An Investigation of Language-Specific Adaptation: Transformer-based Machine Translation

1st Yerkebulan Akbay
Department of Computational Mathematics and Data Science
Astana IT University
Astana, Kazakhstan
211533@astanait.edu.kz

2nd Dilnura Gabdollayeva
Department of Computational Mathematics and Data Science
Astana IT University
Astana, Kazakhstan
211494@astanait.edu.kz

***Abstract*—Neural machine translation is commonly recognized as a significant yet the hardest technological advancement within the field of Natural Language Processing. This paper will first present an overview of data preparation and its preprocessing. Upon examination of the model's architecture, particular attention is given to Transformers that rely on attention mechanisms. Furthermore, a detailed examination of the process of transforming textual data into numerical representations, including byte-pair encoding, and word embedding, and afterwards decoding the resulting output sequence will be undertaken. Subsequently, model validation will be conducted by loading a graph of checkpoints followed by the evaluation of performance through benchmarking and measuring accuracy, leveraging the BLEU score. After, the Transformer model will be adapted specifically for English and Russian languages. In conclusion, the finalized pipeline is designed to receive input in English language and produce output in Russian as a translation and vice versa.**

***Index Terms*—MT, NLP, Transformer, Self-Attention, Encoder, Decoder**

I. Introduction

Machine Translation (MT) is a product of human ingenuity which facilitates the automated transfer of written communication from one language to another. Its significance has been progressively acknowledged in contemporary times. Due to the massive convenience and fast transportation platforms, frequent communications among people in different countries are really common and drive a deeper need of translations for everyone anytime and anywhere [6]. However, MT strictly operates by substituting terms from one language to another, yet such a method may not completely guarantee proficient translation.

The integration of contemporary *Natural Language Processing* (NLP) and *Neural Network* (NN) advancements has facilitated the augmentation of computerized MT systems. The present study highlights the task of identifying multiple phrases in a given text, which can be achieved without the requirement of manual intervention. The objective can be efficiently accomplished through automated machine-driven conversion procedures.

Recurrent Neural Networks (RNNs), where elements connect to each other and form a directed sequence, have been used for the first architectures that have surpassed the development and features of machine translation. The RNN-based NMT approach, or RNMT, was quickly established as the de-facto standard for NMT, and gained rapid adoption into large-scale systems in industry, e.g. Baidu, Google, and Systran [3]. Then, with the help of Convolutional Neural Networks (CNNs), which are capable of fully parallelizing learning and have the advantage of fast work, which makes them more efficient in dealing with large amounts of data, MT has been taken to a new level. The ConvS2S model was shown to outperform the original RNMT architecture in terms of quality, while also providing greater training speed [3]. Despite the advantages of these two types of neural networks, there are drawbacks that have a negatory effect, such as attenuation, gradient vanishing, not being able to model long contexts and other overheads which significantly reduce performance.

In the year 2017, a publication by Google researchers presented a novel Neural Network framework designed to model sequence [11]. The *Transformer* architectural model exhibited superior performance in machine translation tasks compared to the RNN model, showcasing preferable translation quality and cost-efficiency in its training. The aforementioned technological progressions have the development of two of the most sophisticated transformers to date: the Generative Pre-trained Transformer (GPT) and Encoder-from-Transducer Bidirectional Representation (BERT), which integrate transformer architecture with unsupervised learning. The utilization of models has negated the necessity of acquiring task-specific structures from the ground up and surpasses nearly all evaluations in NLP by substantial margins. Without explicitly adding structural constraints, transformers are able to encode a large amount of structural information. However, being a rather new architecture, little is known about what the model exactly learns internally. A better understanding of the internal representations of neural models has become a major challenge in NMT [8].

This paper provides an exploration aimed at enhancing the performance of the basic transformer machine translation model, with the same or similar parameters from the source paper, through experimentation with various hyperparameters and evaluation metrics. Additionally, the study focuses on improving the explainability and interpretability of the model by employing visualization techniques such as heatmaps and alignment matrices, providing valuable insights into the decision-making process of the model. Furthermore, the research investigates the adaptation of the model specifically for English and Russian languages by analyzing the linguistic characteristics and challenges inherent to these languages. Proposed modifications include the incorporation of language-specific positional encodings, strategies for handling the agglutinative nature of the Russian language, and the customization of attention mechanisms to accommodate language-specific word orders. The findings of this study contribute to the understanding machine translation models, particularly in the domains of performance enhancement, explainability and interpretability, and language-specific adaptation for English and Russian languages.

