MOS11

by Mo S

Submission date: 26-Jul-2022 09:28PM (UTC+0530)
Submission ID: 1875475220
File name: Research_paper_Monika.docx (1.22M)
Word count: 11278
Character count: 63500

Multilingual Neural Machine Translation using Attentional Encoder Decoder and Transformer Network

***Abstract*— Neural machine translation (NMT) seeks optimal translations using a parallel corpus consisting of several parallel sources and target sentence instances and using this to train neural networks. This project uses NMT to develop a messaging application for translating messaging of English into some of the most spoken languages around the world such as Spanish, French, German, Hindi, and Bengali. NMT has been implemented in this project using two types of networks: attentional encoder-decoder networks and transformer networks. The predicted translation performance of the model for different languages has been evaluated using the Bilingual Evaluation Understudy(BLEU) score, Word Error Rate (WER) of the translations, and METEOR score. Translations are also evaluated by human evaluators to assess the quality of translation in terms of its adequacy, fluency, and correspondence with human-predicted translation. Our implemented transformer model achieved a BLEU score of 39.21 and 39.91 for German and French translations respectively. We have developed an android application for multilingual messaging using the trained transformer models.**

***Keywords*—Neural machine translation, Transformer, Attention mechanism, Encoder-Decoder, BLEU, Android**

I. Introduction

Communication and information exchange between people is necessary not only for business purposes but also for people to share feelings, thoughts, opinions, and facts. But language barriers between different countries pose a significant problem for the effective exchange of information. This language barrier is a primary reason for ineffective communication. Information sharing between people is not only important for business purposes but also necessary for sharing feelings, opinions, and acts. To this end, translation plays an important role in minimizing the communication gap between different people.

India, for instance, is a multilingual country with people from different states speaking different regional languages. It has 23 constitutionally recognized official languages and several hundred unofficial local languages. Despite the population of India is approximately 1.3 billion, only about 10% of them speak English. Some research shows that out of these 10% English speakers only 2% can speak, write, and read English well, and the rest 8% can merely understand simple English and speak broken English with an amazing variety of accents. Considering a significant amount of valuable resources is available on the web in English and most people in India can not understand it well, it is essential to translate a wide range of content into local languages to facilitate effective communication among people.

Considering the enormous amount of information available, manually translating the content is not feasible. Also, it is not feasible to have human translators everywhere, we need effective approach which do this job with as little human effort as possible. Hence, it is essential to translate text from one language to another language automatically. Machine Translation (MT) is defined as the process of translating text or speech from one natural language to another, with as little human effort as possible. Machine Translation bridges communication barriers and eases interaction among people having different linguistic backgrounds. Machine Translation mechanisms make use of a range of linguistic resources and techniques for the prediction of translation. MT aims to achieve quality translations that are semantically equivalent to the source sentence and syntactically correct in the target language. MT performs substitution of words but this procedure alone is not enough, as recognition of whole phrases and their closest counterparts in the target language are necessary for context recognition. This enables the machine to translate better results based on the source and target sequences from the parallel corpus.

This project aims at an accurate and effective translation of English to the 5 most commonly spoken languages around the world using neural machine translation. This project implements NMT using two types of networks: attentional encoder-decoder network and transformer network. The purpose of this project is to develop a multilingual messaging android application that will translate the mentioned language to English and vice-versa. We have worked with 5 languages: English-German, English-Spanish, English-French, English-Hindi, and English-Bengali. This project is one of the most difficult applications of NLP. The types of neural networks for this purpose comes under the class of Seq2Seq models. The use of a transformer network for NMT is better than that of encoder-decoder models using LSTMs. The purpose of this project is to evaluate the performance of these two types of networks. The easy availability of parallel corpora of the mentioned languages was useful for the creation of a training dataset for implementation. The corpus has been cleaned and necessary preprocessing is carried out before modeling. The encoder-decoder and transformer networks are implemented using Keras API of the Tensorflow framework.

A lot of work has been done on NMT. Previous works in NMT have been done in these particular languages. There are benchmarks available for English-French and English-German on WMT'14 datasets with about 38.95 and 24.67 BLEU scores respectively. [1] The approach in this project is to develop an NMT model with better translation accuracy compared to previously developed models. The models developed will be compared using the BLEU score, WER score and METEOR score as the use of accuracy alone is not an ideal metric for seq2seq models. The outcome of the project is an android messaging application. The purpose of this application is to help peers-to-peer communication in the language of their choice. The application also has a translation feature that will help to find the translation of English to a language of the user's choice. A minimum of 8GB RAM. And GPU with at least 2GB RAM.

The rest of the paper is presented in five sections. The introduction is followed by a discussion of a few related works related to this project. The next section presents a

background study of NMT along with related techniques related to it. The following section mentioned the methodology of the project which is followed by the result and discussion of the implemented NMT models and their evaluation. The paper is concluded in the last section.

II. Related works

Rule-based machine translation is the first approach toward machine translation. This strategy utilizes a lot of human efforts for part-of-speech tagging, syntactic parsers, and bilingual dictionaries. Dr. Siddhartha Ghosh, Sujata Thamke, and Kalyani U.R.S [2] in their paper have used such a rule-based translation strategy to translate Telugu to Marathi and vice-versa. Their research focuses on idioms and proverbs of both languages. The direct translation was used to translate these two languages as they have the same grammatical arrangement of sentences. Their approach is based on POS tagging and the authors have concluded that many complex sentences have their words interchangeable to get translation into the final language in direct translation.

Zhixing Tan et al.[3] in their paper have discussed methods related to NMT and have mentioned strategies related to architectures, decoding, and augmentation. The authors have mentioned two previously developed techniques: statistical machine translation(SMT) and neural machine translation(NMT). However, their research is focused on NMT and its associated architects along. They have mentioned the beam search algorithm used for NMT. Their research has mentioned the use of attention mechanisms [4] in transformer networks. Unsupervised NMT was also mentioned by the authors for translation of languages whose parallel translated corpora are not available. They have also summarized open-source NMT tools and tools for evaluation and analysis. The authors have concluded their paper by discussing the challenges and future scope for NMT tasks.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio [5] in their paper have compared and discussed two models for NMT, the encoder-decoder model using RNN and their proposed model: gated CNN. The models are implemented using beam-search algorithms for translation of English to French. They evaluated the models using the BLEU score. Their proposed model, grConv consists of 2,000 neurons as compared to 1,000 neurons of the RNNenc model. On training the models on English-French pairs, the authors have concluded that the performance of NMT suffers greatly from the length of associated sentence length. They have concluded that both the models can translate the sentences with similar accuracy, however, their performance decreases when the sentence length increases.

Felix Stahlber [6] in this research has presented a review of all the present NMT strategies and techniques. They have discussed in detail the available NMT architectures and associated algorithms. They have discussed the use of transformer networks with attention mechanisms and greedy and beam search algorithms. They have reviewed commonly used architectures such as recurrence, convolutions, and attention. They have also discussed the advantages and disadvantages of different strategies along with the factors affecting the performance of mentioned models. They have concluded the paper by discussing the future aspects of machine translation.

Amarnath Pathaket and Pa[illegible]a Pakray [7] in their paper have implemented NMT using an encoder-decoder network with attention to the translation of English to Hindi, Punjabi, and Tamil from a parallel corpus of Indian languages. They have evaluated their model using the BLEU score. The experimental setup used by the authors in their research used BLEU to evaluate the translation performance on different epochs, data sizes, and different sentence lengths. The authors have concluded that the translation performance depends largely on the size of training corpora and the performance of the models heavily enhanced by using the attention mechanism in the encoder-decoder network.

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio [8] in their research have proposed a novel approach to neural machine translation. They have translated English to French with a parallel corpus of 61M words. The authors in their proposed model RNNsearch have extended the basic encoder-decoder by letting a model (soft-)search for a set of input words. The authors have shown that this approach frees the model to encode a fixed-length vector and lets the model focus on relevant information for the generation of the next word. Their model outperforms the traditional RNNencdec model. Their proposed model performed well while predicting longer sentences. The authors concluded that their model achieved results comparable to existing phrase-based statistical machine translation.

Kartik Revanuru, Kaushik Turlapaty and Shrisha Rao [9] in their research paper have created a system that has various models and they have applied the Neural Machine Translation techniques for the creation of this system. It has been applied to six Indian language pairs. In their research paper, they have demonstrated that they were able to achieve good accuracy with fewer data and shallow networks of two layers. After comparing their test set with the Google translate, their models for Urdu-Hindi, Gujarati-Hindi, and Punjabi Hindi outperformed Google translate with the BLUE score of 17, 30, and 29 respectively. The authors concluded the research paper by discussing the future aspects of how it can also be extended for real-time speech to speech translation.

Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu and Ming Zhou [10] in their research paper have proposed a novel Sequence-to-Dependency Neural Machine Translation Method. In this method, the construction and the modeling of the target word sequence and its dependency structure will be done together. This structure will be used as context to simplify word generations. According to the authors, their proposed method can improve the quality of the translation of Neural Machine Translation systems. The authors have concluded that their method can outperform baselines of state-of-art Japanese-English and Chinese-English translations.

Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratn Shah, and Ponnurangam Kumaraguru [11] in their research paper have proposed a novel Neural Machine Translation technique that uses Byte-Pair-Coding along with using word embedding. The proposed NMT technique has been applied to the English-Tamil pair language. They have shown that this technique performed better than complex techniques specifically for the Indian languages. The main aim to propose this technique was to overcome the Out-of-Vocabulary (OOV) problem for the languages whose translations are not much available online. The authors have concluded that their proposed MIDAS translator was able to outperform Google Translate with a BLUE score margin of 4.58.

