A NEURAL MACHINE LANGUAGE TRANSLATION SYSTEM FROM GERMAN TO ENGLISH

AKASH M[1]

[1]*Department of Computer Science and Engineering, Anna University - CEG campus, Chennai India*

Abstract - *Machine Translation is a Natural Language Processing, attaining significant attention to fully automate the system that can translate source content into the target languages. The traditional systems focus on the word level context while neglecting the sentence level context most often, resulting in meaningless translation. This paper focuses on the Transformer based MT which is used for translation integrated with Fuzzy Semantic Representation (FSR) for handling the occurrence of rare words in the sentence level context. In the existing method there is a mismatch of translation but the proposed system is more superior due to the sentence context inclusion. The algorithms proposed in this paper can be used for multilingual translation systems. The FSR group rare words together and represent sentence level context as Latent Topic Representation (LTR) using convolutional neural network. In particular, our algorithm can achieve a significant performance for WMT En-De bilingual parallel corpus and is used for translation handling the Out of Vocabulary words using clustering of <UNK> tags. The model performance is enhanced with hyper parameter optimization obtaining a significantly high BLEU score, and significantly low TER score with a be*

Key Words*: Natural Language Processing, Neural Machine Translation(NMT), Transformer MT, Fuzzy Semantic Representation(FSR), Latent Topic Representation(LTR), Convolutional Neural Network (CNN).

I. INTRODUCTION

Machine translation (MT) refers to fully automated software which translates source language into target language. Humans may use machine translation (MT) to help them render speech and text into another language, or the MT software may operate without human intervention. Allowing a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word[1].

Automatic or machine translation is one of the most challenging AI tasks given the fluidity of human language. The classical approach of the rule-based machine translation systems were used for translation, which were replaced by statistical methods in the 1990s. In recent times, deep neural network models achieve state-of-the-art results for the language translation that is aptly named as Neural Machine Translation (NMT).

Neural machine translation, or NMT in short, uses the neural network models for machine translation. NMT approaches are more powerful at capturing context beyond phrase boundaries and were shown to better exploit available training data[4]. The key benefit for translation is that a single system can be trained directly on source and target text, thereby not requiring the pipeline of specialized systems which are used in statistical machine translation(SMT). As such, NMT systems are said to be end-to-end systems as only one transformer-like model is required for the translation.

Transformer is an end-to-end model with encoder-decoder architecture for seq-seq translation. The encoder consists of a set of encoding layers that processes the input iteratively one layer after another and the decoder consists of a set of decoding layers that does the same thing to the output of the encoder. Each of the encoder layers function is to process its input to generate word encodings (a.k.a word embeddings), containing information about which parts of the input sentences are relevant with each other. It passes its set of word encodings to the next encoder layer as inputs to get a better translation. Each decoder layer takes the output of the encoder layer as input by taking all the word encodings and processes them, using their incorporated contextual information to generate an output sequence[12].

To achieve this, each of the encoder and decoder layers makes use of an attention mechanism (teacher enforcing), which for each input word weighs the relevance of every other input word and draws information (context) from them accordingly to produce the output sentence. Each layer of decoder also has an additional attention weight to draw additional information from the outputs of previous decoders, before the decoder layer draws information from the encodings. Both the encoder and decoder layers have a feed-forward neural network for additional processing of the outputs, and contain residual connections and layer normalization steps[13].

As in the current scenario there exist several languages in the world. All the languages can't be understood, so there is a need for a language translator between the source and target language. Natural Language Translation, which is a

NLP task, is the process of converting source language into the target language using a Seq-Seq Encoder - Decoder model[14]. A machine translator is a NLP task which solves the problem of taking the input sentence of the source language and giving it to the Neural Machine Translator (NMT) for processing the language and giving the output of the target language. Neural machine translation (NMT) is an approach to machine translation that uses an artificial neural network to predict the likelihood of a sequence of words, typically modeling entire sentences in a single integrated model. Topic modeling techniques, such as Latent Dirichlet Allocation (LDA) is very widely used for to capture semantics of the sentence[2]. This is an effective approach for language translation.

The proposed system uses a Fuzzy Semantic Representation (FSR) of rare words to overcome the <UNK> tag in the sentence along with the Latent Topic Representation (LTR) using CNN to obtain the sentence level context. FSR and LTR are integrated with the NMT model to predict the target sentence given the source sentence. Thus an efficient machine translation is generated using the Transformer based encoder - decoder Machine Translator[15].

II. RECENT WORKS

Kehai Chen et.al[3] have discussed sentence-level context as a latent topic representation and designs a topic attention to integrate source sentence-level topic context information into both Attention based and Transformer based NMT. LTR is represented using a Convolutional Neural Network (CNN). The performance of NMT improves by modeling source topics and translations jointly. This study explored the dependence of source topic information on its sentence-level context and proposed a topic attention that integrated latent topic representations into the existing NMT architecture to improve translation prediction. To enhance target word prediction it needs to exploit explicit source topic information. It focuses on sentence-level context to learn translation knowledge instead of document-level context information, multi-source information, multi-model information or syntax information.