II. Data

Our study on Neural Machine Translation involved the usage of a comprehensive dataset from Yandex encompassing a vast corpus of one million pairs of parallel sentences in both Russian and English, selected in a randomized manner from experimental corpora derived from parallel documents that were automatically scraped from the Internet. The dataset was chosen as it contains a large volume of textual data in both languages, which are used to make up the main corpus for various machine translation tasks. A parallel character of data allows the model to run on pairs of sentences, which have a significant improvement in its quality. Moreover, they mitigate problems related to correlation of different languages in translation.

Sentences were encoded through the use of byte-pair encoding, which incorporates a source-target vocabulary comprised of almost 150,000 tokens. The pairing of sentences was determined by grouping them according to their approximate length of sequential units.

III. Data Preparation and Data Preprocessing

A. Denoising

The concept of data noise has been demonstrated to have a significant impact on the performance of neural machine translation (NMT) systems. The noise is an unpleasant factor that may result from a variety of reasons such as grammatical errors, spelling mistakes, different contextual and language ambiguity, the presence of idioms, slang, etc. The usage of direct translation approach poses a challenge, as it may result in the loss of intended meaning and the detection of inconsistencies within target message. This demonstrates the subtlety of identifying noisy data for machine translation. It is not a simple binary problem: Some training samples may be partially useful to training a model, and their usefulness may also change as training progresses [13].

B. Normalization

The process of normalization is needed when handling unprocessed strings with the objective of enhancing their cleanliness. One potential illustration could be the elimination of white space and the exclusion of characters possessing diacritical marks. *Unicode normalization* is a frequently employed operation used by numerous tokenizers to address the issue of disparate manners of representing a given character. Schemes, including NFC, NFD, NFKC, and NFKD, are implemented to substitute diverse methods of expressing identical characters with conventional configurations. The subsequent example of normalization that we made use of for our studies is *lower case*. The present methodology is capable of diminishing the scope of the dictionary that is necessary for a given task. After the implementation of such normalization techniques, a representative string will look like this: “привет, друг”.

C. Pretokenization

This step entails segregating the text into smaller units compared to its initial state. These units provide an upper limit on the final tokens that would be generated upon completion of the model training process. In order to make data easier to understand or work with, pretokens can then be further processed and interpreted on a regular basis. One common pre-tagging technique is byte-pair encoding (BPE), which involves identifying frequently occurring character sequences and treating them as tokens. The procedure of pretokenization involves the segmentation of textual content by means of the pretokenizer, which proceeds to partition the given data into discrete objects, or words that later form the ultimate set of tokens. Since we are using English and Russian languages, the strings can be divided into words by spaces and punctuation marks.

D. *Tokenization*

The sentences are broken down into a single word or token at the next stage. *Word tokenization* is a method for tokenization that we have used in our project. It is a critical step in the processing of natural languages, and it's often used as an early stage for applications such as MT. The input text is split into tokens based on the whitespace between words. This means that the whole word of the sentence is represented by each token. Some word tokenizers have extra rules for punctuation. One can also apply *stemming* or *lemmatization*, which normalizes words to their stem (e.g., "great", "greater", and "greatest" all become "great"), at the expense of losing some information in the text [10].

E. *Padding*

During the tokenization process, the length of each sentence was obtained and represented as a "ragged tensor". One method of dealing with variable length sequences is to append filler data that helps ensure the consistency of all sequence lengths. Due to the variability in length, the implementation of padding results in the creation of uniform sentence length within a "tensor object". The padding function is called using "pad_sequences" command. It requires three parameters: x, a list of sequences, length, the length to pad too, and a NumPy array containing padded sequences [9].