The translation is an open vocabulary problem, but neural machine translation usually works with fixed vocabulary. Some previous works address the out-of-vocabulary translation problem by using backing off to a dictionary. Rico Sennrich, Barry Haddow, and Alexandra Birch [12] proposed a simpler and more effective approach to resolving out-of-vocabulary problems, by encoding rare and unknown words as a sequence of subword units and the author did this without using a back-off model for rare words. The authors used encoding (rare) words via subword units and byte pair encoding (BPE) for word segmentation, to perform open-vocabulary neural machine translation. Their research concluded that subword models achieve better accuracy than large-vocabulary models and back-off dictionaries for the translation of rare words. The author also mentioned that their model can generate new words which were not present in training time.

Nadeem Jadoon Khan, Waqas Anwar, and Nadir Durrani [13] in their research paper used Phrase-based Statistical Machine Translation (SMT) to analyze the performance of multiple Indian Languages. The author mentioned the performance of Indian languages (Bengali, Gujarati, Hindi, Malayalam, Punjabi, Tamil, Telugu, and Urdu) to the English language with an average accuracy of 10% on baseline system translation. The language used by the author in their research paper has sparse resources; due to that, they carry out a low BLEU score with a mean of 0.12. The author also mentioned the different BLEU scores for different language pairs. They concluded the paper by discussing the future aspects of Statistical Machine Translation (SMT) by using discrete approaches to develop quality language models.

K Hans and Milton R S [14] in their paper have compared and discussed five different models for English to Tamil language translation pairs. The models are Statistical Machine Translation (SMT), Phrase-Based SMT, RNNSearch, RNNSearch with Word2Vec, and RNNMorph. It was observed that the performance of the Phrase-Based SMT model was inferior to the RNNSearch Model in terms of BLEU score, when the RNNSearch model is been used with word2vec vectors there has been a slight increment in the BLEU score as compared to the BLEU score of RNNSearch, use of morphological segmentation enhances the performance of RNNMorph neural machine translation by 7.05 BLEU points on top of RNNSearch Model. They concluded their paper by discussing future aspects to carry out an end-to-end translation methodology for morphologically rich languages.

Raj Nath Patel, Prakash B. Pimpale and Sasikumar M [15] presented an English to Indian language machine translation that poses the challenge of structural and morphological divergence. We used pre-ordering and suffix separation for translation. Pre-ordering or reordering transforms the source sentence into a target-like order using the syntactic parse tree of the source text. One of the main issues in this translation was English uses the Subject-Verb-Object (SVO) order and most the Indian languages, including the ones under study, primarily use Subject-Object-Verb (SOV). Out of all the models, Factored SMT with suffix separation and reordering performs better. Transliteration as postprocessing further helps to improve the translation quality. However, there were problems while translating between English to Malayalam and Punjabi.

Roee Aharoni, Melvin Johnson, and Orhan Firat [16] presented research on Multilingual neural machine translation (NMT) that enables the training of a single model that supports translation from multiple source languages into multiple target languages. However, it was shown in a somewhat extreme case with more than 100 languages trained jointly, where we saw that in some cases the joint training may harm the performance of some language pairs (i.e. German-English above).

Rico Sennrich and Biao Zhang [17] demonstrated the performance of neural machine translation (NMT) that drops starkly in low-resource conditions, underperforming phrase-based statistical machine translation (PBSMT) and it requires large amounts of auxiliary data to achieve competitive results. Results show that low-resource NMT is very sensitive to hyperparameters such as BPE vocabulary size, word dropout, and others, and by following a set of best practices, we can train competitive NMT systems without relying on auxiliary resources.

Minh-Thang Luong, Hieu Pham Christopher, and D. Manning [18] in their research have implemented neural machine translation with two attention types: local and global. The models developed by the authors are trained on the WMT'14 training data consisting of 4.5M sentence pairs (116M English words, 110M German words). Their local attention approach yielded a gain of 5.0 BLEU over non-attentional models. Their English-German approach using the attention model has achieved state-of-the-art results for both WMT'14 and WMT'15 and outperformed existing models by more than 1 BLEU.

Lucia Benková and Lubomir Benko[19] in their paper have discussed the two most common approaches to neural machine translation; statistical machine translation(SMT) and neural machine translation(NMT). The author discussed the advantages and disadvantages of both approaches and concluded that NMT provides better translation as compared to SMT. They have also mentioned that SMT fulfills certain shortcomings of NMT.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan [20] in their research have implemented SMT AND NMT on

the augmented parallel corpora of two languages: English-Hindi and English-Tamil. Initially, they have extracted parallel corpora from Wikipedia pages using Siamese BiRNN encoder using GRU as the activation function. The models implemented yielded a percentage increase in BLEU scores of 11.03% and 14.7% for en–ta and en–hi pairs respectively, due to the use of parallel sentence pairs extracted from comparable corpora using the neural architecture.

III. Literature Review

A. *Long Short-term Memory(LSTM)*

Sepp Hochreiter and Jürgen Schmidhuber [21] proposed Long Short-Term Memory, a novel recurrent network architecture with an efficient gradient-based algorithm. This network architecture was developed to mitigate gradient-based problems such as vanishing and exploding gradients in recurrent neural networks(RNN). This kind of instability is the result of successive multiplication with the recurrent weight matrix at different time stamps.

LSTM is a type of RNN (Recurrent Neural Network) that has been specifically developed to resolve sequential prediction problems. LSTM is an advanced variant of RNN (Recurrent Neural Network) and a sequential network that preserves the information. We used RNN while dealing with short-term dependencies, but when it comes to remembering things for a longer duration of time RNN fails, the reason behind this is the problem of vanishing gradient. In [21] the author analyzed the vanishing gradient specifically. Whenever the gradient of the error function of the neural network is propagated back via a unit of a neural network, it acquires a specific factor that is either significantly more than one or less than one in a majority of the cases. Thus the gradient either signifies the following weight adaptation step or almost gets lost. As a result, the gradient blows up or decays exponentially over time in a recurrent neural network. LSTM is capable of grasping the vanishing gradient problem faced by RNN. The LSTM with gate cells is simplified as a differentiable type of computer memory [22]. That being so, LSTM units also stand for LSTM memory cells occasionally [23] for their ability to solve the problem of vanishing gradient with a small added complexity.

The LSTM is an enhancement of RNN wherein the recurrence conditions are changed as to how the hidden states $\bar{h}_t^{(k)}$ are propagated. The cell state $\bar{c}_t^{(k)}$ can be considered as a long-term memory that retains a part of the information in earlier states using a combination of partial "forgetting" and "increment" operations on the previous cell states. The advantage of this approach is that the network can model long-range dependencies in a sequence extended over a large number of tokens. The updation of these cell states over time creates greater persistence in information storage. This persistence mitigates the problem of exploding and vanishing gradients.[24]

Each cell in LSTM are computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \quad (1)$$

$$f_t = \sigma(W_f \cdot X + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot X + b_i) \quad (3)$$

$$o_t = \sigma(W_o \cdot X + b_o) \quad (4)$$

$$c_t = f_t \odot c_{t-1} + i_t \tanh \odot (W_c \cdot X + b_c) \quad (5)$$

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

The vectors f_t, i_t and o_t are referred to as forget, input, and output gates. These gates are used as Boolean gates for deciding whether to forget a cell state or whether to add to a cell state or whether to allow leakage into a hidden state from a cell state. The input and forget gates regulate the amount of change to be made to the previous cell state to retain long-term memory. LSTMs operate with a series of 'gates' that oversee how the information in a sequence of data comes into, is stored in, and leaves the network. Each gate carries out a discrete functionality. The cell states can be viewed as continuously updated long-term memory wherein the forget bits decide whether to reset the cell states from the previous time-stamp and forget the past and input bits decide whether to increment the cell states from the previous time-stamp to incorporate new information into the long-term memory from the current word.[24]. The output gate decides the selection of useful information from previous time steps onto the next time steps depending on its value. The forget, input, and output gates are shown in (2), (3) and (4) respectively. Equation 5 shows the process of selectively forgetting and adding to the long-term memory of the cell-states which represents the cell-state vector. Equation 6 shows the selective leaking of long-term memory to hidden states. h_t is the hidden state vector [25]. σ is the sigmoid function and $\odot$ represents element-wise multiplication.

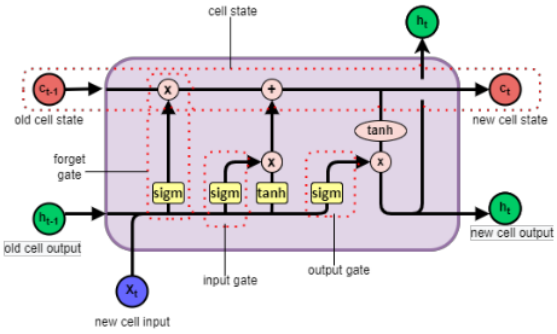

Fig. 1. An LSTM cell structure

TABLE I. COMMONLY USED SCORING FUNCTIONS

B. *Attention Mechanism*

The problem primarily faced by the previously mentioned encoder-decoder with LSTM networks is that they often produce poor translation for longer sentences. This could be attributed to the fixed-length encoding of source sentences. A fixed-length encoding of source sentences does not possess enough capacity to encode a sentence with a complex structure and complicated meaning [5]. Also,

sentences with varying lengths convey varying amounts of information.

An approach to mitigate this problem is the attention mechanism. Attention is a milestone approach in NMT architecture first introduced by [8]. The main task of the concept of attention mechanism is to avoid fixed vector representation of source sentences. The approach no longer encodes and uses a context vector representation $c(x)$of the entire source sentence. In contrast to that, the attentional decoder places its ability only on certain parts of the source sentences which are useful for generating the next tokens. This process results in the replacement of a constant context vector $c(x)$with a series of context vectors $c_j(x)$; one for each time step j. j is referred to as the 'time step' due to the sequential structure of autoregressive models used for the left-to-right order of NMT decoding. More generally it specifies a position in the target sentence.