Muyun Yang , Shujie Liu, Kehai Chen , et.al (2020) proposes to build a fuzzy semantic representation (FSR) method for rare words through a hierarchical clustering method to group rare words together, and will be integrated with the help of encoder–decoder framework. Chinese-to-English sentence translation tasks confirm a significant improvement in the translation quality for the proposed method. To best preserve the semantic information, this article proposes a hierarchical clustering embedding approach to fuzzy semantic representation (FSR) of rare words in NMT.

Xing Wang, Zhaopeng Tu, and Min Zhang (2015) have explained a general framework for incorporating the SMT word knowledge into NMT to alleviate above word-level limitations. The NMT decoder makes more accurate word prediction that is to be translated by referring to the SMT word recommendations in both training and testing phases. The SMT word predictions are used as prior knowledge to adjust the NMT word generation probability, which unitizes a neural network-based classifier to digest the discrete word knowledge.

Rui Wang, Masao Utiyama, Andrew Finch and et.al. (2018) have discussed the 4 goals for sentence level NMT domain adaptation. First: The NMT's internal sentence embedding is exploited and the sentence embedding similarity is used to select out-of-domain sentences which are close to the in- domain corpus. Second: propose three sentence weighting methods, i.e., sentence weighting, domain weighting and batch weighting, to balance the data distribution during NMT training. Third: In addition propose dynamic training methods to adjust the sentence selection and weighting during NMT training. Fourth: To solve the multi-domain problem in a real-world NMT scenario where the domain distributions of training and testing data often mismatch, proposed a multi-domain sentence weighting method to balance the domain distributions of training data and match the domain distributions of training and testing data.

Haipeng Sun , Rui Wang , Kehai Chen and et.al (2020) have discussed about empirical relationships between unsupervised neural machine translation (UNMT) and unsupervised bilingual word embedding (UBWE) pre-training and cross-lingual masked language model (CMLM) pre-training. So a novel UNMT structure with cross-lingual language representation agreement to capture the interaction between UBWE / CMLM and UNMT during UNMT training.

Kehai Chen , Tiejun Zhao, Muyun Yang and et.al (2018) have given a novel neural approach to source dependence-based context representation for translation prediction. The proposed model is capable of not only encoding source long-distance dependencies but also capturing functional similarities to better predict translations. It is incorporated into phrase-based translation (PBT) and hierarchical phrase-based translation (HPBT) models. The evaluation metrics used is Case Insensitive BLEU-4 metric which outperforms all the baseline systems with better translation. NMT still performs better than the proposed model with a significantly greater BLEU score[5].

The methodologies and architectures employed in the existing systems for machine translation were studied and the issues in the existing systems include: inadequate training of models, insufficient data, some models are built for a specific language alone and are not applicable for other common languages like English. Some models translate sentences that resemble machine generated sentences that do not sound like the language that humans use to speak[6][7]. Machine Translation is needed to solve this problem and the existing methods and related works that were also studied. Transformer based NMT are more natural and less artificial when compared to other models.

III. PROPOSED SYSTEM

The transformer is an end to end system having encoder-decoder architecture. The encoder consists of a set of encoding layers that processes the input sentence iteratively one layer after another to produce sentence vectors along with the rare words with the help of FSR module and the decoder consists of a set of decoding layers that does the same thing to the output sentence vector of the encoder along with the topic vector to get a more accurate translation.

The function of each encoder layer is to process its input language sentence (English) to generate encodings in vector form referred to as embeddings, containing information about which parts of the inputs are relevant to each other. It passes its set of encodings (obtained from the previous encoding layer) to the next encoder layer as inputs. Each decoder layer does the opposite of encoder, taking all the encodings obtained from the encoder and processes them, using their incorporated contextual information to generate an output sequence. To achieve this, each encoder and decoder layer makes use of an attention mechanism (assigns weight to the input word embeddings that is to be translated), which for each input, weighs the relevance of every other input and draws contextual information from them accordingly to produce the output. Each decoder layer also has an additional attention mechanism which draws information from the outputs of previous decoders (assigns weight to the output word embeddings that is being translated), before the decoder layer draws information from the output of encodings. Figure 1 shows the transformer integrating LTR for sentence context along with LTR to handle rare words which are OOV. Both the encoder and decoder layers have a feed-forward neural network for additional processing of the outputs, and contain residual connections and layer normalization steps.