Padding for machine learning algorithms requiring long linear inputs, such as a large number of neural networks used to read the text, is considered a necessary step. While padding is essential to ensure consistency and standardisation in the data set, it is also required when inserting new information into sequences.

A demonstration of all the above steps applied on applications can be seen in Fig. 1.

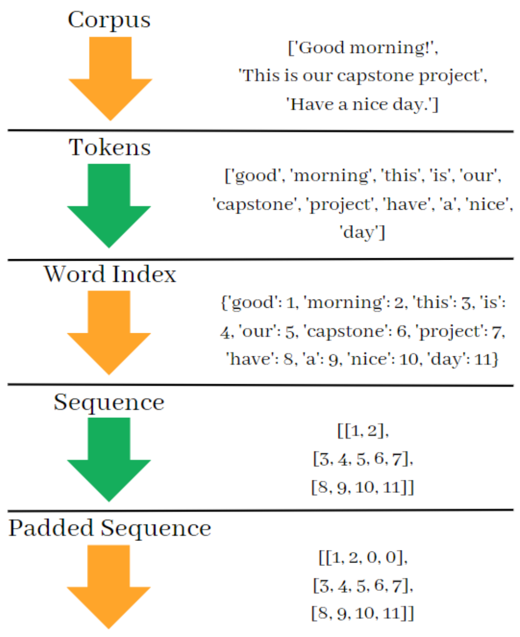
Corpus
['Good morning!',
'This is our capstone project',
'Have a nice day.']
Tokens
['good', 'morning', 'this', 'is', 'our',
'capstone', 'project', 'have', 'a', 'nice',
'day']
Word Index
{'good': 1, 'morning': 2, 'this': 3, 'is':
4, 'our': 5, 'capstone': 6, 'project': 7,
'have': 8, 'a': 9, 'nice': 10, 'day': 11}
Sequence
[[1, 2],
[3, 4, 5, 6, 7],
[8, 9, 10, 11]]
Padded Sequence
[[1, 2, 0, 0],
[3, 4, 5, 6, 7],
[8, 9, 10, 11]]

Fig. 1: Illustration of padding on a sample corpus

IV. Model Architecture

The Transformer system's architecture is based on the encoder-decoder paradigm, which is trained from beginning to end. The model has already shown in the article [11], that in terms of complexity it dominates under the RNN and CNN. The main methods spin around parallel multiplication of vectors, preventing vanishing gradients, accessing information across an entire sentence, which prevents the problem of global information access.

A. *Transformer*

The task of a transformer in a machine translation system is to convert an input sequence of text in context language into a corresponding sequence of text in target language. Transformer is composed of encoder and decoder parts. Each of which plays a big role when processing a sequence and accurately conveying the meaning of the input context sequence in the output target sequence.

During the encoding phase, the transformer takes the input sequence and maps it to a sequence of hidden states that capture the information contained in the input. These hidden states are then used by the decoder to generate the output sequence, which produces one token at a time based on the previous tokens generated and the information in the hidden states [8].

B. *Attention*

In an attention mechanism, there are three crucial concepts namely query, key and value. It would be more appropriate if these concepts in self-attention mechanism are unwrapped with the note below, because the Transformer architecture takes its advantages solely from the attention mechanism, eliminating the need for recurrent or convolutional networks.

Given a set of vectors $\{x_i\}_{i=1}^{t}$, where $x_i \in \mathbb{R}^d$, we obtain $\{q_i\}_{i=1}^{t} = Q$, $\{k_i\}_{i=1}^{t} = K$, and $\{v_i\}_{i=1}^{t} = V$ by performing linear transformations on the input matrix $X \in \mathbb{R}^{t \times d}$ as follows: $X_{t \times d} W_{d \times d}^{q}$, $X_{t \times d} W_{d \times d}^{k}$, and $X_{t \times d} W_{d \times d}^{v}$. Thus, Q, K, and V are of size $R^{t \times d}$.

To calculate the attention scores, we compute the dot product of Q and the transpose of K: $QK^T = S \in \mathbb{R}^{t \times t}$. By applying the softmax function to the attention scores, we obtain a softmax distribution of attention weights: $A_{t \times t} = softmax(S)$. Finally, to update the input X, the attention weights are multiplied with the values V: $\hat{X} = A_{t \times t} V_{t \times d}$.

