ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Neural Machine Translation from Polish to English Using Various Techniques

Manav Jain[1, a)], Vatsal Chheda[1, b)], Bhavya Lakhani[1, c)] and Lakshmi Kurup[1, d)]

Author Affiliations

[1]*Computer Engineering, D.J Sanghvi College of Engineering, Mumbai, India*

Abstract: Neural Machine Translation is a new approach to Machine Translation that makes use of Artificial Neural Networks. Machine Translation can be used to overcome the language barrier hence can reduce the communication gap between individuals. There are few models available for translation of polish to English. This paper discusses various approaches of Neural Machine Translation, namely Sequence to Sequence, Sequence to Sequence with Attention Mechanism, and Transformer Model. Encoder-Decoder Structure is widely used in the models mentioned above. The three models mentioned above are trained and carefully studied with the help of metrics such as BLEU, WER and GLEU. The three models are compared with the help of metrics mentioned above and have proven that the best model available is the Transformer Model. The BLEU score for Transformer Model is maximum indicating the precision and predictiveness of the Transformer Model is the best amongst the three models with the value of 59.68. Sequence to sequence and Sequence to Sequence with Attention mechanism achieved an accuracy of 30.45 and 34.33, respectively.

Keywords: *Attention Mechanism, BLEU, Encoder-Decoder, GRU, Long Short Term Memory Cells, Machine Translation, Transformer*

INTRODUCTION

There are a number of languages that are spoken in today's world. According to a survey conducted by World Almanac and Book of facts, approximately 50 million people in this world can communicate in polish [22]. Polish is written in the traditional 32-letter Polish alphabet and can sometimes include x, q, v in the extended 35-letter alphabet. This 32-letter alphabet consists of 23 consonants and 9 written vowels including 2 nasal vowels and is most closely related to Slovak and Czech. There's no use of articles and pronouns are dropped frequently in Polish. This accounts to our interest in choosing Polish as the source language for translation. In the English sentence, the sentence is ordered according to the Subject-Verb-Object format. However, in polish there's no specific order but the most dominant form is SVO form. This makes it difficult to translate as multiple polish sentences can have same English translation or vice versa. For Example, "Zrobiłem tłumacza do mojego projektu na studia", "Zrobiłem tłumacza dla mojego projektu uczelni" and "Zrobiłem tłumacza dla mojego projektu uniwersyteckiego" translates to "I made a translator for my college project" in English. English being a universal language, it makes sense to make a translator that takes input as Polish and provides English as the output.

Machine Translation (MT) acts as a bridge for Bilingual Communication. When a language is translated, MT preserves the meaning of the source language in the target language. Basically, corpora-based MT systems are classified into Statistical Machine Translation System (SMT), Phrase-Based Translation System, Example-Based Machine Translation System [13], and Neural Machine Translation [18]. Neural Machine Translation (NMT) is an innovative approach to Machine Translation that makes use of an Artificial Neural Network to predict the probability of the sequence of words. With the use of ANNs, a typical NMT can be trained directly on the source and target text, thereby eliminating the need of pipelining specialized systems used in SMTs. Neural Machine Translation consists of various types of Encoder-Decoder structures like sequence to sequence, Sequence to Sequence with attention mechanism, transformers etc. [12]. NMTs makes use of the vector representation of words, and the structure of the model and internal states is comparatively simpler than a phrase based or an example-based system.

We have used Neural Machine Translation techniques like Sequence to Sequence, Sequence to Sequence with Attention Mechanism, and Transformer Model for translation of Polish sentences to English. Various metrics have been used to calculate the accuracy of models such as GLEU, WER, and BLEU scores. Models are built using LSTMs and GRUs cells as they have proven to be more effective with NMT Systems because of its inherent ability to store longer sequences.

The paper is further divided into 5 sections. The second section of the paper i.e., Literature Review gives valuable insights into various NMT models. In the third section, we have explained the details about the dataset and how data is preprocessed before being fed to the model. The fourth section details about our proposed architecture of the system. Here, we have used three Models namely Sequence to Sequence, Sequence to Sequence with Attention Mechanism and Transformer Based Model. Further, the paper ends with results and conclusion.