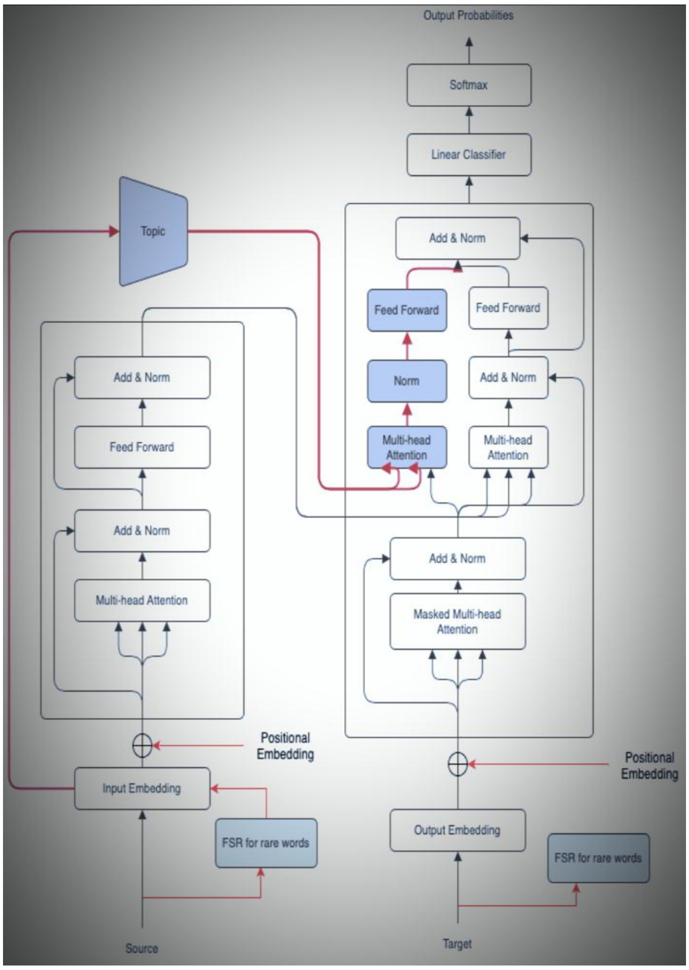


Figure 1: Transformer Architecture with FSR + LTR

The modules required to build an end-to-end Encoder Decoder system for translation is described here. From figure 1 the high level block diagram where the dataset of bilingual parallel English-German corpus is preprocessed to get a clean sentence by removing special characters and punctuation marks. The cleaned dataset is splitted into separate files of German sentences and English sentences. The words which are OOV are given to the FSR most commonly related together under a universal <UNK> tag. The embedding module converts the sentence sequence into its corresponding vector representation. The positional encoding module uses sinusoidal functions to indicate the position of the words in the sentence at both source and target side. The context of the sentence is handled as Latent Topic Representation which uses CNN to extract the context of the sentence. The transformer encoder and decoder which uses a self attention mechanism to translate. Finally the decoded vector is passed to the linear classifier and softmax functions. Finally the google translator is integrated into our NMT system to check the correctness of the predicted sentence using the sentence prediction module.

A. TRANSFORMER ENCODER - DECODER

Each encoder consists of two major components: a self-attention mechanism and a feed-forward neural network. The self-attention mechanism takes in a set of input encodings from the previous encoder and weighs their relevance to each other to generate a set of output encodings. The feed-forward neural network then further processes each output encoding individually. These output encodings are finally passed to the next encoder as its input, as well as the decoders. The first encoder takes positional information and embeddings of the input sequence as its input, rather than encodings. The positional information is necessary for the Transformer to make use of the order of the sequence, because no other part of the Transformer makes use of this[8].

Each decoder consists of three major components: a self-attention mechanism, an attention mechanism over the encodings, and a feed-forward neural network. The decoder functions in a similar fashion to the encoder, but an additional attention mechanism is inserted which instead draws relevant information from the encodings generated by the encoders. Like the first encoder, the first decoder takes positional information and embeddings of the output sequence as its input, rather than encodings. Since the transformer should not use the current or future output to predict an output though, the output sequence must be partially masked to prevent this reverse information flow. The last decoder is followed by a final linear transformation and softmax layer, to produce the output probabilities over the vocabulary[9].

Algorithm :

Continuous Bag of Words - Word2Vec model

Input: One hot vector of input/output sentence of size V
Output: Word Vector Representation - dependence of one word on other words in sentence

```
Begin
1. Padding bits are added to the one hot vector to make equi length sentence
2. Wvn is the weight matrix that maps input X to the hidden layer (V*N dimensional matrix)
3. Wnv is the weight matrix that maps hidden layer output to the final output layer  (N*V dimensional matrix)
4. Hidden layer neurons copy the weighted sum of inputs to the next layer
5. Non linearity Activation function of Softmax is used in the Output layer
End
```

The encoder receives a source sentence x and encodes each prefix using a recurrent neural network that recursively combines embeddings xj for each word position j:

$$\overrightarrow{h}_j = f(x_j, \overrightarrow{h}_{j-1})$$

where f is a non-linear function. Reverse encodings h_j are computed similarly to represent suffixes of the sentence. These vector representations are stacked to form h_j, a representation of the whole sentence focused on position j. The decoder predicts each target word y_i sequentially according to the distribution

$$P(y_i|y_{i-1}, ..., y_1, \mathbf{x}) = g(y_{i-1}, s_i, c_i)$$

where s_i is a hidden decoder state summarizing the prefix of the translation generated so far, c_i is a summary of the entire input sequence, and g is another non-linear function. Encoder and decoder parameters are jointly optimized to maximize the log-likelihood of a training corpus[5].