The self-attention mechanism is specifically designed to allow each position x_i to attend to all other positions, including itself, within the input set $x_{i}{}_{i=1}^{t}$. This allows the model to capture relevant information for each input entry. Eq. (1) represents a compact notation derived from the original paper [11], describing the process of self-attention.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

The Multi-head attention is h number of attention mechanism $h >= 2$, that takes an input $X_{t\times d}$ and for each head the $\bar{X}_{t\times \frac{d}{h}}$ will be distributed and fed. After feeding in Multi-head attention outputs are concatenated into $t \times d$ dimensional matrix as in initial time before feeding. Eq. (2) from [11] shows what is being said:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

where each $head_i = Attention(QW_i{}^Q, KW_i{}^K, VW_i{}^V)$, with parameter matrices $W_i{}^Q, W_i{}^K, W_i{}^V$ and W^0. A purpose of Multi-head attention is to allow the model to capture different types of information and relationships in the input sentence simultaneously.

C. Encoder-Decoder

The encoder and decoder are key components of a MT model. The encoder typically consists of a neural network that takes the context input sentence as a sequence of vectors and passes it through a series of layers. First layer we encounter is an Multi-Head Self-Attention layer Eq. (2), which processes a sequence of vectors and represents a new same sized vectors. After each sublayer, layer normalization is applied. It is used mainly for preventing from noises added by shifting inputs from earlier encoder blocks. Following layer normalization, there is a skip/residual connection around each of the two sub-layers. A reason for the last one is that a depth leads to earlier information being ”forgotten” over blocks, and vanishing gradients [11].

Each layer processes the context input sequence and adds the information so that the information or meaning is stored in a new representation of the sequence, which passes to the next layer. The final output of the encoder is a summary of useful data, of the same size as the initial sequence of vectors, passed through all the layers and blocks.

The decoder adds a third sub-layer to each encoder layer in addition to the two already there, performing multi-head attention over the encoder stack's output.

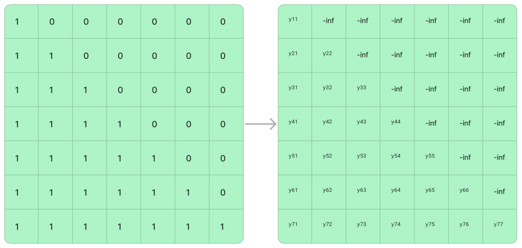
1 0 0 0 0 0 0
1 1 0 0 0 0 0
1 1 1 0 0 0 0
1 1 1 1 0 0 0
1 1 1 1 1 0 0
1 1 1 1 1 1 0
1 1 1 1 1 1 1
y11 -inf -inf -inf -inf -inf -inf
y21 y22 -inf -inf -inf -inf -inf
y31 y32 y33 -inf -inf -inf -inf
y41 y42 y43 y44 -inf -inf -inf
y51 y52 y53 y54 y55 -inf -inf
y61 y62 y63 y64 y65 y66 -inf
y71 y72 y73 y74 y75 y76 y77

Fig. 2: Attention masking

The decoder's first layer as shown in Fig. 3 takes a target language sequence and feeds it with Causal Multi-Head Attention with masking of only y_i target vector and its previous $\{y_j\}_{j=1}^{i}$ vectors at given time and for next positions its masking is set to $-\inf$. So that the model is freed from cheating and tries to learn and predict based on previous predictions the next word. As an example of masking in Fig. 2.

The decoder's second layer, namely Cross Multi-Head attention, on the other hand, takes the encoded representation of the source language sentence as input and generates the target language sentence. Unlike in Multi-Head Self-Attention, where key, value, and query are linear transformations of X itself, in Cross Multi-Head attention K, V come from encoder's output and query Q is an target sequence. The target query sequence tries to generate attention scores for each position in the encoder's key sequence. The key and value sequences provide the information to attend to. Same as in encoder block there are residual connections around each of the sub-layers, and then layer normalizations [11].