RELATED WORK

The paper[1] by Wołk, K., et. Al, was aimed to create a translator from Polish to English. The authors of the paper have created this system by using the Statistical Machine Translation System concepts. The model was trained on the medical text. The number of sentences used was 1,044,764. MGIZA++ tool was used for word and phrase alignment. Various metrics were calculated, such as BLEU, NIST, METEOR, RIBES, and TER. In the case of TER, the lower the value, the better the model, and in the case of other metrics greater the value, the better the model. In the case of the bleu score, the highest score achieved was 72.51. In case of NIST, METEOR, and RIBES, the average values were 10.99,85.17, and 85.12, respectively. The hierarchical Phrase-Based method was also used to build the model. In this method, they combine phrase-based translation and syntax-based translation methods. The search model used in this case is very much like the syntactic chart parsing and these models can be classified into tree-based and grammar-based models.

The paper[2] by Dutta, A., et. al, attempts at minimizing the language barrier between people by proposing the machine translation between English and Hindi. This paper has proposed namely two models, i.e. NMT-1 and NMT-2. NMT-1 is based on Long Short Term Memory and uses the Attention Mechanism. NMT-2 is based on the Transformer Model. They have used two different test sets for testing their models. They have used the Bilingual Evaluation Understudy Score (BLEU Score) to calculate their models' metrics. The BLEU score from the NMT-1 model achieved on test set1 was 35.89, and on test set 2 was 19.91. The BLEU score from the NMT2 model achieved on test set1 was 34.42, and on test set 2 was 24.74. They were able to achieve better accuracy than the current NMT Systems for English to Hindi. NMT-1 was used because it was able to make predictions more like the Google and Bing translator. In their research, it was seen that NMT-1 had performed better on interrogative marks than NMT-2. NMT-2 was able to perform well in case if an unknown word came up in the source sentence compared to NMT-1.

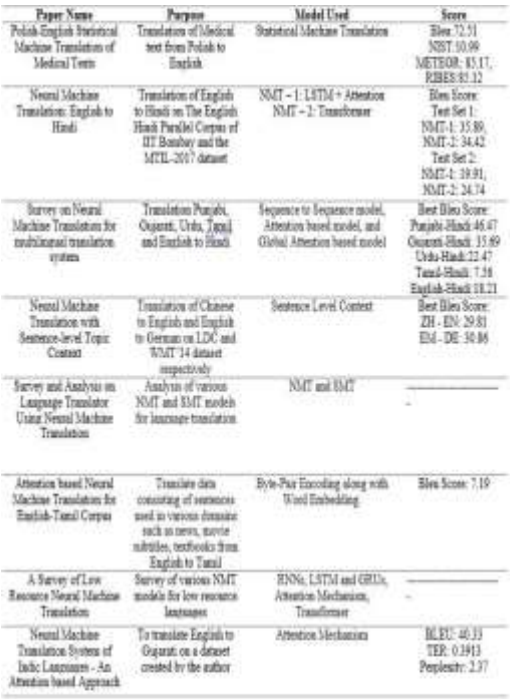
The paper[3] by Kaushal, S., et. al, attempts to make a model for Neural Machine Translation between English and Many native Indian Languages. They have used layers of Long Short Term Memory (LSTMs) to expand their memory. LSTMs are the extension of RNNs. They have used various models such as Sequence to Sequence model, Attention-based model, and Global Attention-based model. The authors of this paper have configured their model by changing the layers of LSTM by also changing the optimizer and have shown how the accuracies have changed. They have even out-beaten the current Google Translator in a few scenarios. For English to Hindi, they used ILCC, UFAL, CFLIT Datasets, and the number of tokens used was always greater than 8k. For Punjabi to Hindi, they achieved a bleu score of 46.47, and for Gujarati to Hindi, they have achieved a bleu score of 35.69.