Name	Scoring Function	Citation
Additive	$score(Q,K)_{p,q} = v^T \tanh(WQ_{p,:} + UK_{q,:})$	[5]
Dot-Product	$score(Q,K) = QK^T$	[18]
Scaled dot-product	$score(Q,K) = QK^T d^{-0.5}$	[4]

The fundamental definition of attention as described by Vaswani et al. [26] is the mapping of n query vectors to n output vectors via a mapping table of m key-value pairs. The attention network computes the relevance of each value vector based on a query and key vectors. Given a set of m query vectors $\boldsymbol{Q} \in \mathbb{R}^{m \times d}$, a set of n key vectors, $\boldsymbol{K} \in \mathbb{R}^{n \times d}$and associated value vectors$\boldsymbol{V} \in \mathbb{R}^{n \times d}$, the computation of the attention network finally involves the computation of the weighted sum of the value vectors for each query vector. The weights are determined by a similarity score between the query vectors and the key vectors. The attention networks can be roughly classified based on the scoring function: additive attention and dot-product attention. In practice, the dot-product attention is much faster compared to additive attention. However, increasing dimensionality d of the attention layer results in less stability of dot-product scoring than additive scoring function. [26] showed that the dot-product increases in magnitude as d increases, which may result in extremely small gradients when the softmax function is applied. To mitigate this issue, the dot-product is scaled by$\frac{1}{\sqrt{d}}$. The commonly used scoring function used is presented in table 1.

The output of the score(Q, K) is a $n \times m$ matrix of similarity scores. The softmax function normalizes over the columns of that matrix to sum the weights for each query vector to one. The attention mechanism is represented in Eq. 7.

$$Attention(K,V,Q) = Softmax\big(score(Q,K)\big)\,V \qquad (7)$$

A common way of using the attention network in NMT is at the interface of the encoder and decoder network. The hidden decoder statess_j are used as query vectors. Both the key and value vectors are derived from the hidden statesh_i of the recursive encoder. Simplifying the idea, $Q = s_i$ are the query vectors, n=J is the length of the target sentence, $K = V = h_i$ are the key and value vectors and m=I is the length of the source sentence. The output of the attention layer is used as a time-dependent context vector $c_j(x)$rather than using a fixed-length context vector $c(x)$ of the encoded source sentence. At each time step j source word representations are queried from memory where represented source words are stored. An attention matrix$\boldsymbol{A} \in \mathbb{R}^{J \times I}$ captures cross-lingual word relationships in the case of NMT.

Self-attention network(SAN)[26] is a type of attention network widely used in both encoder and decoder networks of NMT. It is a special case of attention mechanism wherein the keys, queries, and values are obtained from the same sentence through linear mapping of the input representations. The major disadvantage of SAN is that it ignores the word order in a sequence. Scaled-dot product scoring is used in SAN as described in (8).

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \qquad (8)$$

An important generalization of attention is multi-head attention [26]. The fundamental idea is to perform the H attention operation where H is several heads instead of one attention operation. The value of H is usually 8. The key, query, and value vectors for the attention head are linear transforms of Q, K, and V. The output of the multi-head attention network is the concatenation of the outputs of individual attention heads. To avoid increasing the number of parameters, the dimensionality of the attention heads is usually divided by H. Multihead attention is described by (9).

$$MultiHeadAttention(K,V,Q) = Concat(head_1,\cdots,head_H)W^O \qquad (9)$$

With weight matrix $W^O \in \mathbb{R}^{d \times d}$where

$$head_h = Attention(KW_h^K, VW_h^V, QW_h^Q) \qquad (10)$$

with weight matrices $W_h^K, W_h^V, W_h^Q \in \mathbb{R}^{d \times \frac{d}{H}}$ $for\ h \in H$].

Attention mechanisms have become an integral part of compelling sequence modeling, especially in NMT allowing modeling of dependencies without regard to their distance in the source or target sequences. [8][27] The use of an attention mechanism in NMT has significantly improved the translation performance.

C. *Encoder-Decoder Architecture*

The basic NMT network consists of an encoder and a decoder. A Recurrent Neural Network (RNN)[21] is a common choice for encoders and decoders for is the ability to map sequences ahead of time[28]. But due to the problem of long-term dependency with RNNs, LSTM is predominantly used. The primary concept of encoder-decoder was first presented by Nal Kalchbrenner and P. Blunsom[29], which used the fixed-length representation of the source sequence to generate the target sentence. Gated Recurrent Units(GRU) [30] are also used for alleviating the problem of exploding ad vanishing gradients. Using stacked LSTMs in deep architectures is used by [28]. The LSTM in the encoder part the encoder-decoder network computed this fixed-length representation of the source sentence and the decoder used this representation to generate the output sentence. Let $X = x_1^I$be the source sentences and$Y = y_1^J$ be the target sentences. The LSTM in the decoder network

generates the output sentence at each time-steps using a conditional probability distribution as shown in (11).

$$P(y|x) = \prod_{j=1}^{J} P(y_j|y_1^{j-1}, x) \quad (11)$$

The encoder LSTM represents the input sequence as a fixed-length vector c(x) and using the chain rule, the next word in the output sequence is predicted using the source sentence vectors and the predicted till the last time-step. This operation is defined in (12).

$$P(y_j|y_1^{j-1}, x) = g(y_j|s_j, y_{j-1}, c(x)) \quad (12)$$

Where s_j is the state of the LSTM decoder network. g(.) is a non-linear multi-layered function that takes the decoder state s_j as input along with the previous target token embedding y_{j-1} and computes the output with softmax over all the words in the vocabulary. Along with this, g(.) can also input the intermediate source sentence encoding c(x) as input for conditioning the source sentence. [29, 30]. The decoder states s_1 are initialized by c(x) [8, 28]. After the source sentences are encoded, the target sentences are generated the first sentence y_1 which is then fed back into the LSTM decoder to produce the second work y_2. The prediction of the target sentence is terminated only when end-of-sentence </s> token is produced.

Due to the problem of long-term dependencies of LSTM also, an attention mechanism is used to mitigate this issue. The primary idea of attention mechanisms is encoding each word in the input sentence into a vector instead of each sentence into a single vector [8] and referencing these vectors while decoding. The attention mechanism is broadly classified into two categories, global and local [18].

Each word in the input sentence is encoded bypassing it in an LSTM. The output produced by the LSTM is stored in a hidden matrix H consisting of h_j hidden vectors. The matrix is of dimension $d \times j$ where d is the size of the hidden layers and j is the length of the input sentence. Every word of the input sentence is represented by each column of the matrix. This hidden matrix consists of a variable number of columns but a fixed dimension is accepted by the decoder to obtain the context. This implies that the context vector must be of a fixed length. To get a fixed-length context vector, the matrix is multiplied by the attention vector as shown in (13).

$$c_t = H\alpha_t \quad (13)$$

Where H is the matrix, c_t is the context vector, and α_t is the attention vector. The crucial idea of the attention vector is to assert "importance" to a particular input word at a particular time step. A larger value α_t will have more impact on a word while predicting the next word in a sentence. Attention scores are calculated before the calculation of the attention vector [9]. The attention scores are calculated using a function taking two vectors as input and outputs a score between 0 and 1 as an indication of the importance of this specific encoding h_j at the time step h_t. The actual attention vector is obtained by normalizing using softmax over the scores as shown in (14).

$$\alpha_t = softmax(\alpha_t) \quad (14)$$

The context vector c_t is generated by weighing this attention vector with encoded representation H for the current time-step. The attention scores are calculated using (15), where h_s are the hidden source states and h_t is the target state as presented in (15) and (16).

$$\alpha_t(s) = \frac{\exp(score(h_t, \bar{h}_s))}{s'\exp(score(h_t, \bar{h}_s))} \quad (15)$$

$$score(h_t, \bar{h}_s) = h_t^T W_a \bar{h}_s \quad (16)$$

While decoding, these context vectors are used. These vectors focus selectively on certain words in the input sentence and thus provide a better translation. It is for the decoder to decide which part of the source sentence to pay attention to.

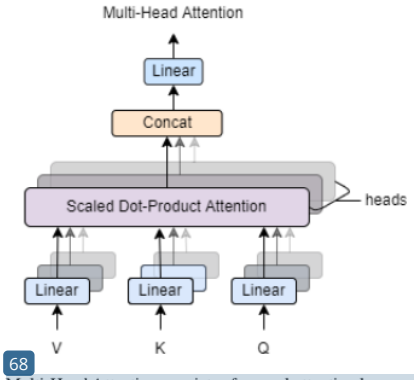

Fig. 2. Multi-Head Attention consists of several attention layers running in parallel.

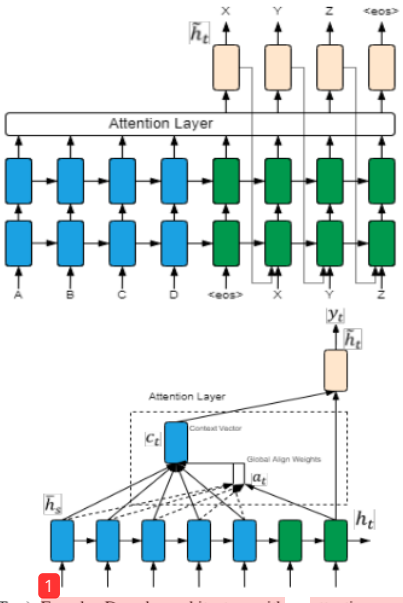

Fig.3.(Top) .Encoder-Decoder architecture with an attention mechanism. (Bottom) The global attentional model is implemented in this paper.