D. Positional Encoding

The Transformer architecture lacks of information about positions of sequence entries, so it introduces an approach of storing information about the relative or absolute location of the tokens in the sequence. Locations are calculated in an original paper [11] with the expressions in Eq-s (3), (4). The Positions are calculated and added to the embedding vectors of each token, so embedding vectors implicitly hold information about location. In this work, we held this approach:

$$PE_(pos, 2i) = \sin \frac{pos}{10000^{\frac{2i}{d_{model}}}} \tag{3}$$

$$PE_(pos, 2i+1) = \cos \frac{pos}{10000^{\frac{2i}{d_{model}}}} \tag{4}$$

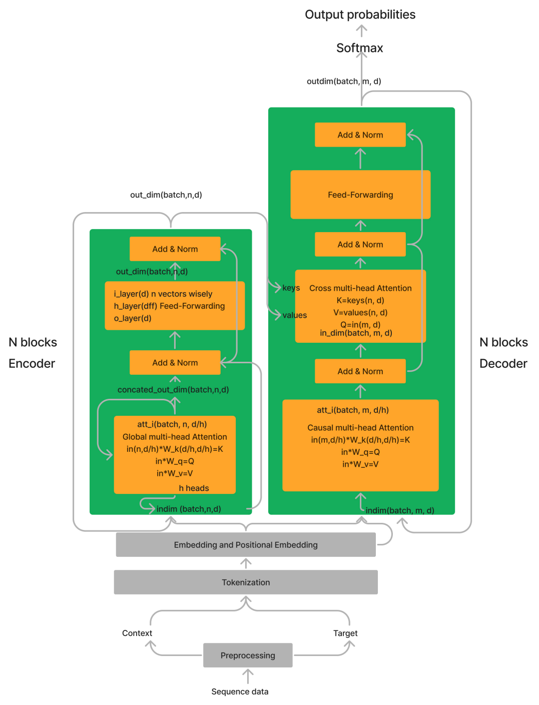

Fig. 3: The Transformer Architecture [11]

V. Exploration

A. Performance enhancement

In order to handle the processing of large-scale data corpus within the constraints of our computer resources, we conducted experiments and obtained results to derive meaningful insights even though, the results of machine translation are very low. To manage the computational load, we initially worked with a limited amount of data corpus. Our approach involved removing sentence pairs that exceeded a fixed length ratio above a certain threshold. Thereby filtering out sentence pairs with imbalanced lengths. Furthermore, we implemented additional measures to ensure the quality and efficiency of the data. We excluded sentence pairs with excessively short sentences, requiring a minimum of 4 words for each sentence. This criterion allowed us to focus on sentences with sufficient context and linguistic information. Additionally, we enforced a maximum sentence length of 100 words across all datasets to prevent potential computational bottlenecks caused by exceptionally long sentences. Upon analysing the parallel data, we found that some sentences were in wrong language. To address this issue, we performed filtering the entire data corpus. This step aimed to identify and eliminate any additional sentences that contain text in wrong language. So we hold pureness of text in its language.

B. Linguistic analysis

When working with machine translation, specifically the process of translating from a source language to a target language, the occurrence of linguistic issues is often noted. In the present study, it is discernible that there exist distinctive syntactic and semantic variations between the two languages, alongside divergences in grammar and vocabulary. English follows a subject-verb-object word order, whereas Russian follows a more flexible subject-object-verb word order. Moreover, Russian is known for its rich system of grammatical cases, where nouns, pronouns, and adjectives change their forms based on their grammatical roles in a sentence. English, on the other hand, relies more on word order and prepositions to convey similar meanings.

Using the BLEU score, we assessed the quality of the translated text. Also, for comparison, we used the results of the word and subword tokenization approaches' scores, which can be seen in the Tables I and II.