B. FSR - FUZZY SEMANTIC REPRESENTATION

Algorithm : **Embedding Tree Construction**

```
Begin
1. Train toolkit Word2Vec is used for Monolingual Data
2. Words OOV but appears >3 times are Clustered into M classes by k-means
3. Class Embedding = Mean(Word Embeddings in the class)
4. Each class has <UNKj> containing remaining words that belong to this class (Cosine Similarity)
5. <UNKj> embedding = Mean(Remaining words in class)
End
```

The 3-level embedding tree is constructed based on the K-means clustering and is categorised based on the class which fits it into which is as shown in fig.2

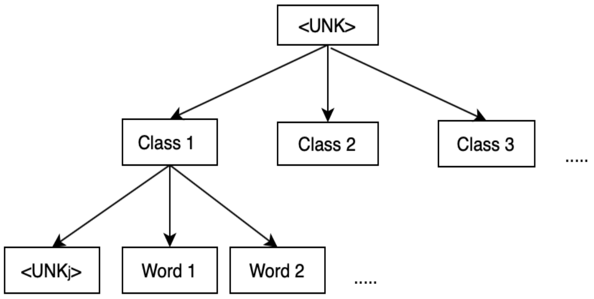


Figure 2 : Classification of words in embedding tree

Algorithm : **FSR Generation**

```
Begin
1. Input Vector is Path from root to the rare word, to deal with data sparseness
2. Differentiate class information and word information for Rare words
3. Computes Source and Target Rare word Input vector
```

End

For the encoder, with previous encoder state h_{i-1}, previous input word embedding Ex_{i-1}, encoder computes the source rare word input vector Exunk with attention mechanism as

$$e_j = \alpha(\boldsymbol{h}_{i-1}, \boldsymbol{Ex}_{i-1}, \boldsymbol{Ep}_j)$$

$$w_j = \frac{\exp(e_j)}{\sum_k \exp(e_k)}$$

$$\boldsymbol{Ex}_{\text{unk}} = \boldsymbol{U}_x \Big(\sum_j w_j \boldsymbol{Ep}_j\Big), \quad j = 1, 2$$

where Ep_2 and Ep_1 are the embeddings of word and class node in the path, respectively, and U_x is a random initialized weight matrix. Similar to the encoder, the target rare word input vector Eyunk is based on previous decoder state s_{i-1}, previous predicted word embedding Ey_{i-1}, previous context vector c_{i-1}, and weight matrix U_y

$$e_j = \alpha(\boldsymbol{s}_{i-1}, \boldsymbol{Ey}_{i-1}, \boldsymbol{c}_{i-1}, \boldsymbol{Ep}_j)$$

$$w_j = \frac{\exp(e_j)}{\sum_k \exp(e_k)}$$

$$\boldsymbol{Ey}_{\text{unk}} = \boldsymbol{U}_y \Big(\sum_j w_j \boldsymbol{Ep}_j\Big), \quad j = 1, 2.$$

The words which are out of vocabulary (OOV) are difficult to translate and needs special attention for those words in a sentence. The FSR module takes care of data sparseness where the words which are

out of dictionary can be translated with better accuracy by Differentiate class information and word information for Rare words.

To integrate the proposed FSR into the existing transformer-based NMT for improving target translation, especially translations of rare words. Specifically, the proposed FSR is only applied to the input representation layer, whose output is the summation of word embeddings and positional embeddings in the input sentence. Take the source sentence as an example, if this word x_j is not in-vocabulary, the learned Ex_{unk} is used to replace the original OOV vector;

if one word x_j is in-vocabulary, its word vector is x_j

$$\overline{\boldsymbol{x}}_j = \begin{cases} \boldsymbol{x}_j, & x_j \in V_s \\ \boldsymbol{Ex}_{\text{unk}}, & x_j \notin V_s. \end{cases}$$

The xj is, then, used to replace the existing xj to learn the input representation.

C. LTR - LATENT TOPIC REPRESENTATION

Figure 3 shows all the layers of CNN model with (D X L) dimensions of the sentence and extracts the meaning of the sentence (context) for getting better translation results.