D. *Transformer Networks*

The transformer model is a state-of-the-art network architecture that completely relies on the attention mechanism instead of recurrence for application in NMT. It entirely relies on attention to retrieve global dependencies between input and output sequences. Transformers were first introduced by [26]. It is difficult for RNNs to deal with long-

range dependencies. LSTMs also face the difficulty to learn dependencies from distant positions as the number of operations required for relaying signals from two random inputs and output points increases in the distance between points [31]. Another problem faced by RNNs and LSTMs is the dependency of each hidden state on its previous state makes it difficult to parallelize making it inefficient on GPUs. Transformers network relies entirely on self-attention for computation of representations of input and output sequences instead of using sequence-aligned recurrent neural networks(RNNs). This network allows a significant level of parallelization to be trained on GPU.

In an encoder-decoder architecture, the encoder maps a sequence $x = (x_1, \cdots, x_n)$ to an intermediate continuous representation $z = (z_1, \cdots, z_n)$. The decoder generated an output sequence $(y_1, \cdots, y_m)$ for each element at a one-time step. The transformer follows the overall structure of a traditional encoder-decoder with stacked self-attention and point-wise fully connected layers for both encoder and decoder as proposed by [26]. The transformer architecture is presented in fig. 4.

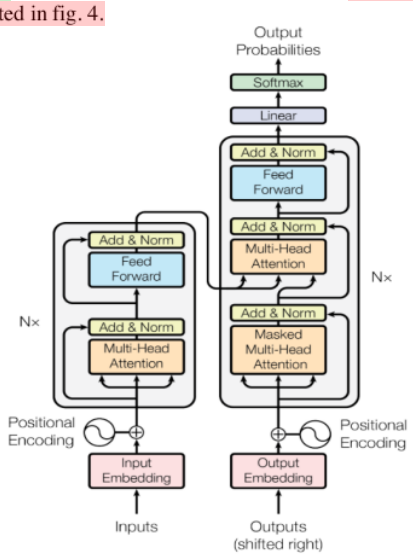

Fig 4. The transformer network architecture[4].

Encoder-Decoder Stacks: The transformer encoder is a stack of 6 homogenous layers. Each layer has 2 parts. The first is a multi-head self-attention mechanism and the second is a simple fully-connected feed-forward network. The authors [21] have employed a residual connection [32] around each of the sublayers, followed by a layer normalization[33]. The encoder creates an intermediate representation using each word in the input sentence. An attention scores is generated based on the comparison of the intermediate representation of a word and all the other words in the sentence. These attention scores are then fed to a fully-connected network as weights which generates new representations for the keyword. This process is carried on for all the words and the representation is then passed on to the decoder so that it has all the dependencies needed for predictions.

The decoder part of the transformer also has a stack of N=6 identical layers. Each layer of the decoder is composed of 3 parts. Two are similar to that of the encoder part. In a third party, a masked multi-head self-attention mechanism is added. The decoder has access to all hidden states of the decoder that are used to predict the next words at each time step. Like the encoder, the decoder also employs residual connections followed by layer normalization(also known as batch normalization). The important aspect of the decoder is the masking of the multi-head self-attention mechanism. This modification is done to prevent posterior information from the decoder. This ensures that the predictions for the position I can depend only on the known outputs at a position less than i. Without masking the subsequent positions from the decoder stacks, the model will not be able to learn anything and will only repeat the target sentence.

b) Application of Multi-head Self-attention: The mapping of a query vector and a set of key-value vector pairs to output is defined as an attention function. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The transformer networks proposed by [26] implemented scaled-dot product attention. In practice, the input consists of key and query matrices denoted by K and Q and value matrix V. The key, query, and value vectors are packed in matrices k, Q, and V respectively. The matrix output is computed using (8).

A scaling factor $\frac{1}{\sqrt{d}}$ is implemented in the attention to counteract the effect of large values d_k, which increases the magnitude of dot-products which in turn pushes the softmax function to regions of extremely small gradients [28]

Multi-head attention has its benefit to project the queries, keys, and values linearly h times with different learned linear projections to $d_k, d_q d_v$ and, dimensions. The attention function is performed in parallel to these projected versions of key, query, and values to the yield d_v-dimensional output vector. These are once again concatenated and projected. The jointly attending of information from different representation subspaces at different positions by the model facilitated by multi-head attention which was inhibited by averaging of single attention heads.

The transformer uses multi-head attention in three distinct ways:

- This implementation is similar to[8, 33]. Implementing an attention layer between encoder-decoder structure allows every position in the decoder the knowledge of all the positions of the input sequence. This is achieved by taking the queries from the previous decoder layer and key and value vectors from the output of the encoder.
- Attention is contained within the encoder. In the self-attention layer, the query, key, and value vectors are obtained from the output of the encoder's previous layers. Each point in the encoder attends to all the positions in the encoder's previous layer. [26]
- An identical approach to attention in the encoder is implemented in the decoder with a small tweak in the attention mechanism. Masking of attention is used to prevent the leftward flow of information in

the decoder to preserve the auto-regressive property of the model.

c) Residual Connections: The idea behind the residual connection is to easily optimize the network[35]. It is responsible to preserve the information before each operation. This enables faster learning of parameters in the backpropagation phase of training.

d) Feed Forward networks: Along with the attention mechanism, each layer of the encoder and decoder consists of a fully connected feed-forward network separately and identically applied at each position. A linear transformation of the ReLU activation function is implemented in between.

e) Embedding and Softmax: Embeddings are used to convert input and output tokens into vectors of dimension d_{model}. The embedding layer output is obtained by multiplying the weight matrix $\sqrt{d_{model}}$. The softmax activation function in the output decoder layer is used to convert decoder output into predicted next-token probabilities.

f) Positional Encodings: Additional information is necessary to make use of the order of sequence. This information must be about the relative or absolute position of sentence tokens in the sequence which will retain the position information of tokens that is significant for the next steps. Positional Encodings are added to the input embeddings at the bottom of encoder and decoder stacks. These encodings have the same dimensions the embeddings, d_{model} The authors in [26] have used sine and cosine functions of various frequencies. It is shown in (17) and (18).

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \quad (17)$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \quad (18)$$

Here pos and are the position and dimension respectively. Each dimension of the positional encoding corresponds to a sinusoid. The wavelength forms a geometric progression from $2\pi 10000 \cdot 2\pi$ to. According to the authors' hypothesis, it would allow the model to easily learn to attend to relative positions since any fixed offset k, $PE_{pos} + k$ can be represented as a linear function PE_{pos}.

E. Why self-attention

The use of self-attention in the transformer as specified by[26] is due to the computational complexity per layer. Another reason is the amount of parallelizable computation which is measured by the number of required sequential operations. Another reason for self-attention is the long-range dependencies path length in the network. Learning long-range dependencies is a major challenge in traditional models for many sequence transduction tasks. One prime factor hindering the ability to learn these dependencies is the path length to be traversed by forwarding and backward signals in the network. Short path length makes it easier to learn long-range dependencies[31].

In terms of computational complexity, self-attention layers are faster as compared to recurrent layers when the sequence length is n and is smaller than the representation dimensionality d. For a maximum path length complexity $O(1)$, the complexity of sequential operation is $O(1)$ with $O(n^2 \cdot d)$ complexity per layer. As a result, self-attention can yield more interpretable models.

F. Bilingual Evaluation Understudy (BLEU)

Bilingual evaluation understudy(BLEU) is an important translation evaluation metric first proposed by Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu [36]. It is an NLP metric that has been developed to overcome the limitations of existing metrics. The range of bleu score is generally from 0 to 1. A score of 1 is very unlikely because the translations may differ in terms of choice of words or the order in which they are used.

According to the authors[36], the cornerstone of this metric is the precision measure. To compute precision, one simply counts up the number of candidate translation words (unigrams) that occur in any reference translation and then divides by the total number of words in the candidate translation.

BLEU considers multiple reference translations, each of which may use a different word choice to translate the same source word.[36] Hence, the foremost task of the Bleu score to compare the n-grams of the predicted and the actual translation and count the number of matches. These matches do not depend on the individual position of the words.

To calculate the Bleu score, Clipped precision used. Now, n-gram matches are computed sentence by sentence. The clipped n-gram counts are then added and divided by the number of n-grams in the test corpus. The result obtained is known as the modified n-gram precision score p_n, using n-grams up to length N and positive weights w_n summing to one. Here $w_n = 1/N$.

After calculating these modified precision scores, they are combined and the resultant score is known as Geometric Average Precision which is defined by (19).

$$Geometric\ Average\ Precision(N) = \exp(\sum_{n=1}^{N} w_n log\ p_n) = \prod_{n=1}^{N} p_n^{w_n} \quad (19)$$

Modified n-gram precision penalizes the predicted sentences that are greater in length than the target sentences. To penalize the sentences that are too short, Brevity Penalty is used. Brevity Penalty is defined by (20).

$$Brevity\ Penalty = \begin{cases} 1, & if\ c > r \\ e^{(1-r/c)}, & if\ c \leq r \end{cases} \quad (20)$$

where c is the predicted sentence length and r is the target sentence length.

This ensures that the brevity penalty cannot have a value greater than 1, even when the predicted sentence is greater than the target sentence. If it predicts few words, then this value will be small.

To final calculate the Bleu score, the brevity penalty and the Geometric Average Precision is multiplied as shown in (21).

$$Bleu(N) = Brevity\ Penalty * Geometric\ Average\ Precision(N) \quad (21)$$

The ranking property is more immediately apparent in the log domain. This formula can also be written as shown in (21).

$$log\ BLEU\ = \min\left(1-\frac{r}{c},0\right)+\sum_{n=1}^{N} w_n log\ p_n \quad (21)$$

G. *Word Error Rate*

Word Error Rate (WER) is a common metric used to measure the performance of speech recognition[39] as well as MT. WER is calculated by dividing the total words by the number of errors. To find out the number of different words between predicted output and reference transcript, WER compares the predicted output with reference transcript word by word. A lower WER corresponds to better translations.

H. *Meteor*

METEOR is another automatic metric used for machine translation evaluation. It is based on the generalized concept harmonic mean of unigram matching between predicted output and reference transcript. METEOR uses explicit word-to-word matching between the translation and a given reference translation to compute a score and then uses that score to evaluate a translation[40]. A higher METEOR score corresponds to better translations.