VI. Training and Evaluation

A. Optimizer

In order to update the parameters of the model properly we used an optimization algorithm called *Adam*. Below is an expressions used behind the scenes of the Tensorflow API.

$$v_t = \beta_1 * v_{t-1} - (1-\beta_1) * g_t \tag{5}$$

$$s_t = \beta_2 * s_{t-1} - (1-\beta_2) * g_t^2 \tag{6}$$

$$\Delta\omega = -\eta \frac{v_t}{\sqrt{s_t + \epsilon}} * g_t \tag{7}$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t \tag{8}$$

The Adam optimizer is an optimization algorithm used to update the parameters of a machine learning model during training. It is commonly used in deep learning and is known for its efficiency and effectiveness in optimizing neural networks.

B. BLEU

Bilingual Evaluation Understudy is a widely-used metric in the field of machine translation for evaluating the quality of translated text. It provides an objective measure of the correspondence between a machine-generated translation and one or more reference translations.

The BLEU score is based on the concept of n-gram precision, which measures the overlap of n-gram sequences between the candidate translation and the reference translations. By considering multiple n-gram orders, BLEU captures both the adequacy and fluency of the translation.

To compute the BLEU score, the candidate translation is compared to a set of reference translations. The n-gram precision is calculated by counting the number of matching n-grams in the candidate translation and dividing it by the total number of n-grams in the candidate translation. The precision values for different n-gram orders are then combined using a weighted average. Finally, a brevity penalty is applied to account for translations that are shorter than the reference translations. BLEU scores range from 0 to 1, with higher scores indicating better translation quality [12].

C. *ROUGE-L*

Similar to BLEU, the Recall-Oriented Understudy for Gisting Evaluation is a ubiquitously used metric employed to assess the quality of summarization systems. Its purpose is to evaluate the degree of overlap between the candidate summary and multiple reference summaries by calculating the longest common subsequence shared between them.

The primary focus of the ROUGE-L score lies in measuring the recall, which captures the candidate summary's ability to encompass the essential information contained within the reference summaries. This is accomplished by determining the Longest Common Subsequence between the candidate and reference summaries and subsequently dividing it by the total number of words present in the reference summaries. The resulting recall value serves as an indication of the coverage achieved by the candidate summary in relation to the reference summaries.

ROUGE-L scores are scaled from 0 to 1. Similar to BLEU, the ROUGE-L metric operates on a reference-based basis, necessitating the availability of reference summaries for accurate evaluation. Its purpose is to provide an assessment of the candidate summary's informativeness and coherence compared to the reference summaries [2].

D. *Experiments*

Table I: Word tokenization

Model	Scores	
	BLEU	**ROUGE-L**
Transformer	0.48	0.35

Table II: Subword tokenization

Model	Scores	
	BLEU	**ROUGE-L**
Transformer	0.65	0.57

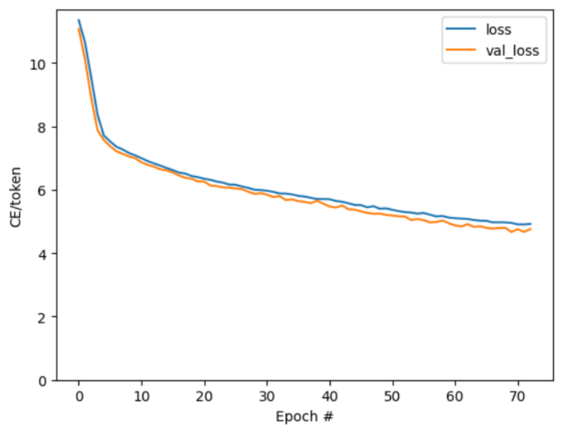

Fig. 4: Performance in loss

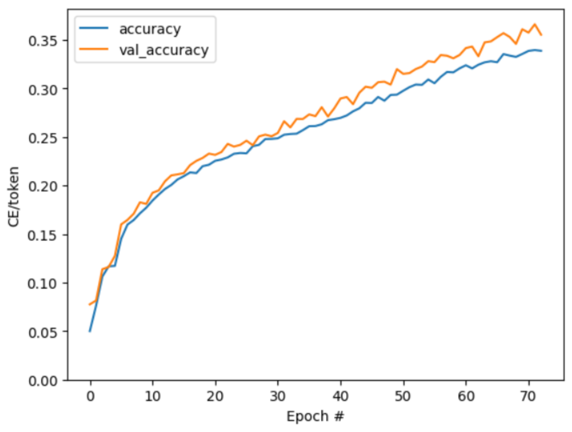

Fig. 5: Performance in accuracy

E. *Visualizations*

During the process of training the model, significant enhancements can be observed as evidenced by the decreasing trend of the loss index (see Fig. 4) and the increase in the accuracy metric (see Fig. 5). Such indicators highlight the model's proficiency and aptitude to execute our intended task effectively.