The paper[4] by Sumita, E., et. al, has used another approach to increase the accuracy of the Machine Translation Systems. The traditional NMT systems have used a word-level context for predicting the target language. The authors of this paper have proposed an NMT System which is based on Sentence Level Context. To achieve this, they have used Convolutional Neural Network, which designs topic attention, and this can be used on both Attention-based Model and Transformer Based Model. They have shown that the model they have used outperformed the existing model and have shown that this approach is more effective. They have conducted their experiments on Chinese to English and English to German.

The paper[5] by Jain, A., et. al, aimed to analyze how Neural Machine Translation is better as compared to Statistical Machine Translation. They have used the encoder-decoder structures and have shown how it is better compared to the SMTs. The paper has shown various methods of performing NMT from which the accuracy of the model can be increased. Models such as a Simple LSTM Encoder-Decoder, Encoder-Decoder LSTM with Attention Mechanism and Bidirectional Encoder-Decoder with an attention mechanism were implemented. They have also explained all of the models mentioned above in depth. They have used only Single Bi-directional Layer as the training time increases when using bidirectional layers, whereas, in the case of Simple LSTM Encoder-Decoder and LSTM Encoder-Decoder with attention mechanism, multiple layers can be used because their architecture is not that and, the model gets trained within a reasonable amount of time.

The paper[6] by Pathak, A., et. al, has used the Attention Based Neural Machine Translation to create their model. The source language was English, and the target language was Tamil. The authors of the paper have employed an NMT Technique called Byte-Pair Encoding along with Word Embedding. Word Embedding can be used to overcome the Out of the Vocabulary (OOV) issue. Word Embedding was used because some certain phrases and idioms do not have an equivalent sentence in English available, So to overcome that issue, they have used Word Embedding in their model. The authors of the paper have carried out the task approximately on 220k lines of Bilingual Corpus. They have achieved a bleu score of 7.19. They have proved in their report that when you use Byte-Pair Encoding and word embedding, the results turn out to be satisfactory and hence can we be used for other language pairs as well.

TABLE 1: *Multiple papers with the method used and scores*



Paper Name	Purpose	Model Used	Score
Polish-English Statistical Machine Translation of Medical Texts	Translation of Medical text from Polish to English	Statistical Machine Translation	Bleu 72.51 NIST 10.99 METEOR: 85.17, RIBES 85.12
Neural Machine Translation: English to Hindi	Translation of English to Hindi on The English Hindi Parallel Corpus of IIT Bombay and the MTIL-2017 dataset	NMT – 1: LSTM + Attention NMT – 2: Transformer	Bleu Score: Test Set 1: NMT-1: 35.89, NMT-2: 34.42 Test Set 2: NMT-1: 19.91, NMT-2: 24.74
Survey on Neural Machine Translation for multilingual translation system	Translation Punjabi, Gujarati, Urdu, Tamil and English to Hindi	Sequence to Sequence model, Attention based model, and Global Attention based model	Best Bleu Score: Punjabi-Hindi 46.47 Gujarati-Hindi: 35.89 Urdu-Hindi 22.47 Tamil-Hindi: 7.58 English-Hindi 18.21
Neural Machine Translation with Sentence-level Topic Context	Translation of Chinese to English and English to German on LDC and WMT'14 dataset respectively	Sentence Level Context	Best Bleu Score: ZH - EN: 29.81 EN - DE: 30.86
Survey and Analysis on Language Translator Using Neural Machine Translation	Analysis of various NMT and SMT models for language translation	NMT and SMT	———————— -
Attention based Neural Machine Translation for English-Tamil Corpus	Translate data consisting of sentences used in various domains such as news, movie subtitles, textbooks from English to Tamil	Byte-Pair Encoding along with Word Embedding	Bleu Score: 7.19
A Survey of Low Resource Neural Machine Translation	Survey of various NMT models for low resource languages	RNNs, LSTM and GRUs, Attention Mechanism, Transformer	———————— -
Neural Machine Translation System of Indic Languages - An Attention based Approach	To translate English to Gujarati on a dataset created by the author	Attention Mechanism	BLEU: 40.33 TER: 0.3913 Perplexity: 2.37