IV. METHODOLOGY

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll-down window on the left of the MS Word Formatting toolbar.

A. *System SetUp*

The implementation of the encoder-decoder model and transformer network with specified configurations described in the previous section is done using Keras Backend[48] The training of the encoder-decoder model is carried out on a quad-core CPU with 8 GB RAM. The transformer network is trained on NVIDIA Tesla K80 GPU.

Algorithm 1 Implementation of Seq2Seq NMT Networks

Input: Source sentences (X) and ground-truth target sentences (Y)

Output: Predicted target sentences ($\hat{Y}$)

1: Consider T={English, German, Spanish, French, Bengali, Hindi}
2: **for** each X and Y pairs in T **do**
3: Convert to lowercase
4: Eliminate punctuations
5: Obtain vectorized X and Y sequences
6: **end for**
7: Obtain batches dataset
8: Train transformer network and LSTM- encoder-decoder with attention
9: Generate the predicted sentences from the trained models
10: Evaluate the predicted sentences with ground-truth sentences using BLEU, WER, and METEOR score
11: **return** $\hat{Y}$

B. *The Dataset*

The mentioned encoder-decoder network and transformer network are implemented in five languages. This includes three commonly spoken languages worldwide, i.e. German, Spanish and French, and the two most common Indian languages, Hindi and Bengali. About 43.63% of the population and 8.03% of the population in India speaks Hindi and Bengali respectively. The data used in this project is obtained from the Tatoeba project. [46,47]. The language pairs in each bilingual corpus used here consist of translations between English and the above-mentioned languages. The sentence pairs in each of the language corpus are tab-separated. The details of sentence pairs of $English \rightarrow X, X \in \{German, Spanish, French, Hindi, Bengali\}$ for are presented in table II.

TABLE II. NUMBER OF SENTENCE PAIRS AND VOCABULARY SIZE OF THE ORIGINAL DATASET

$English \rightarrow X$	Number of sentence pairs in the corpus	Vocabulary
German	249230	38,407
Spanish	138437	28,338
French	192341	35,624
Bengali	4617	3,393
Hindi	2934	3,052

The vocabulary size of the used corpus is 21,363. The number of training, tuning testing data taken for $Eng \rightarrow Deu$, $Eng \rightarrow Spa$ and $Eng \rightarrow Fra$ are 85,000, 14,000, and 1,000 respectively, and for $Eng \rightarrow Ben$ $Eng \rightarrow Hin$ and the number of sentence pairs are belonging to training, tuning and testing pairs are in the ratio of about 70%, 20%,10% respectively for implementation on transformer network. Whereas only 20,000 sentence pairs with 18,000 training airs and 2,000 test pairs are taken for $Eng \rightarrow Deu$ $Eng \rightarrow Spa$, $Eng \rightarrow Fra$ and 4,617 $Eng \rightarrow Ben$ and 2,934 $Eng \rightarrow Hin$ with 80%-20% partition for training and test pairs are created for encoder-decoder model implementation.

A typical neural machine translation model depends on the vector representation of the words. This project aims to develop a multilingual translation application. To this end, the vectors of English words are created from the combined data of the above five mentioned corpora. The word vectors of the other five languages are created from individual datasets. The vectors are generated according to the translation of English to X or from X to English and contain <unk>, <sos> and <eos> tokens accordingly. The data was encoded in UTF-8 format.

C. *LSTM- Encoder-Decoder Model*

The encoder-decoder model is implemented with 4 LSTM layers in the encoder and 1 LSTM in the decoder with 500 cells in each layer and 500-dimensional word

embeddings. The attention mechanism[8] is implemented with residual connections. A time-distributed dense layer is added to bridge the encoder and decoder networks. The English and German vocabulary size i$Eng \rightarrow Deu$s 3,400 and 5,413 respectively with an input sequence length of 12 and output sequence length of 5. The English and Spanish vocabulary size i$Eng \rightarrow Spa$ s 3,566 and 7,428 respectively with an input sequence length of 12 and output sequence length of 6. The English and French vocabulary size$Eng \rightarrow Fra$s are 3,321 and 6,930 respectively. The English and Bengali vocabulary size i$Eng \rightarrow Ben$ s 1,801 and 3,209 respectively. The English and Hindi vocabulary size$Eng \rightarrow Hin$ is 2,268 and 2,894 respectively. The result is obtained by using softmax over all these target vocabulary for each translation.

The model is implemented using Keras with TensorFlow[37] as a backend for the above encoder-decoder model implementation. Adam[38] optimization function is used for convergence of the model during training. The models are trained for 50 epochs with a batch size of 512 for$Eng \rightarrow Deu$, $Eng \rightarrow Spa$ and$Eng \rightarrow Fra$ translations and for 100 epochs for$Eng \rightarrow HinEng \rightarrow Ben$ translation. The default learning rate of 0.001 is reduced by 0.5 after the subsequent 2 epochs. Evaluation of the model for all these translations is presented in later sections.

D. *The Transformer Model*

The transformer network is implemented with the specifications mentioned in the previous sections. This implementation is similar to what had been presented by the authors in[26] with certain tweaks in the hyperparameters. The transformer network has a 256-dimensional embedding vector as well as an input and output vector. The maximum vocabulary size of 100,000 is taken for the encoder and decoder parts. The vocabulary for English is obtained by concatenating English sentences from five corpora. The number of encoder and decoder units in the implemented transformer network is 6 each. This number is specified by the authors of transformer networks[26]. The feed-forward networks in the transformer have a dimension of 1024. Multiheaded self-attention with 8 heads is used for the transformer network. The number of heads is kept to be 8 as presented in[26]. The key, query, and value vectors used in this network are 32 dimensional. A constant source and target sequence length of 20 is used for the network. Using the attention mechanism eliminates the need for fixed vector representation of the source sentences after the encoder units. The dataset used for training the network is cached into batches of size 128.

The transformer network is implemented for the translation of English to German, Spanish, French, Hindi, and Bengali and from these languages to English alike. This approach is adopted for evaluating the performance of the model as well as for the mentioned messaging application. The transformer network is implemented using Keras with TensorFlow as the backend. Adam optimizer with a default learning rate of 0.001 is used for minimizing the training loss. The model is trained for 25 epochs$Eng \rightarrow Deu$, $Eng \rightarrow Spa$ and $Eng \rightarrow Fra$ vice-versa. Similarly, the network is trained for 100 epochs$Eng \rightarrow HinEng \rightarrow Ben$ and vice-versa. The training time for the network is around 3 hours. The number of training, test, and tuning sentence pairs is mentioned in the previous section. The evaluation of the network for all the aforementioned translations is presented in further sections.

E. *Android Application*

This project aims to develop an application for messaging in different languages. To this end, we developed an android messaging application using the trained transformer networks. This application can be used by users for effective communication in their native language.

Django REST framework is used to develop the API. The API uses threads to translate an input sentence into multiple languages for faster execution. The API is developed with two endpoints: message translate endpoint and translate to all endpoints. The message translate endpoint accepts post requests and requires these parameters in its post body. Translate to all endpoints accept the input sentence English sentence and translate it into all the other languages implemented in the application. The message translate endpoint consists of a few parameters. They are:

- Message: The actual sentence input by the user/
- Position: The position of that message in the client app message. This is required so that when a response is given back to the app, the app will mark that message as "Sent".
- t_keys: This parameter contains a string like "eng-spa", which implies that the message is in the "English" language and it needs to be translated to "Spanish".
- Room_name: Unique room name from which this message is sent.
- sender_id - Unique id of sender
- message_id - Unique new id of the message.

The application is built using native Java. The messages and other parameters data are stored in a database using the Firebase Realtime Database. A feature of the application is user authentication before accessing the application. Our application consists of three fragments: Translate, Room, and Profile. The translate fragment is designed specifically for translation of any English sentence in other languages and vice-versa. The profile fragment lets the user edit their profile in the application. The most important fragment is the room fragment which contains the essence of this project. In the room fragment, users can create rooms that will have English as a primary language and can choose any other language as a secondary. This selection creates room for language-specific users. After the creation of a room, members of that room have to select their primary language from the two languages of the room for communication. There is ChatActivity in Room Fragment. Users will be shown messages in their selected language. This implementation did bridge the gap between different languages.

V. Results And Observations

The attentional encoder-decoder model and transformer network implemented and trained as shown in the previous section are evaluated using three commonly used translation metrics: BLEU score, WER, and METEOR score. Let X={Deu, Spa, Fra, Hin, Ben}. A comparison of$Eng \rightarrow X$ translations and vice-versa of the transformer network using

the mentioned metrics are presented in table III. A similar comparison of the attention encoder-decoder model for the five language translations is shown in table IV. The BLEU score, WER, and METEOR score presented in this section are calculated on the unseen test set. These scores are averaged over 1,000 test samples $Eng \rightarrow Deu$ $Eng \rightarrow Spa$, $Eng \rightarrow Fra$ and 100 test samples of $Eng \rightarrow Hin$ $Eng \rightarrow Ben$ and. All the values presented from this point forward are the average score of the mentioned samples.

For a comparative analysis, we have used only 3K samples from all five datasets for training and evaluation of the transformer model. The transformer model is trained for 100 epochs for all the translations.