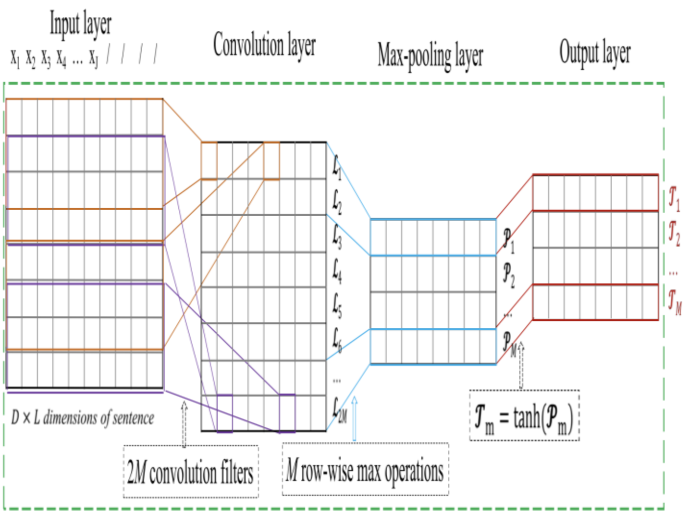


Figure 3 : CNN model for LTR

The layers of CNN extracts the semantics of the sentence based on the topics which are being defined initially. The semantics closely matched with the most appropriate topic.

Algorithm : Extracting Key Information

INPUT :
{T1, T2, T3,..., Tm} is M Topics
X={ x1, x2, x3 ,..., xj } is input sentence (Word embedded)

OUTPUT:

Topic distribution of a sentence - topic context vector

```
Begin
1. J*D input matrix layer is passed to 2M Convolution Filters for 3 consecutive rows where D is the dimension of the word vector and J is the sentence Length
2. The convolution Layer performs M row-wise max operations to generate Max Pooling Layer
3. Max Pooling Layer undergoes Tanh function to generate Output Layer
4. Finally T is mapped to {Key,Value} pair
End
```

Max-Pooling Layer: This takes row-wise maximum over pairs of consecutive rows of L:

$P_m = \max(L_{2m-1}, L_{2m}), 1 \le m \le M$

The resulting output feature map is P:

P = {P1, P2,..., PM}.

Note that a row pair-wise max-pooling operation is applied to L_{2m-1} and L_{2m-1} to gain a D-dimensional topic feature vector Pm. Compared with the max-pooling of the sentence classification task, we use a D-dimensional topic feature vector Pm to represent a topic in the input sentence instead of a topic feature value. As we all know, the vector representation has a better ability to encode word or topic information. Meanwhile, the vector representations of topics are also easier integrated into the existing NMT architecture.

Output layer:This applies the tanh function over Pm to obtain a LTR T m:

$T_m = \tanh(P_m)$,
$T = \{T_1, T_2,..., T_M\}$.

The T is our proposed LTRs, which will later be used to learn the topic context vector for NMT. The model parameters for learning T are as the following:

$\eta = \{W_1,W_2,...,W_{2M}; b_1, b_2,...,b_{2M}\}$.

The CNN model layers extracts the key information as the topic context vector for the NMT to work effectively.

D. DATASET DESCRIPTION

WMT '14 is open source data available on the Stanford site. The modality of the dataset is text sentences. The training data consists of 4.5 Million sentence pairs of both German and English sentence pairs. The training and validation is splitted in the ratio of 80% and 20% respectively. The dataset is of rich text format and must be preprocessed so that it can be fed into the model for training. The dataset also includes the vocabularies of words which occur more frequently, say top 50K frequent words. The dictionary is formed by extracting from alignment data[10][11].

IV. EVALUATION

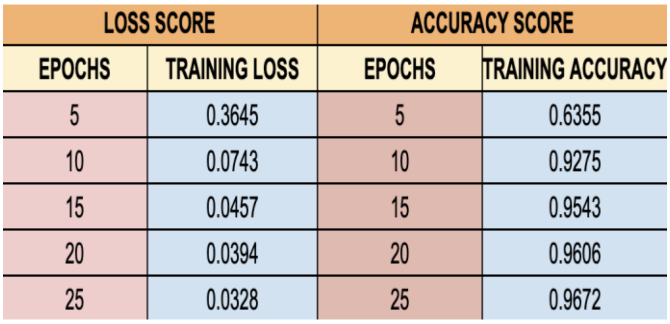
The translated sentences were evaluated using Training accuracy, BLEU and TER metrics. The loss and accuracy of training is considered as one of the most important steps towards evaluation. It is evident from Table 1 that accuracy increases and loss decreases with the increase in epochs. The loss is calculated based on the cross entropy.

Cross-entropy Loss= $-\sum_{i=1}\sum_{j=1} y_{i,j}\log(p_{i,j})$

$$\textbf{Accuracy} = \frac{\textbf{No. of correct predictions}}{\textbf{Total no. of predictions}}$$

Loss = 1 - Accuracy

TABLE 1 : LOSS AND ACCURACY SCORES



LOSS SCORE		ACCURACY SCORE	
EPOCHS	TRAINING LOSS	EPOCHS	TRAINING ACCURACY
5	0.3645	5	0.6355
10	0.0743	10	0.9275
15	0.0457	15	0.9543
20	0.0394	20	0.9606
25	0.0328	25	0.9672

Bilingual Evaluation Understudy (BLEU) is a score for comparing target sentence translation with one or more reference translations. The formula for calculating BLEU Score is given as :

BLEU = BP . exp($\sum W_n \log p_n$ **)**

$$\textbf{Brevity Penalty (BP)} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \le r \end{cases}$$

where r - count of words in a reference translation ;
c - count of words in a candidate translation.