F. *Parameters*

The hypermaraters and amounts of our models are listed in Table III and Table IV.

Table III: Table of parameters

Parameters	Transformer
loss	Sparse-Categorical-Crossentropy
optimizer	ADAM
learning-rate	Dinamically Allocated

VII. CHALLENGES

Our investigation into this subject matter has encountered a myriad of formidable challenges that have had a

Table IV: Table of amounts

Amounts	Transformer
num-unique-words	500000
max-seq-len	100
vector-units	512
epochs	100
num-of-parallel-sentences	1000000
num-of-blocks	6
num-of-encoder-layers	6
num-of-heads for each attention type	8

profound impact on the breadth and depth of our research efforts. In this section, we elaborate on several significant hurdles we faced, shedding light on the complexities involved.

A. *Challenges specific to language pairs*

The linguistic idiosyncrasies inherent to the Russian (Cyrillic) and English (Latin) languages have presented unique challenges, necessitating careful consideration and strategic approaches. One of the primary difficulties we encountered was the misrendering of specific letters and punctuation marks during the translation process. The complex nature of these linguistic variations demanded a meticulous and nuanced approach to address the issue effectively.

To tackle these challenges, we implemented the technique of normalization, a multi-faceted process aimed at aligning and standardizing the linguistic features of both languages. This comprehensive normalization procedure involved various intricate steps, meticulously detailed in the dedicated section on data preparation and preprocessing.

The process encompassed the elimination of replicated content, removal of unsuitable sentence pairs, conversion to lowercase letters, and transformation into romanized form. Additionally, we employed sophisticated methodologies to resolve issues related to sentence alignment, ensuring coherency and accuracy in the translation outputs.

B. *Model complexity*

The complexity of the NMT model posed a significant challenge throughout our research, warranting meticulous attention to detail and demanding an extensive allocation of computational resources. The training and testing of the NMT model necessitated a significant allocation of computing power, especially through the utilization of advanced GPUs and CPUs. These state-of-the-art hardware components played a pivotal role in attaining the modest level of performance and accuracy in translation. Given the computationally intensive nature of the task, we were aware that achieving optimal results would require substantial computational resources or even beyond our power. However, recognizing the inherent value and importance of our research, we strived to leverage and push the boundaries of the available resources to their maximum threshold. By employing feasible GPUs and CPUs, we aimed to extract every ounce of computational power available to us, ensuring that our research efforts were not impeded by technological limitations. This approach allowed us to explore the capabilities and potential of the NMT model to their fullest extent, maximizing the research value and impact of our study. While the computational expense associated with our project was undeniable, we were committed to doing our utmost to overcome these challenges. We recognized that settling for suboptimal computational resources would have undermined the overall value and significance of our research. Hence, we made every effort to optimize the use of available resources and push the boundaries of computational capacity to ensure that our study yielded meaningful and impactful results. Although the training and testing of the NMT model demanded substantial computing power, we endeavored to maximize the available threshold in order to deliver a research study to a contributory direction.

Moreover, due to the extensive size of our dataset, comprising one million comparable sentence pairs, the evaluation of the model demanded substantial time-frames, spanning several hours. This temporal challenge necessitated patience and careful planning in order to accommodate the necessary evaluation period.

Furthermore, the implementation of subword tokenization, a technique used to partition lexical units into smaller components, added another layer of complexity to the data preprocessing stage. This intricate process required in-depth analysis and expertise to ensure optimal utilization of subword units and maximize translation quality.

Additionally, the complexity of the NMT model created challenges in terms of parameter tuning. The model comprises numerous crucial hyperparameters that must be appropriately adjusted to optimize its performance. Fine-tuning these parameters proved to be a time-consuming and labor-intensive task, requiring iterative experimentation to find the most effective settings.