It can be observed from Fig. 5 that the transformer network has better performance in terms of BLEU score, WER, and METEOR score for $Eng \rightarrow Hin$ and $Eng \rightarrow Ben$ as compared to translations into the other three languages. These results are obtained despite comparatively fewer training samples(Vocabulary sizes of only 3,393 and 3,052) as compared to samples of three languages with a vocabulary size of 38,407, 28,338, and 35,624 respectively. Despite training equal number of samples and for equal number of epochs, the $Eng \rightarrow Hin$ and $Eng \rightarrow Ben$ translations have

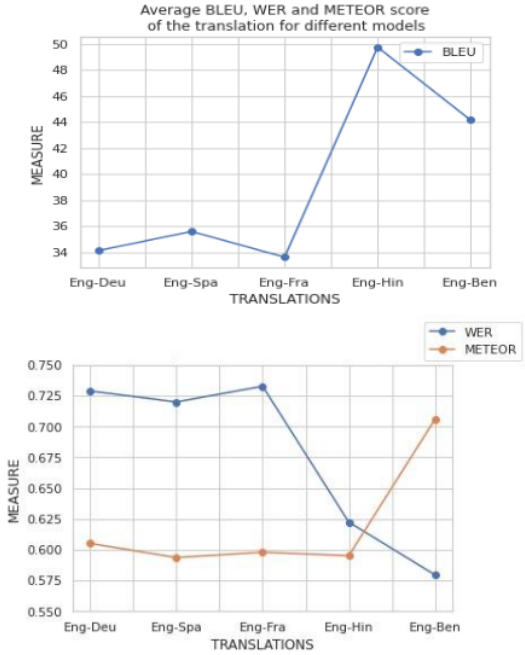

Fig. 5. (Top) Average BLEU score, (Bottom) WER and METEOR score of the transformer network evaluated on 3K sample pairs each after training for 100 epochs.

TABLE III. COMPARISON OF TRANSFORMER NETWORK USING BLEU, WER, AND METEOR SCORE FOR TRANSLATION OF ENGLISH TO 5 LANGUAGES AND VICE-VERSA.

Translations	BLEU	WER	METEOR Score
$Eng \rightarrow Deu$	39.295	0.55461	0.75166
$Eng \rightarrow Spa$	38.176	0.44208	0.79856
$Eng \rightarrow Fra$	36.342	0.55080	0.71370
$Eng \rightarrow Hin$	51.781	0.09868	0.93813
$Eng \rightarrow Ben$	56.285	0.15218	0.90877
$Deu \rightarrow Eng$	39.889	0.48389	0.76922
$Spa \rightarrow Eng$	39.481	0.45236	0.81766
$Fra \rightarrow Eng$	39.913	0.51848	0.74378
$Hin \rightarrow Eng$	36.939	0.36341	0.84731
$Ben \rightarrow Eng$	43.308	0.38865	0.85328

TABLE IV. COMPARISON OF ENCODER DECODER MODEL USING BLEU, WER FOR TRANSLATION OF ENGLISH TO 5 LANGUAGES.

Translations	BLEU	WER
$Eng \rightarrow Deu$	36.248	0.97127
$Eng \rightarrow Spa$	36.259	1.44166
$Eng \rightarrow Fra$	36.418	1.32036
$Eng \rightarrow Hin$	19.881	1.79101
$Eng \rightarrow Ben$	32.980	1.27520

better performance as compared to that of $Eng \rightarrow Deu$, $Eng \rightarrow Spa$ and $Eng \rightarrow Fra$. We suspect that this abnormality arises due to dissimilarity of base script of the Indian languages as compared to the other three languages.

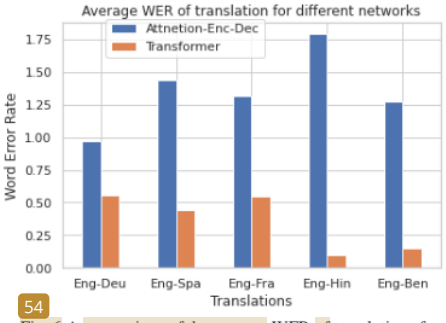

Fig. 6.A comparison of the average WER of translations for encoder-decoder model and transformer network

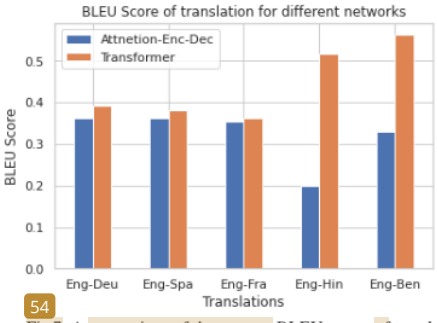

Fig.7. A comparison of the average BLEU scores of translations for encoder-decoder model and transformer network

It can also be observed that the BLEU score, METEOR score, and WER of $Eng \rightarrow Deu$, $Eng \rightarrow Spa$ and $Eng \rightarrow Fra$ is comparatively similar to that of the translations into English. Comparing the BLEU score and WER of the transformer network and attentional encoder-decoder, it can be observed that the transformer has a higher BLEU score for all the language-translation as compared to the attentional encoder-decoder model.

The transformer model a[illegible]has lower WER for the same translations as compared to the attentional encoder-decoder model. This is because the transformer network uses positional embedding to encode the input sequences wh[illegible] preserves the sentence order. Although both the models use an attention m[illegible]hanism to encode each position, the use of self-attention in the encoder-decoder model and multi-head [illegible]-attention mechanism in the encoder stack along with the masked multi-head attention mechanism in the decoder stack [illegible]bles the model to learn and generalize better. The comparison of the two models is presented in fig. 6.

The average BLEU score of the transformer network from all the generated translations of the five mentioned languages is presented in fig. 7. Fig. 7 shows the average 1-gram precision BLEU for sentence lengths ranging from 3 to 20 words. There is a gradual drop in the average BLEU score as the sentence length increases. This gradual decrease is better than a sharp drop in the BLEU score for encoder-decoder in [5]. This enhanced performance for longer sentences is attributed to the implementation of the attention mechanism in the encoder and decoder stacks of the transformer network and in between encoder and decoder in the attentional encoder-decoder

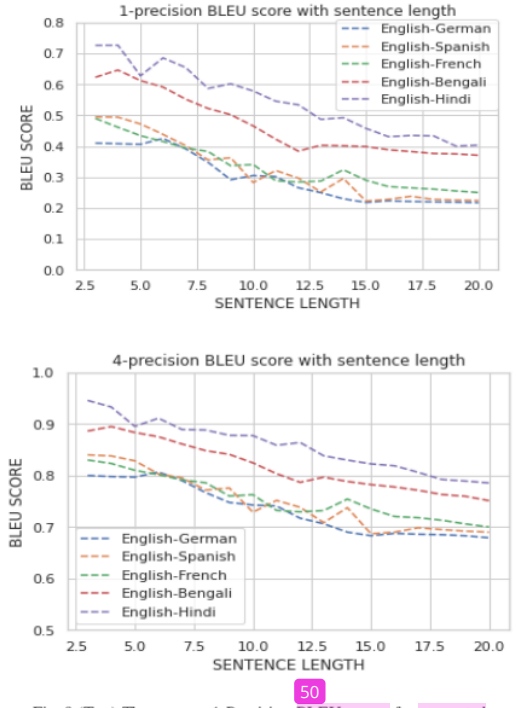

Fig. 8.(Top) The average 1-Precision BLEU scores for generated translations of the test set by the transformer n[illegible]ork for the length of the sentences. (Bottom) The average 4-Precision BLEU scores for generated translations of the test set by the transformer network for the length of the sentences.

TABLE V: A QUANTITATIVE ANALYSIS OF OUR BASELINE TRANSFORMER NETWORK AND LSTM ENCODER-DECODER MODEL WITH PREVIOUSLY DEVELOPED NMT MODELS

Method/System	*Eng → Deu*	*Eng → Fra*
Sutskever et al. (2014) [28]	-	36.5
Best WMT'14 result [41]	20.7	37.0
Jean et al. (2014) [42]	21.6	37.5
[illegible]et al. (2016) [43]	24.61	39.92
Luong et al. (2015) [18]	25.9	-
Vaswani et al. (2017) [4]	28.4	41.0
Chen et al. (2018) [45]	28.5	41.0
Shaw et al. (2018) [44]	29.2	41.5
4 LSTM+Att Enc-Dec model	36.24	36.25
Our baseline Transformer model	39.29	39.91

model. The attention mechanism accredits to the mitigation of long-term dependency problems in longer sentences which in tur[illegible]creased its average BLEU score.

The results achieved in this paper are comparable to previously implemented Seq2Seq models for NMT problems.

The BLEU is calculated using translated and corresponding reference sentences. BLEU scores of our baseline transformer model and encoder-decoder model have avera[illegible] over 1,000 test samples. A quantitative comparison of the average BL[illegible] of the baseline model in this paper is presented in table V for English to German and English to French translations. The number of parameters in our transformer mode is 84 million. The BLEU of the transformer for English to French translation is better than some previously implemented encoder-decoder models, however, the transformer network of [4] with 213 million parameters has a BLEU of +1.09. The transformer outperformed the LSTM-attentional encoder-decoder model by +3.66 BLEU. The baseline transformer model for English-German translation outperformed all previously developed systems. The network has a +3.05 BLEU than that of the LSTM-a[illegible]ntional encoder-decoder. This quantitative analysis shows that our transformer model is lightweight and has a better translation performance than most previously developed systems. One differencing factor is the different training samples used for the implementation of the systems. However, this constitutes a small factor for the varying BLEU of the models.

VI. CONCLUSION

In this project, we implemented two neural machine translation models for five languages and evaluated the performance [illegible] three automatic evaluation metrics. We implemented an attentional encoder-decoder model with LSTM and transformer network on translations from English to German, Spanish, French, Hindi, Bengali, and vice-versa. The translations are evaluated on BLEU, WER, and METEOR. We observed that the BLEU and METEOR score for the transformer network is quantitively more than the attentional encoder-decoder model for each language. Also, the WER is less for the transformer network as compared to the encoder-decoder model. We also compared the BLEU score with increasing sentence length for all the languages using the transformer network. It can be concluded that the BLEU for longer sentences is not decreasing as much as it would decrease for a model with no attention. Another unique observation that can be concluded from the results i[illegible]hat the BLEU of predicated translations is dependent on a trade-off between the number of training samples and several epochs the model is trained, but since the language corpus used here is small for some languages, their better-predicted translations are factored by the greater number of epochs.