The paper[7] by Ma, N., et. al, has proposed a Neural Machine Translation System on Low Resource Language. Low Resource languages are the languages for which corpus is not available much. There are many problems when you work with low-resource languages. There are many challenges one has to face while working with such a corpus. Attention-Based Models and Transformers Models can be made even more effective. The authors of this paper have proposed a few methods that can increase the accuracy of the models. They have made use of the Integrated bilingual dictionary method. In this method, they first introduce discrete probability words into Neural Machine Translation. Then they find the target word by using the Traditional NMT. The lexical probability of the statement is converted into conditional prediction probability, and then the matrix is formed from the input statement. The matrix which was created in the previous step is then transformed into the prediction probability of the next target word, and each and every column of the matrix is weighted. Finally, the efficiency of the Machine Translation System can be improved by making use of automated dictionaries and manual dictionaries. Multitask Multilingual Neural Machine Translation is also one method that the authors of the paper propose. The authors have also suggested that the Transfer Learning Method can increase the accuracy of the Neural Machine Translation on the Low Resource Language.

The paper[8] by Shah, P., et, al, has emphasized on creating the NMT model on Indic Languages. They have translated English to Gujarati and have approximately used 65000 lines of the corpus. They have made use of the Attention mechanism on top of the encoder-decoder structure. For their model, they have used two LSTM layers to increase the model's efficiency. They have used two LSTM layers on both sides i.e., the encoder side and decoder side. In the LSTM layer, 128 LSTM cells were used. They were able to achieve better accuracy when they made use of Attention-Based Mechanism. For English to Gujarati, they outperformed the Google translator. They achieved a BLEU score of 40.33, whereas the BLEU score of GNMT is 33.66. They also achieved a TER of 0.3913, whereas GNMT had a TER of 0.5217. TER value should be as low as possible because lower the TER better the model.

PROPOSED METHOD

Data Used

The authors have used the parallel Corpus data from "https://www.manythings.org/anki/". We have processed around 40045 samples from the dataset. The size of the input vocabulary (Polish) is 16259 and the size of the output vocabulary (English) is 12483.

Dataset Pre-Processing

The corpus data was almost clean. The authors removed all the non-breaking periods in the sentences. The paper uses sentences with a maximum sentence length of 50. The target sentences were prefixed and suffixed by the START and END Keyword. The authors padded the shorter sentences after the sentence using the Keras pad_sequences method. The dataset was tokenized using the TensorFlow dataset's SubWordTextEncoder.

TABLE 2: *Dataset Example*

Polish	English
To nie krew.	It's not blood.
Chciałbym mieć trochę czasu, żeby porozmawiać z Tomem.	I'd like some time to talk with Tom.
Chyba jestem w sporych tarapatach.	I think I'm in big trouble.