Figure 4 shows that the training loss decreases as the number of training epochs increases. At the range of 15 to 25 epochs the loss doesn't decrease significantly and reaches the optimal point with a loss of almost 3% only. Further increasing the epochs there is no further improvement and the training time also increases with the increase in the number of epochs.

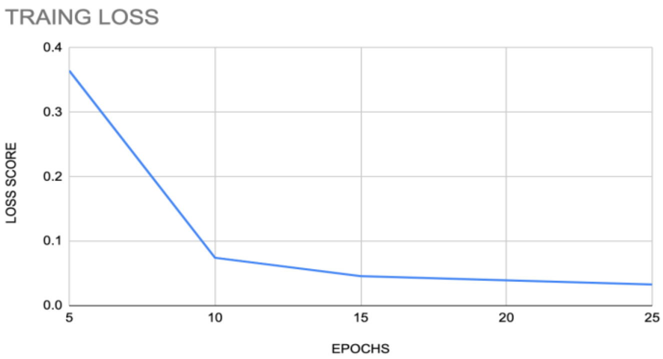


Figure 4: Graph for Training Loss Score vs Epochs

Figure 5 shows that the training accuracy increases as the number of training epochs increases. At the range of 15 to 25 epochs the accuracy doesn't increase significantly and reaches the optimal point with an accuracy of almost 98%. Further increasing the epochs there is no further improvement and the training time also increases with the increase in the number of epochs.

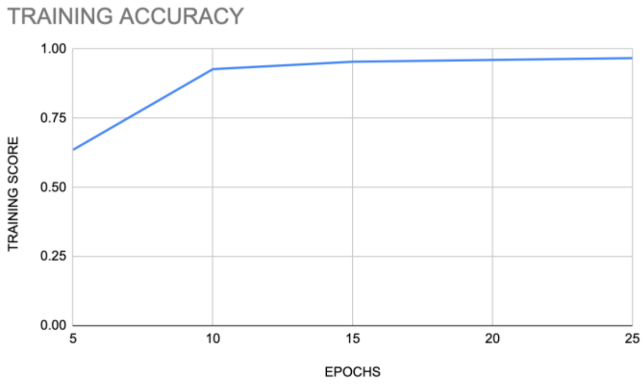


Figure 5: Graph for Training Accuracy Score vs Epochs

Figure 6 shows the comparative study of the loss and accuracy for 5 different epochs, which shows the improvements for epochs range between 15 to 25.

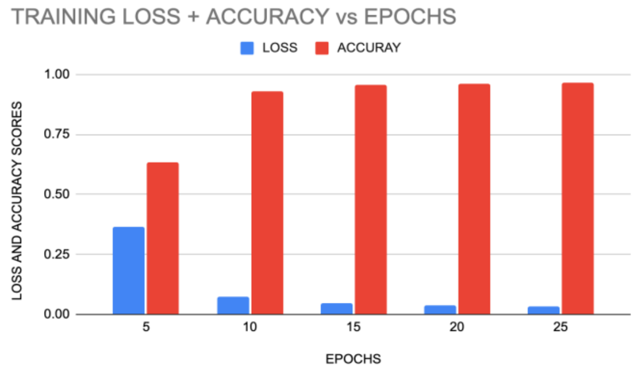


Figure 6: Bar Plot for Loss and Accuracy with Epochs

From Table 2, the BLEU score is calculated for the translated sentence by our system with the google translated sentence for the sample test cases and achieve a maximum score of 0.84 for the third test case. We infer that the short sentences get translated optimally by our algorithm.

TABLE 2 : BLEU SCORE FOR 5 TEST CASES

TEST CASES BLEU SCORE	
TEST CASE NO.	BLEU SCORE
T1	0.67
T2	0.71
T3	0.84
T4	0.64
T5	0.8

From the figure 7, shows the BLUE evaluation metric for 5 different sample sentences from German to English translation. The scores are obtained based on the above mentioned BLUE score formula (Brevity Penalty) for the translated sentences.