The BLEU of both the models is compared with previously developed models by other researchers for *Eng →*

$DeuEng \rightarrow Fra$and. It can be observed that the BLEU of our transformer model is better than previously developed models with BLEU of 39.29 and 39.91 $Eng \rightarrow DeuEng \rightarrow Fra$ and respectively. A notable phenomenon is that our transformer model consists of only 84 million parameters which are less compared to previously developed models. Most importantly, we demonstrated that a better translation accuracy can be achieved by simple approaches even with fewer training epochs if the training samples are substantially better.

We have unleashed the potential of real-time speech-to-speech translation by developing an android application for multilingual messaging. This approach will help thousands of people around the globe who have inconvenience in some form regarding effective communication because of the language barrier. Going further, we can optimize our architecture. This could enable our models to run on embedded devices which will open a whole new possibility for effective communication. Moving further, we will add more common languages around the world and India to our application.

Adding other Indian languages to our application is important as a large diverse population in India requires an effective solution to the problem of the language barrier. The results shown in our paper can be used and referred to by research in the future. Our messaging application can be used to bridge the gap between different languages.

REFERENCES

[1] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. CoRR abs/1609.08144 (2016). http://arxiv.org/ abs/1609.08144

[2] Ghosh, Siddhartha & Thamke, Sujata & U R S, Kalyani. (2014). Translation Of Telugu-Marathi and Vice-Versa using Rule-Based Machine Translation. Computer Science & Information Technology. 4. 10.5121/csit.2014.4501.

[3] Tan, Zhixing & Wang, Shuo & Zonghan, Yang & Chen, Gang & Huang, Xuancheng & Sun, Maosong & Liu, Yang. (2020). Neural machine translation: A review of methods, resources, and tools. AI Open. 1. 5-21. 10.1016/j.aiopen.2020.11.001.

[4] Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. *ArXiv, abs/1706.03762*.

[5] Cho, Kyunghyun & Merrienboer, Bart & Bahdanau, Dzmitry & Bengio, Y.. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. 10.3115/v1/W14-4012.

[6] Stahlberg, F. (2019). Neural Machine Translation: A Review and Survey. *arXiv: Computation and Language*.

[7] Pathak, A., & Pakray, P. (2019). Neural Machine Translation for Indian Languages. *Journal of Intelligent Systems, 28*, 465 - 477.

[8] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR, abs/1409.0473*.

[9] Revanuru, K., Turlapaty, K., & Rao, S. (2017). Neural Machine Translation of Indian Languages. *Compute '17*.

[10] Wu, S., Zhang, D., Yang, N., Li, M., & Zhou, M. (2017). Sequence-to-Dependency Neural Machine Translation. ACL.

[11] Choudhary, H., Pathak, A.K., Saha, R.R., & Kumaraguru, P. (2018). Neural Machine Translation for English-Tamil. WMT.

[12] Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. ArXiv, abs/1508.07909.

[13] Khan, N.J., Anwar, W., & Durrani, N. (2017). Machine Translation Approaches and Survey for Indian Languages. ArXiv, abs/1701.04290.

[14] Krupakar, H., & Milton, R.S. (2016). Improving the Performance of Neural Machine Translation Involving Morphologically Rich Languages. ArXiv, abs/1612.02482.

[15] Patel, R.N., Pimpale, P.B., & Sasikumar, M. (2019). Machine Translation in Indian Languages: Challenges and Resolution. Journal of Intelligent Systems, 28, 437 - 445.

[16] Aharoni, R., Johnson, M., & Firat, O. (2019). Massively Multilingual Neural Machine Translation. ArXiv, abs/1903.00089.

[17] Sennrich, R., & Zhang, B. (2019). Revisiting Low-Resource Neural Machine Translation: A Case Study. ArXiv, abs/1905.11901.

[18] Luong, Minh-Thang & Pham, Hieu & Manning, Christopher. (2015). Effective Approaches to Attention-based Neural Machine Translation. 10.18653/v1/D15-1166.

[19] Benková, Lucia & Benko, Ľubomír. (2020). Neural Machine Translation as a Novel Approach to Machine Translation.

[20] Ramesh, Sree & Sankaranarayanan, Krishna. (2018). Neural Machine Translation for Low Resource Languages using Bilingual Lexicon Induced from Comparable Corpora. 112-119. 10.18653/v1/N18-4016.

[21] Hochreiter, S., Schmidhuber, J., "Long Short-Term Memory", Neural Computation 9 (8), 1997, pp. 1735–1780

[22] Graves, A., Schmidhuber, J., "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures", Neural Networks, Vol. 18, Issue 5–6, 2005, pp. 602–610

[23] Sundermeyer, Martin & Ney, Hermann & Schluter, Ralf. (2015). From Feedforward to Recurrent LSTM Neural Networks for Language Modeling. Audio, Speech, and Language Processing, IEEE/ACM Transactions on. 23. 517-529. 10.1109/TASLP.2015.2400218.

[24] Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Cham: Springer. ISBN: 978-3-319-94462-3

[25] Wang, Yequan & Huang, Minlie & Zhu, Xiaoyan & Zhao, Li. (2016). Attention-based LSTM for Aspect-level Sentiment Classification. 606-615. 10.18653/v1/D16-1058.

[26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[27] Kim, Y., Denton, C., Hoang, L., & Rush, A.M. (2017). Structured Attention Networks. *ArXiv, abs/1702.00887*.

[28] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks

[29] Kalchbrenner, Nal & Blunsom, P.. (2013). Recurrent continuous translation models. EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. 3. 1700-1709.

[30] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014b). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. URL: https://www.aclweb.org/anthology/D14-1179. doi:10.3115/v1/D14-1179.

[31] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016

[33] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[34] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

[35] He, Kaiming & Zhang, Xiangyu & Ren, Shaoqing & Sun, Jian. (2016). Deep Residual Learning for Image Recognition. 770-778. 10.1109/CVPR.2016.90.

[36] Papineni, Kishore & Roukos, Salim & Ward, Todd & Zhu, Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.

[37] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Lavie, Alon & Agarwal, Abhaya. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. 228-231.

[38] Nadir Durrani, Barry Haddow, Philipp Koehn, and Kenneth Heafield. Edinburgh's phrase-based machine translation systems for wmt-14. In WMT, 2014.

[39] Jean, Sébastien & Cho, Kyunghyun & Memisevic, Roland & Bengio, Y.. (2014). On Using Very Large Target Vocabulary for Neural Machine Translation. 1. 10.3115/v1/P15-1001.

[40] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J.R., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G.S., Hughes, M., & Dean, J. (2016). Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv, abs/1609.08144*.

[41] Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software was available from tensorflow.org.

[42] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[43] Park, Y., Patwardhan, S., Visweswariah, K., & Gates, S.C. (2008). An empirical analysis of word error rate and keyword error rate. *INTERSPEECH*.

[44] Shaw, Peter & Uszkoreit, Jakob & Vaswani, Ashish. (2018). Self-Attention with Relative Position Representations. 464-468. 10.18653/v1/N18-2074.

[45] Chen, Mia & Firat, Orhan & Bapna, Ankur & Johnson, Melvin & Macherey, Wolfgang & Foster, George & Jones, Llion & Schuster, Mike & Shazeer, Noam & Parmar, Niki & Vaswani, Ashish & Uszkoreit, Jakob & Kaiser, Lukasz & Chen, Zhifeng & Wu, Yonghui & Hughes, Macduff. (2018). The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation. 76-86. 10.18653/v1/P18-1008.

[46] Tiedemann, J. (2020). The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT. *WMT*.

[47] https://www.manythings.org/bilingual/

[48] François Chollet and others. 2015. Keras. https://github.com/fchollet/keras. (2015)

MOS11

ORIGINALITY REPORT

31%	25%	20%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.aclweb.org Internet Source	2%
2	Karthik Revanuru, Kaushik Turlapaty, Shrisha Rao. "Neural Machine Translation of Indian Languages", Proceedings of the 10th Annual ACM India Compute Conference on ZZZ - Compute '17, 2017 Publication	2%
3	www.repository.cam.ac.uk Internet Source	2%
4	www.statmt.org Internet Source	2%
5	deepai.org Internet Source	1%
6	www.arxiv-vanity.com Internet Source	1%
7	dokumen.pub Internet Source	1%

8	Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, Yang Liu. "Neural machine translation: A review of methods, resources, and tools", AI Open, 2020 Publication	1 %
9	arxiv.org Internet Source	1 %
10	Zhou Fang, Dingwen Wang, Xinrong Wang, Yan Li. "Research and Application of Big Earth Data Distribution and Sharing System", 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), 2021 Publication	1 %
11	aclanthology.org Internet Source	1 %
12	hdl.handle.net Internet Source	1 %
13	huggingface.co Internet Source	1 %
14	www.sciencegate.app Internet Source	<1 %
15	Submitted to International Institute of Information Technology, Hyderabad Student Paper	<1 %
	dvelopery0115.github.io	

16	Internet Source	<1%
17	Submitted to Universiti Kebangsaan Malaysia Student Paper	<1%
18	medium.com Internet Source	<1%
19	web.archive.org Internet Source	<1%
20	kobiso.github.io Internet Source	<1%
21	Rico Sennrich, Barry Haddow, Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units", 'Association for Computational Linguistics (ACL)', 2016 Internet Source	<1%
22	Submitted to University of Houston System Student Paper	<1%
23	nats-www.informatik.uni-hamburg.de Internet Source	<1%
24	Jiang Zhang, Chen Li, Ganwanming Liu, Min Min et al. "A CNN-transformer hybrid approach for decoding visual neural activity into text", Computer Methods and Programs in Biomedicine, 2022 Publication	<1%