VIII. Conclusion

In conclusion, our study has demonstrated that the integration of Neural Machine Translation, leveraging advanced Transformer-based architectures, has led to a significant enhancement in translation accuracy when applied to the English-to-Russian language pair. By employing a robust neural network architecture, we were able to capture and model the intricate interdependencies among words, entities, and sentence structures, resulting in more faithful and accurate translations.

The meticulous data preprocessing and postprocessing procedures implemented during the preparation and cleansing of our dataset have proven instrumental in eliminating noise and optimizing the accuracy and fidelity of the translation outputs.

Despite facing temporal constraints and technological limitations, we have achieved satisfactory results given bottlenecks that we have faced. Our team has acquired invaluable knowledge in Natural Language Processing including tackling sequence-to-sequence tasks, understanding Transformer architecture and Attention mechanism and expertise in the realm of data processing techniques and experimental design throughout the course of our experimentation, which will undoubtedly contribute to the advancement of future research endeavors in this or adjacent fields. The insights gained from this study will serve as a foundation for further exploration and development in the exciting domain of Neural Machine Translation.

References

[1] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Software available from tensorflow.org. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[2] Clamp;eacute;ment Brutti-Mairesse. *Rouge and BLEU scores for NLP model evaluation*. Dec. 2021. URL: https://clementbm.github.io/theory/2021/12/23/rouge-bleu-scores.

[3] Mia Xu Chen et al. "The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 76–86. DOI: 10.18653/v1/P18-1008. URL: https://aclanthology.org/P18-1008.

[4] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[5] Kai Jiang and Xi Lu. "Natural Language Processing and Its Applications in Machine Translation: A Diachronic Review". In: *2020 IEEE 3rd International Conference of Safe Production and Informatization (IICSPI)*. 2020, pp. 210–214. DOI: 10.1109/IICSPI51290.2020.9332458.

[6] Huey-Ing Liu and Wei-Lin Chen. "Re-Transformer: A Self-Attention Based Model for Machine Translation". In: *Procedia Computer Science* 189 (2021). AI in Computational Linguistics, pp. 3–10. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.05.065. URL: https://www.sciencedirect.com/science/article/pii/S1877050921011509.

[7] Venkata Sai Rishita Middi, Middi Raju, and Tanvir Ahmed Harris. "Machine translation using natural language processing". In: *MATEC Web of Conferences* 277 (Jan. 2019), p. 02004. DOI: 10.1051/matecconf/201927702004.

[8] Alessandro Raganato and Jörg Tiedemann. "An analysis of encoder representations in Transformer-based machine translation". In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (2018). DOI: 10.18653/v1/w18-5431.

[9] Basu Dev Shivahare et al. "Survey Paper: Study of Sentiment Analysis and Machine Translation using Natural Language Processing and its Applications". In: *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)*. 2022, pp. 652–656. DOI: 10.1109/ICIEM54221.2022.9853044.

[10] L. Tunstall, L. von Werra, and T. Wolf. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, Incorporated, 2022. ISBN: 9781098103248. URL: https://books.google.com/books?id=pNBpzwEACAAJ.

[11] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[12] Haifeng Wang et al. "Progress in Machine Translation". In: *Engineering* 18 (2022), pp. 143–153. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2021.03.023. URL: https://www.sciencedirect.com/science/article/pii/S2095809921002745.

[13] Wei Wang et al. "Denoising Neural Machine Translation Training with Trusted Data and Online Data Selection". In: *CoRR* abs/1809.00068 (2018). arXiv: 1809.00068. URL: http://arxiv.org/abs/1809.00068.

[14] Linting Xue et al. "mT5: A massively multilingual pre-trained text-to-text transformer". In: *CoRR* abs/2010.11934 (2020). arXiv: 2010.11934. URL: https://arxiv.org/abs/2010.11934.

[15] Jiacheng Yang et al. "Towards Making the Most of BERT in Neural Machine Translation". In: *CoRR* abs/1908.05672 (2019). arXiv: 1908.05672. URL: http://arxiv.org/abs/1908.05672.