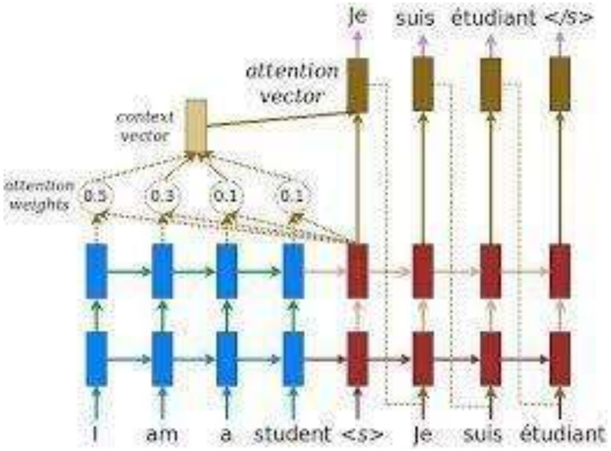
Model Architecture

Sequence to Sequence

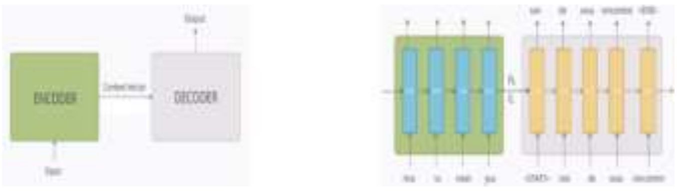
Sequence to Sequence Model, also known as seq2seq, is a type of Recurrent Neural Network (RNNs). Machine Translation makes use of many to many type of sequence models. Sequence to Sequence can be built by using LSTMs (Long Short Term Memory) and GRU (Gated Recurrent Unit). LSTMs and GRUs are used instead of RNNs because those models help in solving the vanishing gradient problem. Sequence to Sequence does not perform too well on long sentences as it does not have that much memory to store the longer sequences [20]. The most common structure adopted by the Sequence to Sequence model is Encoder-Decoder Structure. The encoder-Decoder structure consists primarily of an encoder block, context vector, and decoder block.

Encoder-: The function of the encoder block is to process each and every token present in the input sentence, and then it tries to squeeze the information of the input sequence into a vector of fixed length. The context vector contains all the information about the sequence. The context vector is then fed to the decoder block.

Decoder -: The context vector is processed in the decoder block. The decoder block first reads the block, and then it tries to predict the sequence of the other language token by token [19].



The image (Figure 1(a)) shows what's happening inside the encoder and decoder structure. Encoder and Decoder structure contains LSTMs or GRUs cells or maybe some other special type of RNN structure. The number of units in the cells is decided by the maximum length of the sentence in that particular language. The cells are fed with the input sequence over some time again and again. These cells contain states. There are two states namely: hidden state(state_h) and cell state(state_c). The role of the hidden states is basically to carry them forward to the decoder structure because these states serve as the initial state for decoder structure, and they can be used for predicting the target word [20].



(a) (b)

FIGURE 1: *High-Level Overview of Seq2Seq Model, Working of Sequence to Sequence, (a), (b)*

The diagram (figure1(b)) shows that in the encoder block, the input sequence is fed to the units at timesteps. The output of the encoder i.e., the internal states and the context vector, is passed to the decoder block. On the decoder side, a token such as "<START>" or "START_" is given to the decoder structure, which signifies that it is the start of the output sequence. As that token is passed as the first input to the decoder block, it produces the target word with the help of internal states, which were provided by the encoder block [19]. The target word produced is then passed as an input to the decoder, and then another target word is produced. This process goes on and on until an end token is produced. As the end token is produced, the model comes to know that the sequence has been completed, and the output sequence is a machine-translated equivalent for the source language. This is how an ideal encoder-decoder structure works. This is how the model works in the case of the testing phase. The training phase is a bit different as they are used to adjust the internal states. Usually, the training phase uses Teach Forcing, which makes the training faster [19].

The authors in this paper have made use of LSTMs. The encoder block consists of an Embedding Layer and LSTMs cells. Similarly, on the decoder block, the authors have used Embedding and LSTM layers. There is one dense layer that was added for the prediction of the target word. The activation function used is a softmax function. The number of units is decided by the number of unique tokens in the target language. Teach Forcing was used to make the learning faster in the training phase because if not used, these models are likely to take a lot of time to get trained [19].

Attention Model

The standard Encoder-Decoder Seq to Seq Model had significant drawbacks. The output of the encoder of standard sequence to sequence relies on the hidden states of the encoder and in cases of long sentences, the initial context is lost at the end of the sequence most of the time. It does not perform well on longer sentences [14]. This was mainly due to the meaning of the sentence getting lost in the translation. So, one of the solutions to this is to find a mechanism where the meaning of the preceding parts of sentences is not lost [14]. To implement this, we made use of the attention mechanism, which was designed by Bahdanau. Bahdanau created a mechanism in which the meaning of previous words is stored. There are two types of attention: a. Global Attention b. Local Attention. In global attention, the meaning of all the previous words i.e., states of all the previous encoders, are stored, whereas in case of Local attention, only a few selected states are stored. The architecture of the model used in this paper was Encoder-Decoder Architecture.