25	procjx.github.io Internet Source	<1%
26	Moein Salimi Sartakhti, Mohammad Javad Maleki Kahaki, Seyed Vahid Moravvej, Maedeh javadi Joortani, Alireza Bagheri. "Persian Language Model based on BiLSTM Model on COVID-19 Corpus", 2021 5th International Conference on Pattern Recognition and Image Analysis (IPRIA), 2021 Publication	<1%
27	blog.tomrochette.com Internet Source	<1%
28	tsukuba.repo.nii.ac.jp Internet Source	<1%
29	www.qt21.eu Internet Source	<1%
30	Naseeb Singh, V.K. Tewari, P.K. Biswas, L.K. Dhruw, C.M. Pareek, H. Dayananda Singh. "Semantic segmentation of in-field cotton bolls from the sky using deep convolutional neural networks", Smart Agricultural Technology, 2022 Publication	<1%
31	Submitted to University of Witwatersrand Student Paper	<1%
32	aclweb.org	

	Internet Source	<1%
33	Submitted to University of Edinburgh Student Paper	<1%
34	epub.ub.uni-muenchen.de Internet Source	<1%
35	www.itm-conferences.org Internet Source	<1%
36	Vipul Mann, Venkat Venkatasubramanian. "Retrosynthesis Prediction using Grammar-based Neural Machine Translation: An Information-Theoretic Approach", Cambridge University Press (CUP), 2021 Publication	<1%
37	archive.org Internet Source	<1%
38	mafiadoc.com Internet Source	<1%
39	www.mdpi.com Internet Source	<1%
40	Charu C. Aggarwal. "Neural Networks and Deep Learning", Springer Science and Business Media LLC, 2018 Publication	<1%
41	core.ac.uk Internet Source	

		<1%
42	Submitted to Queen Mary and Westfield College Student Paper	<1%
43	acl.eldoc.ub.rug.nl Internet Source	<1%
44	citeseerx.ist.psu.edu Internet Source	<1%
45	ethen8181.github.io Internet Source	<1%
46	web.eecs.umich.edu Internet Source	<1%
47	Charu C. Aggarwal. "Artificial Intelligence", Springer Science and Business Media LLC, 2021 Publication	<1%
48	Submitted to University of Mumbai Student Paper	<1%
49	typeset.io Internet Source	<1%
50	Jiangbin Zheng, Zheng Zhao, Min Chen, Jing Chen, Chong Wu, Yidong Chen, Xiaodong Shi, Yiqi Tong. "An Improved Sign Language Translation Model with Explainable	<1%

Adaptations for Processing Long Sign Sentences", Computational Intelligence and Neuroscience, 2020
Publication

51	Aadil Gani Ganie, Samad Dadvandipour. "Identification of online harassment using ensemble fine-tuned pre-trained Bert", Pollack Periodica, 2022 Publication	<1 %
52	Submitted to Universidade do Porto Student Paper	<1 %
53	www.zora.uzh.ch Internet Source	<1 %
54	L. Trailovic, L.Y. Pao. "Computing Budget Allocation for Efficient Ranking and Selection of Variances With Applicationto Target Tracking Algorithms", IEEE Transactions on Automatic Control, 2004 Publication	<1 %
55	Submitted to Lehigh University Student Paper	<1 %
56	Sepp Hochreiter, Jürgen Schmidhuber. "Long Short-Term Memory", Neural Computation, 1997 Publication	<1 %
57	Sonja Nießen, Hermann Ney. "Statistical Machine Translation with Scarce Resources	<1 %

	Using Morpho-syntactic Information", Computational Linguistics, 2004 Publication	
58	www.springerprofessional.de Internet Source	<1 %
59	"Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data", Springer Science and Business Media LLC, 2018 Publication	<1 %
60	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2019 Publication	<1 %
61	Submitted to Liverpool John Moores University Student Paper	<1 %
62	Quang-Minh Do, Kungan Zeng, Incheon Paik. "Resolving Lexical Ambiguity in English-Japanese Neural Machine Translation", 2020 3rd Artificial Intelligence and Cloud Computing Conference, 2020 Publication	<1 %
63	Submitted to University College London Student Paper	<1 %
64	dazi.kukuw.com Internet Source	<1 %

65	www.proceedings.com Internet Source	<1%
66	Christopher G. Harris. "Using translational score maps to aid MT evaluation", 2011 7th International Conference on Natural Language Processing and Knowledge Engineering, 11/2011 Publication	<1%
67	Submitted to Higher Education Commission Pakistan Student Paper	<1%
68	S-space.snu.ac.kr Internet Source	<1%
69	"Proceedings of the International Conference on Paradigms of Computing, Communication and Data Sciences", Springer Science and Business Media LLC, 2021 Publication	<1%
70	etd.aau.edu.et Internet Source	<1%
71	ijmce.com Internet Source	<1%
72	openproceedings.org Internet Source	<1%
73	"Smart and Innovative Trends in Next Generation Computing Technologies",	<1%

Springer Science and Business Media LLC, 2018
Publication

---

74 Ahmad A. Muhammad, Amira T. Mahmoud, Shaymaa S. Elkalyouby, Rameez B. Hamza, Ahmed H. Yousef. "Trans-Compiler based Mobile Applications code converter: swift to java", 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), 2020 <1%
Publication

---

75 Benyamin Ahmadnia, Parisa Kordjamshidi, Gholamreza Haffari. "Neural Machine Translation Advised by Statistical Machine Translation: The Case of Farsi-Spanish Bilingually Low-Resource Scenario", 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018 <1%
Publication

---

76 Sanjib Narzary, Maharaj Brahma, Bobita Singha, Rangjali Brahma, Bonali Dibragede, Sunita Barman, Sukumar Nandi, Bidisha Som. "Attention based English-Bodo Neural Machine Translation System for Tourism Domain", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 2019 <1%
Publication

---

77 Submitted to University of Macedonia
Student Paper

<1 %

78 Vasco Alves, Jorge Ribeiro, Pedro Faria, Luis Romero. "Neural Machine Translation Approach in Automatic Translations between Portuguese Language and Portuguese Sign Language Glosses", 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022
Publication
<1 %

79 docplayer.net
Internet Source
<1 %

80 elib.uni-stuttgart.de
Internet Source
<1 %

81 tuprints.ulb.tu-darmstadt.de
Internet Source
<1 %

82 "Advances in Artificial Intelligence", Springer Science and Business Media LLC, 2019
Publication
<1 %

83 Charu C. Aggarwal. "Chapter 7 Recurrent Neural Networks", Springer Science and Business Media LLC, 2018
Publication
<1 %

84 Lucas Pedrosa Soares. "Segmentação automática de cicatrizes de deslizamento de terra em imagens de sensores remotos
<1 %

utilizando aprendizagem profunda de máquina (Deep Learning)", Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2022
Publication

85	Matúš Pikuliak, Marián Šimko, Mária Bieliková. "Cross-lingual learning for text processing: A survey", Expert Systems with Applications, 2021 Publication	<1 %
86	Sameen Maruf, Fahimeh Saleh, Gholamreza Haffari. "A Survey on Document-level Neural Machine Translation", ACM Computing Surveys, 2021 Publication	<1 %
87	bibis.ir Internet Source	<1 %
88	irlab.science.uva.nl Internet Source	<1 %
89	repositorio-aberto.up.pt Internet Source	<1 %
90	www.openaccess.hacettepe.edu.tr:8080 Internet Source	<1 %
91	"Computational Linguistics and Intelligent Text Processing", Springer Science and Business Media LLC, 2018 Publication	<1 %

92	"Text, Speech, and Dialogue", Springer Science and Business Media LLC, 2017 Publication	<1 %
93	Basab Nath, Sunita Sarkar, Surajeet Das. "Chapter 27 Development of Neural Machine Translator for English-Assamese Language Pair", Springer Science and Business Media LLC, 2022 Publication	<1 %
94	Mohammad Taher Pilehvar, Jose Camacho-Collados. "Embeddings in Natural Language Processing", Springer Science and Business Media LLC, 2021 Publication	<1 %
95	Sahinur Rahman Laskar, Abinash Dutta, Partha Pakray, Sivaji Bandyopadhyay. "Neural Machine Translation: English to Hindi", 2019 IEEE Conference on Information and Communication Technology, 2019 Publication	<1 %
96	Submitted to The University of Manchester Student Paper	<1 %
97	amslaurea.unibo.it Internet Source	<1 %
98	booksc.me Internet Source	<1 %
	data-science-blog.com	

99	Internet Source	<1%
100	ddd.uab.cat Internet Source	<1%
101	ir.library.osaka-u.ac.jp Internet Source	<1%
102	journalofcloudcomputing.springeropen.com Internet Source	<1%
103	ore.exeter.ac.uk Internet Source	<1%
104	research.aalto.fi Internet Source	<1%
105	samewood.com Internet Source	<1%
106	tel.archives-ouvertes.fr Internet Source	<1%
107	www.gabormelli.com Internet Source	<1%
108	www.jstage.jst.go.jp Internet Source	<1%
109	www.mitpressjournals.org Internet Source	<1%
110	www.researchgate.net Internet Source	<1%

111	Marta R. Costa-jussà, David Aldón, José A. R. Fonollosa. "Chinese–Spanish neural machine translation enhanced with character and word bitmap fonts", Machine Translation, 2017 Publication	<1 %
112	Zhongxin Liu, Xin Xia, Christoph Treude, David Lo, Shanping Li. "Automatic Generation of Pull Request Descriptions", 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019 Publication	<1 %
113	"Chinese Computational Linguistics", Springer Science and Business Media LLC, 2019 Publication	<1 %
114	"Machine Translation", Springer Science and Business Media LLC, 2019 Publication	<1 %
115	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2018 Publication	<1 %
116	Amarnath Pathak, Partha Pakray. "Neural Machine Translation for Indian Languages", Journal of Intelligent Systems, 2019 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		