train[:,0])

     testX  = encode_sequences(deu_tokenizer , 8 , train[:,1])
     testY  = encode_sequences(eng_tokenizer , 8 , train[:,0])

     print(trainX.shape)
     print(trainX[0])

     print(trainY.shape)
     print(trainY[0])
```

```
(40000, 8)
[203 102  48 397   0   0   0   0]
(40000, 8)
[ 11   3  23  16 385   0   0   0]
```

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:06/June-2021 Impact Factor- 5.354 www.irjmets.com

Compare The Training Loss & The Validation Loss



As you can see from the above structure, the loss of validation ceased to decrease after 20 Epochs. Finally, we can upload a database and make predictions on invisible data - testX.

Converting Predictions into Text (English)

These predictions are numerical sequences and we need to convert these numbers into their corresponding names

```
preds_text = []
for i in preds:
      temp = []
      for j in range(len(i)):
          t = get_word(i[j], eng_tokenizer)
          if j > 0:
              if (t == get_word(i[j-1], eng_tokenizer)) or (t == None):
                  temp.append('')
              else:
                  temp.append(t)
          else:
                if(t == None):
                      temp.append('')
                else:
                      temp.append(t)

      preds_text.append(' '.join(temp))
```

Evaluating Model's Performance

Let's put the first English sentences in the test database and the predicted sentences in the data framework.

Model's Sensitivity

Our Seq2Seq model does a decent job. But there are a few situations where it is difficult to understand keywords. For example, translates "im tired of boston" to "im am boston".

These are the challenges you will always face in NLP. But these are not static obstacles. We can alleviate such challenges by using more training data and building a better (or more complex) model.

Making Predictions and Results

We can print some real vs predictions to see how our model works

```
pred_df.head(15)
```

	actual	predicted
0	do you like my shoes	do you like my shoes
1	i was tired and cold	i was tired and cold
2	toms speech bored me	toms speech bored me
3	tom hates children	tom hates children
4	i love this chair	i love this chair
5	its very cheap	its very cheap
6	tom looks shaken	tom looks shaken
7	i thought that too	i thought that too
8	im hitting the road	im on the
9	hes innocent	hes innocent
10	youre the expert	youre the expert
11	is that your son	is this your son
12	it was simple	it was simple
13	please hurry	please hurry
14	i was late to school	i am to

```
pred_df = pd.DataFrame({'actual' : train[:,0], 'predicted' : preds_text})
```

V. CONCLUSION

The ability to talk with each other is a essential a part of being human. There are about 7,000 one-of-a-kind languages worldwide. As our world becomes more and more connected, language translation provides a critical cultural and economic bridge between people of different nationalities and races. To meet these desires, technology firm's area unit investment heavily in computational linguistics These investments and recent developments in in-depth learning have yielded significant improvements in translation quality. According to Google, the transition to deep reading has produced a 60% increase in translation accuracy compared to the sentence-based approach previously used in Google Translate. Today, Google and Microsoft can translate more than 100 languages and are closer to the accuracy of most of them. From the above, it can be concluded that a language translator needs to be developed to find the most effective way to use it effectively.

FUTURE SCOPE

The results will provide significant contributions to:

- Business in international trade, investment, contracts, finance
- Travel trade, procurement of goods and services abroad, customer support
- Media by accessing search information, sharing information on social networks, local content creation, and advertising. Ideas for sharing ideas, collaborations, translation of research papers.

VI. REFERENCES

[1] M. Anand Kumar, V. Dhanalakshmi,

[2] K. P. Soman and S. Rajendran, Factored statistical machine translation system for English to Tamil language, Pertanika J. Soc. Sci. Hum. 22 (2014), 1045–1061.

[3] P. J. Antony, Machine translation approaches and survey for Indian languages, Int. J. Comput. Linguist. Chinese Language Processing 18 (2013),47–78.

[4] Anuvaadak, Available from: http://www.mysmartschool.com/pls/portal/portal.MSSStatic.ProductAnuvaadak. Accessed 31 May 2017.

[5] D. Bahdanau, K. Cho and Y. Bengio,Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473

[6] A. Bharati, V. Chaitanya, A. P. Kulkarni, and R. Sangal, Anusaaraka:machine translation in Stages, arXiv preprint cs/0306130 (2003).

[7] CDAC-MANTRA, Available from: https://www.cdacindia.com/html/aai/mantr a.asp. Accessed 31 May 2017

[8] S. Chaudhury, A. Rao, and D. M Sharma, Anusaaraka: an expert system-based machine translation system, in Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on, pp. 1–6, IEEE, Beijing, 2010.

[9] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, arXiv preprintarXiv:1409.1259 (2014).

[10] J. Chung, C. Gulcehre, K. H. Cho and Y. Bengio, Empiric evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555(2014).

[11] S. Dave, J. Parikh, and P. Bhattacharyya, Interlingua-based English– Hindi machine translation and language divergence, Mach. Transl. 16 (2001), 251–304.

[12] M. Denkowski and A. Lavie, Choosing the right evaluation for machine translation: an examination of annotator and automatic metric performance on human judgment tasks, AMTA, 2010.

[13] K. Hans and R. S. Milton, Improving the performance of neural machine translation involving morphologically richlanguages, arXiv preprint arXiv:1612.02482 (2016).

[14] R. Harshawardhan, Rule-based machine translation system for English to Malayalam language, Diss. de mestrado. Coimbatore Amrita School of Engineering (2011).

[15] IISC, Available from: http://ebmt.serc.iisc.ernet.in/mt/login.html.Accessed 31 May, 2017.

[16] G. N. Jha, The TDIL Program and the Indian Langauge Corpora Intitiative (ILCI), in: LREC, New Delhi, India, 2010.