FIGURE 2: *Sequence to Sequence with Attention*

The architecture here has used multiple GRU and Embedding units. GRU cell (Gated Recurrent Unit) is similar to LSTM and the newer generation of Recurrent Neural Networks. It uses only two gates: 1) Update Gate 2) Reset Gate. Update gate acts similar to the forget and input gate of LSTM and it decides what information to let go of and what information to remember. Similarly, a reset gate is used to determine how much past information to forget [14].

For Encoding, the authors have used an embedding layer followed by the GRU layer and the output of the encoder which is hidden states and the cell states, are fed to the Decoder object. The encoder takes the tokens (words) from the sequence and looks up for a corresponding embedding vector using the Embedding layer of Keras followed by processing these embeddings into a new sequence using a GRU layer from Keras and will return the processed sequence which is used by the attention head and the Internal state which will be used to initialize the decoder.

In the attention layer, to learn the attention given to each word in the sequence, a feed-forward Neural Network is used whose inputs are hidden states from the encoder as well as the decoder outputs which are fed to tanh activation function here [16]. The attention weights are calculated by applying the softmax function to the output of the first layer. The score is calculated by applying a tanh activation function to the inputs of the decoder followed by a softmax function to calculate attention weights. A context vector is calculated from attention weights and encoder outputs. The following equations are used in this mechanism:

$$= \frac{(\quad (h, h))}{\Sigma_{s'=1} \quad (\quad (h, h))}$$

$$= \sum \quad h$$

$$= \quad (\quad , h) = \quad h(\quad [\quad ; h])$$

$$- \quad h \quad \bar{h}$$

$$(h, h) = \{$$

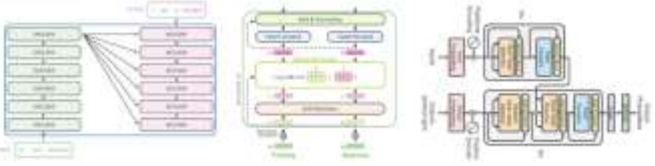
$$h(\quad {}_1h + \quad {}_2\bar{h})$$

EQUATIONS: *Attention Weights(1), Context Vector(2), Attention Vector(3) and Luong's Multiplicative and Badhanau's Additive style(4) [14].*

The Decoder is formed using Embedding, GRU and multiple Dense layers. The sequence from the encoder is passed to the decoder which is passed to the GRU to keep track and then its output is used as a query for the attention layer to get the context vector and produces the attention vector which is used to generate the next token of the sequence. The loss function used here is Adam's Optimizer.

Transformer

All the previous sequence language models were based on complex recurrent or convolutional neural networks in an encoder-decoder configuration. The paper by Ashish Vaswani et. al; discussed a novel approach that is solely based on the self-attention mechanism. This new architecture is more accurate, more parallelizable, and faster to train [9].



(a) (b) (c)

FIGURE 3: *High Level overview of transformer, Transformer Encoder Architecture, Transformer Model Architecture (a),(b),(c)*

In Transformer the authors have used an architecture similar to encoder and decoder. Here, a stack of 6 Encoders and 6 decoders are used. Each encoder consists of two layers. First, it is passed through the Self Attention layer and then fed to the Feed Forward Neural network. Similarly, the decoder is an encoder with an extra layer called the Encoder-Decoder Attention layer stacked between the two layers. Only the first encoder in the stack has an embedding layer. The input is embedded and then passed to each of the two layers of encoders [17].

The embedding of the first word is passed to the self-attention layer which calculates 3 vectors: Key, Value and Query Vectors. To calculate the self-attention score is calculated by calculating the dot product of its corresponding query and key vector. This is then divided by the square root of the dimensions of key vectors to have more stable gradients and then passed to the softmax function to normalize these scores [21]. This softmax score is multiplied by the value vectors and followed by the sum of all the weighted value vectors. The self-attention layer is a multi-headed attention layer so the above process is done 8 times and then concatenated and multiplied with desired weight matrix to get the Final Score. This is the final output of the self-attention layer that is added to the embedding and normalized whose output is passed to the Feed Forward Neural Network [21].