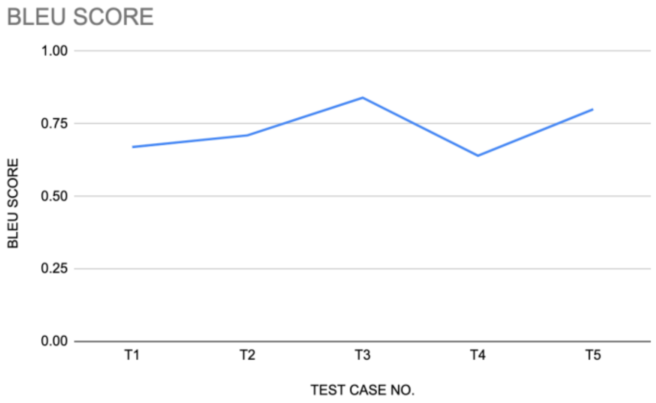


Figure 7: BLEU score for 5 sample translations

Translation Error Rate is the number of edits required to change a system output into one of the references. The formula for calculating TER Score is given as:

$$\mathbf{TER} = \frac{\textbf{Number of edits average}}{\textbf{Number of reference words}}$$

TABLE 3 : TER SCORE FOR 5 TEST CASES

TEST CASES TER SCORE	
TEST CASE NO.	TER SCORE
T1	0.07
T2	0.08
T3	0.33
T4	0.04
T5	0.14

From table 3, the TER score is calculated for the translated sentence by our system with the google translated sentence for the sample test cases to minimize the translation error rate of the system. Similarly the average TER score for the entire corpus can be calculated.

From the figure 8, shows the TER evaluation metric for 5 different sample sentences from German to English translation. From the graph above the Testcase T3 has peak as the sentence length is long. So our model is limited to short length translation sentences.

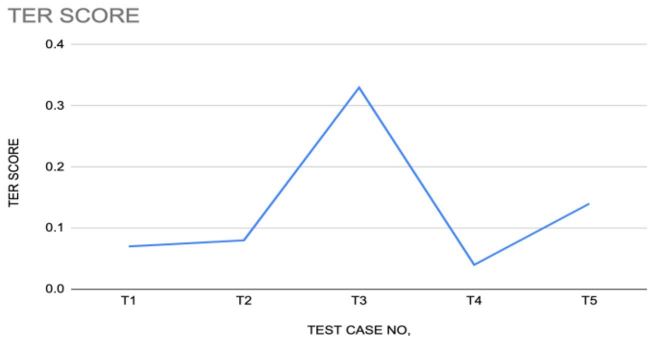


Figure 8: TER score for 5 sample translations

The performance of the system is evaluated based on the BLEU score and the TER score. The aim is to maximize the BLEU score so that the predicted translation which is closely related to the reference translation (human interference) and to minimize the TER score so that the error in the translation becomes less. For sample 5 test cases are evaluated using the above two scores and this can be extended to the entire corpus if required.

For training, the total time taken by the batches per epoch is tabulated below. Each epoch has 6 batches of size 100 to be trained and the sum of all the time of batches per epoch is calculated. The GPU with RAM of 25 GB is used to run the model and the time taken per epoch is given in seconds (sec). The training time for every 5 epochs increases by a slight margin. From the training time for 50000 training sentences the time for every epoch increases slightly and the average time taken for every epoch is almost 45 seconds. To minimize the time per epoch, a faster GPU with increased RAM size is needed.

The total time taken by the batches per epoch is tabulated below. Each epoch has 6 batches of size 100 to be trained and the sum of all the time of batches per epoch is calculated and shown in Table 4. The GPU with RAM of 25 GB is used to run the model and the time taken per epoch is given in seconds (sec). The training time for every 5 epochs increases by a slight margin.

TABLE 4 : MODEL TRAINING TIME VS EPOCHS

EPOCHS VS TIME	
EPOCHS	**TIME (in sec)**
1-5	189.2447
6-10	193.5873
11-15	193.3618
16-20	193.8892
21-25	194.2428

The plotting of Training Time for different epochs is shown in fig. 9, for every 5 epochs the time is almost constant but the time is largely dependent on the corpus size. The time taken for training is directly proportional to the corpus size and the number of epochs. The majority of time is spent in model training for highly accurate translation, so the optimal number of epochs must be maintained for better efficiency.

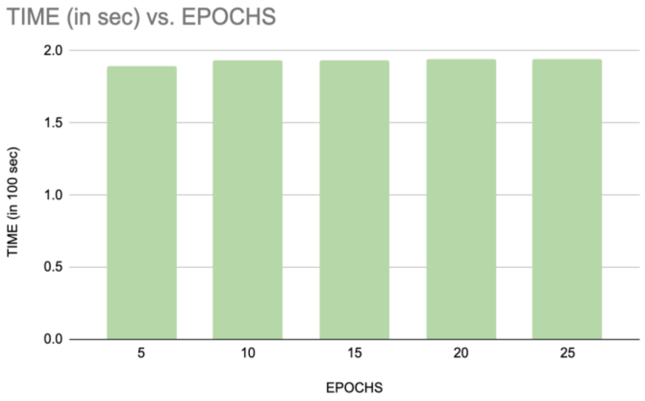


Figure 9: Bar Plot of Training Time for different epochs

V. RESULTS

Thus from the above tables we have the translated sentences output by the model with the google translators output. The end to end system for the translation of simple german sentences to english sentences is obtained. Thus the sample test cases with the Transformer based NMT along with FSR to handle OOV words and LTR to handle sentence context has better translation BLUE scores and minimal TER scores in comparison with all other baseline models. The preprocessing of sentences is done with the help of regular expression by removing the special characters from the sentence, for both german as well as english sentences along with <start> and <end> tag to denote the starting and ending of sentence. The words in a sentence, which has been preprocessed are tagged with numbers based on the ordering given by the dictionary, for both german as well as english sentences. The Transformer model along with FSR and LTR is trained with 25 epochs to get optimal results for the translation. The training along with the accuracy and loss for each batch in epochs are done. The WordtoVec model (pretrained model) is used to obtain the word embeddings (encoded vector). The German to English translation for the trained data is tested and compared with the google's translation for validating the translated output. The attention graph is plotted which is similar to heatmap (i.e) the brightest color is taken as the matched translated word. The transformer model uses the self attention mechanism for the translation.

VI. CONCLUSION

In this work, building an end to end Language Translation system which uses a Transformer based NMT to convert German Language sentences to English Language sentences. The context of a sentence is taken into account so that the target sentence translation will be of the same meaning of the source sentence. The Self Attention mechanism used in the transformer predicts a better translated sentence. The rare words which are OOV in the sentence are handled using FSR with the help of hierarchical clustering which is integrated with the transformer model. The context of the sentence acts as the LTR which uses CNN is also integrated within the system. The Sequence to Sequence system for the dataset takes a significant time for modeling the system. The experiments indicated that the proposed NMT system achieves good performance with a better BLEU score compared to conventional MT systems and minimizes the TER significantly.

VII. FUTURE WORKS

The future work for this project lies in the module of machine translator, where more attention is needed to generate more accurate translation as in accordance with the human translated output rather than google translated output. This work will be related to multiple reference translation using the GAN model so that it achieves a close match with human translation. These translation systems can be applied to many other applications like chatBot systems, Helpdesk applications etc.

Thus we have translated sentences from German to English with the help of Google Translation for the limited sentence sizes. In the near future we are planning to extend this system for lang sentences along with multilingual translations from any to any known scripted languages with the help of google translation package.

REFERENCES

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[2] Cao, Ziqiang, et al. "A novel neural topic model and its supervised extension." Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015..

[3] Chen, Kehai, et al. "A neural approach to source dependence based context model for statistical machine translation." IEEE/ACM Transactions on Audio, Speech, and Language Processing 26.2 (2017): 266-280.

[4] Chen W., Matusov E., Khadivi S., and Peter J., "Guided alignment training for topic-aware neural machine translation," in The 38th Annual Meeting and Symposium of the Antenna Measurement Techniques Association, vol. 1, Austin, Texas, November 2016, pp. 121–134.

[5] Chitnis R. and DeNero J., "Variable-length word encodings for neural translation models," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2088–2093.

[6] Chung J., Cho K., and Bengio Y., "A character-level decoder without explicit segmentation for neural machine translation," in Proceedings of the 54th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2016, pp. 1693–1703.

[7] Fadaee M., Bisazza A., and Monz C., "Learning topic-sensitive word representations," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 441–447.

[8] Haipeng Sun,Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao."Unsupervised Neural Machine Translation With Cross-Lingual Language Representation Agreement." IEEE/ACM Trans on audio,speech,and language processing,Vol. 28, 2020.

[9] Jean S., Cho K., Memisevic R.,and Bengio Y.,"On Using Very Large Target vocabulary for neural machine translation," in Proc. 53rd Annu. Meet. Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., Jul. 2015, pp. 1–10.

[10] Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao."Neural Machine Translation with Sentence-level Topic Context."IEEE/ACM Trans on audio,speech,and language processing, doi:10.1109/TASLP.2019.2937190.

[11] Kim Y., "Convolutional neural networks for sentence classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[12] Muyun Yang , Shujie Liu, Kehai Chen ,Hongyang Zhang, Enbo Zhao, and Tiejun Zhao."A Hierarchical Clustering Approach to Fuzzy Semantic Representation of Rare Words in Neural Machine Translation." IEEE Trans.on fuzzy logic, Vol. 28, no. 5, May 2020.

[13] Rui Wang, Masao Utiyama, Andrew Finch, Lemao Liu, Kehai Chen, and Eiichiro Sumita."Sentence Selection and Weighting for Neural Machine Translation Domain Adaptation."IEEE/ACM Trans. on audio, speech, and language processing, doi: 10.1109/TASLP.2018.2837223.

[14] Snover, Matthew, et al. "A study of translation edit rate with targeted human annotation." Proceedings of association for machine translation in the Americas. Vol. 200. No. 6. 2006.

[15] Xing Wang, Zhaopeng Tu, and Min Zhang" Incorporating Statistical Machine Translation Word Knowledge into Neural Machine Translation".Journal of latex class files, Vol. 14, no. 8, August 2015.