The decoder works similar to the encoder; the only difference is the second self-attention layer. In the decoder, the self-attention layer is only allowed to attend to earlier positions in the output sequence. This is done by masking succeeding positions before the softmax step in the self-attention calculation. It creates its Queries matrix from the layer below it and takes the Keys and Values matrix from the output of the encoder stack. The output of the decoder is sent to the final linear layer followed by the softmax layer which gives us the final translation.

Since the model contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, the authors injected some information about the relative or absolute position of the tokens in the sequence [9].

$$= \quad (\quad /10000^{2\,/} \quad) \qquad (5)$$

(,2)

$$= (\ /10000^{2\ /}\ ) \qquad (6)$$

(,2)

EQUATIONS: *Calculation of Positional Encoding Vector (5,6)*

RESULT AND CONCLUSION

After executing the three models and fine-tuning them, we compared our model using our automation testing. Our test set consisted of 405 sentences of length ranging from 1 to 15 words.

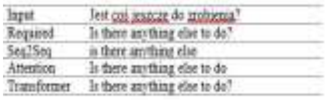
In the first test, we compared the Sacrebleu score of the models on our test set. It is clearly visible that the Transformer outperformed the Attention model as well as the Seq2Seq model.

In the second test, we compared the gleu score. It was seen that all the three models performed similarly but the Transformer model was slightly better than the other two models.

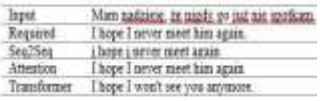
In the third test, we compared the word error rate(WER). It was seen that the attention model has a slightly less word error rate than the transformer. The Seq2Seq model had very high WER for sentences of length greater than 7.

In terms of Bleu Score, Transformer had the highest Bleu score, and Seq2Seq and Attention Model had comparable Bleu Score. However, in terms of word error rate Attention Model performed the best followed by transformer and Seq2Seq. So, overall Transformer architecture performed better than the remaining two models. It was also seen that the sentences generated by the transformer model and attention model were grammatically as well as logically correct.

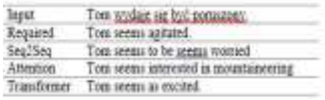
The input and output of various models are listed below:



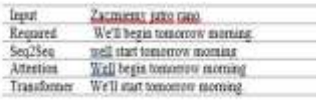
Input	Jest coś jeszcze do zrobienia?
Required	Is there anything else to do?
Seq2Seq	is there anything else
Attention	Is there anything else to do
Transformer	Is there anything else to do?



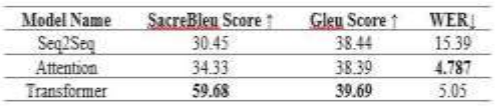
Input	Mam nadzieję, że nigdy go już nie spotkam.
Required	I hope I never meet him again.
Seq2Seq	i hope i never meet again
Attention	I hope I never meet him again
Transformer	I hope I won't see you anymore.



Input	Tom wydaje się być poruszony.
Required	Tom seems agitated.
Seq2Seq	Tom seems to be seems worried
Attention	Tom seems interested in mountaineering
Transformer	Tom seems so excited.



Input	Zaczniemy jutro rano.
Required	We'll begin tomorrow morning.
Seq2Seq	well start tomorrow morning
Attention	Well begin tomorrow morning
Transformer	We'll start tomorrow morning



Model Name	SacreBleu Score ↑	Gleu Score ↑	WER↓
Seq2Seq	30.45	38.44	15.39
Attention	34.33	38.39	**4.787**
Transformer	**59.68**	**39.69**	5.05

TABLE 3: *SacreBleu Score, Gleu Score and WER for various models*

ACKNOWLEDGMENTS

This research was supported by Prof. Lakshmi Kurup. We thank her for providing her valuable insight and expertise, which greatly assisted this research, and her comments have helped us improve our manuscript significantly

REFERENCES

1. Wołk, K., & Marasek, K. "Polish-English Statistical Machine Translation of Medical Texts." *New Research in Multimedia and Internet Systems, 169–179.* (2015)

2. Laskar, S. R., Dutta, A., Pakray, P., & Bandyopadhyay, S. "Neural Machine Translation: English to Hindi." *2019 IEEE Conference on Information and Communication Technology.* (2019)

3. Basmatkar, P., Holani, H., & Kaushal, S. "Survey on Neural Machine Translation for multilingual translation system."

2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). (2019)

4. Chen, K., Wang, R., Utiyama, M., Sumita, E., & Zhao, T. "Neural Machine Translation with Sentence-level Topic Context." *IEEE/ACM Transactions on Audio, Speech, and Language Processing.* (2019).

5. Verma, Ms. Neeta, Abhay Jain, Animesh Basak and Kshitij Bharti Saksena. "Survey and Analysis on Language Translator Using Neural Machine Translation." *International Research Journal of Engineering and Technology (IRJET) Volume: 05 Issue: 04* (2018).

6. Choudhary, Himanshu & Pathak, Aditya & Saha, Rajiv & Kumaraguru, Ponnurangam. "Neural Machine Translation for English-Tamil."*International Research Journal of Engineering and Technology (IRJET) Volume:07 Issue: 04*. (2020).

7. Liu, D., Ma, N., Yang, F., & Yang, X. "A Survey of Low Resource Neural Machine Translation." *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE).* (2019).

8. Shah, P., & Bakrola, V. "Neural Machine Translation System of Indic Languages - An Attention based Approach."

2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP). (2019)

9. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., "Attention is all you need." *Advances in neural information processing systems*

10. Bahdanau, D., Cho, K. and Bengio, Y., “Neural machine translation by jointly learning to align and translate.” *arXiv preprint arXiv:1409.0473*. (2014).

11. Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." In *Advances in neural information processing systems*, pp. 3104-3112. (2014).

12. P. Koehn, *Statistical machine translation*. Cambridge: Cambridge University Press, 2014.

13. Ma, Chunpeng, et al. "Syntax-based Transformer for Neural Machine Translation." *Journal of Natural Language Processing* 27.2 (2020): 445-466.

14. Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." *arXiv preprint arXiv:1508.04025* (2015).

15. Cheng, Yong. "Agreement-based joint training for bidirectional attention-based neural machine translation." In *Joint Training for Neural Machine Translation*, pp. 11-23. Springer, Singapore, 2019.

16. Zhang, Jinchao, Mingxuan Wang, Qun Liu, and Jie Zhou. "Incorporating word reordering knowledge into attention-based neural machine translation." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1524-1534. 2017.

17. Kumar, Amit, Rupjyoti Baruah, Rajesh Kumar Mundotiya, and Anil Kumar Singh. "Transformer-based Neural Machine Translation System for Hindi–Marathi: WMT20 Shared Task." In Proceedings of the Fifth Conference on Machine Translation, pp. 393-395. 2020.

18. "Neural machine translation -Wikipedia", *En.wikipedia.org*, 2021.[Online]. Available: https://en.wikipedia.org/wiki/Neural_machine_translation. [Accessed: 24- Jul- 2021].

19. "Encoder-Decoder Seq2Seq Models, Clearly Explained!!", *Medium*, 2021.[Online]. Available: https://medium.com/analytics-vidhya/encoder-decoder-seq2seq-models-clearly-explained-c34186fbf49b. [Accessed: 24- Jul- 2021].

20. A. Models and G. Blog, "Seq2Seq Model | Understand Seq2Seq Model Architecture", *Analytics Vidhya*, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/08/a-simple-introduction-to-sequence-to-sequence-models/. [Accessed: 24- Jul- 2021].

21. J. Alammar,"The Illustrate Transformer", *Jalammar.github.io*, 2021.[Online]. Available: https://jalammar.github.io/illustrated-transformer/. [Accessed: 24- Jul- 2021].

22. "Polish language -Wikipedia", *En.wikipedia.o*
https://en.wikipedia.org/wiki/Polish_language. [Accessed: 24- J
2021].

23. "English Language Statistics - an Exhaustive List | Lemon Grad", *Lemon Grad*, 2021. [Online]. Available: https://lemongrad.com/english-language-statistics/. [Accessed: 24- Jul- 2021].