THE USA JOURNALS
THE AMERICAN JOURNAL OF ENGINEERING AND TECHNOLOGY (ISSN – 2689-0984)
VOLUME 06 ISSUE09

PUBLISHED DATE: - 12-09-2024
DOI: - https://doi.org/10.37547/tajet/Volume06Issue09-07 **PAGE NO.: - 45-55**

RESEARCH ARTICLE **Open Access**

HIERARCHICAL ENCODING AND CONDITIONAL ATTENTION IN NEURAL MACHINE TRANSLATION

Natalia Trankova
MSc - Skolkovo Institute of Science and Technology, New York, 10280, USA

Dmitrii Rykunov
BSc – National Research University Higher School of Economics, New York, 10013, USA

Ivan Serov
McKinsey & Company, Data Science division, New York, 10007, USA

Ivan Giganov
MSc - Northwestern University, Chicago, IL, 60654, USA

Yaroslav Starukhin
QuantumBlack, AI by McKinsey, Boston, MA 02110 USA

Abstract
The advent of Transformer models has significantly advanced Neural Machine Translation (NMT), particularly in sequence-to-sequence tasks, yet challenges remain in maintaining coherence and meaning across longer texts due to the model's traditional focus on independent phrase translation. This study addresses these limitations by proposing an enhanced NMT framework that integrates cross-sentence context through redesigned positional encoding, hierarchical encoding, and conditional attention mechanisms. The research critiques the shortcomings of existing positional encoding methods in capturing discourse-level context, introducing a novel hierarchical strategy that preserves structural and semantic relationships between sentences within a document. By employing a source2token self-attention mechanism to encode sentences and a conditional attention mechanism to selectively aggregate the most relevant context, the proposed model aims to improve translation accuracy and consistency while reducing computational complexity. The findings demonstrate that this approach not only enhances the quality of translations but also mitigates the computational costs typically associated with processing longer sequences. However, the model's effectiveness is contingent on the presence of clear document structure, which may limit its applicability in more irregular texts. The study's contributions offer significant implications for the development of more contextually aware and computationally efficient NMT systems, with potential applications in domains requiring high fidelity in translation, such as legal and academic fields. The proposed methods pave the way for future research into further optimization of context integration in NMT and exploring its application in multilingual and specialized domain contexts. Limitations include the additional computational overhead introduced by the hierarchical and conditional attention mechanisms, which may affect performance in low-resource environments. Nonetheless, this work represents a substantial step forward in addressing the complexities of document-level translation.

Keywords Neural Machine Translation (NMT), Transformer Model, Cross-Sentence Context, Positional Encoding, Hierarchical Encoding, Conditional Attention, Document-Level Translation, Context-Aware Translation, Discourse-Level Context.

INTRODUCTION

The creation of the Transformer models, which serve as the basis for sequence-to-sequence tasks, has led to notable advances in Neural Machine Translation (NMT). Introduced by Vaswani et al. [1] in 2017, the vanilla Transformer model has found extensive use in natural language processing (NLP). The model handles the links between tokens inside a sequence using self-attention instead of recurrence or convolution. The primary focus of this design is on translating phrases independently, which might present difficulties when translating longer texts because cross-sentence context is necessary to uphold coherence, clear out ambiguities, and preserve the original meaning.

This study primary focus is to address these challenges and improve the quality and consistency of NMT by maintaining the structural and semantic connections between sentences inside a document. Instead of processing each sentence separately, redesigning of the positional encoding mechanism in processing sequences allows it to span numerous sentences.

In order to capture sentence-level dependencies, this paper discusses the shortcomings of current positional encoding approaches in discourse-level contexts, suggests a hierarchical encoding strategy, and presents a novel conditional attention mechanism that allows relevant context to be selected from the most relevant sentences within a document. By enhancing NMT systems' ability to manage lengthier sequences with intricate inter-sentential interactions, these contributions hope to produce translations that are more precise and sensitive to context.

1 POSITIONAL ENCODING

1.1 Motivation for Positional Encoding

The vanilla Transformer model contains no recurrence and no convolution, so when attention used in an unrestricted manner (attention being performed over the whole sequence) the model does not have information about the relative or absolute position of the tokens in the sequence. It could be said that the model operates on a bag-of-tokens derived from the initial sequence. To provide the model with information about the relative and absolute position of the tokens the original paper suggests to use positional encoding. That is an additional embedding of tokens based solely on their position in the sequence that is added to the tokens' original embedding.

1.1.1 Is Positional Encoding Needed at All?

There is no known empirical study of the importance of the positional encoding mechanism for the performance of the Transformer model. However, it could be argued that the results for Convolutional S2S model [2] is relatable to some extent to the Transformer model since it is also a no recurrence model. The Convolutional S2S model was shown to perform well without any positional embedding at all [2]. It was also shown that a vanilla encoder-decoder RNN model could benefit significantly from a refinement of word embeddings with the source sentence's bag-of-words representation [3]. These results suggest that the knowledge of the position of tokens might be of little importance for NMT systems processing texts in the sentence-by-sentence fashion. The order of words in output translation is preserved by the auto-regressive manner of the decoder and in most cases words' meaning could be disambiguated just by the bag-of-words

representation of the source sentence. Now let's consider the case of processing the whole document with the Transformer at once by simple concatenation of sentences together. It is obvious that examination of the sequence (that consists of every word in the document) in a bag-of-words manner would be of no use in the task of disambiguation of a word's meaning if that word used in the document in several meanings. The same logic is applicable to the challenges of cohesion and coherence maintenance (e.g. anaphora resolution). Therefore, a way to condition the attention scores by the relative position of the tokens is needed.

1.2 Redesigning Positional Encoding

Since there is no known prior work on positional encoding with respect to a discourse-level context, the implementation could be suggested based on the various assumptions regarding the context.

Definition 1.1. Structural unit is a part of a text that could be naturally derived from the source text's organization. Examples of structural units are collections, articles, chapters, topics, paragraphs, and sentences.

Assumption 1. Words ordering inside a sentence contains useful information for translation.

Assumption 2. Ordering of structural units in a text contains useful information for translation. (e.g. sentences ordering is important)

Assumption 3. The relevance of parts of one structural unit to themselves tends to be higher than to the parts of another structural unit. (Local relevance depends on boundaries of structural units. For example, words in the same sentence tend to be more relevant to each other than the words from another sentence.)

To utilize all three assumptions stated above it is sufficient for a positional encoding to encode for each token in a sequence its absolute positions with respect to each structural level starting from the beginning of the sequence. So, for example, if words, sentences, and paragraphs could be distinguished in a text, then the 135th word of the text contained in the 4th sentence of the 2nd paragraph would be encoded with PE135,4,2. Such encodings could be learned together with the model [2].

1.3 Existing Approaches Applicability

1.3.1 Sentence Delimiters.

[4] suggests several context fusion strategies of which the best-performing one is the concatenation of the previous source sentence to the sentence being processed with a special sentence-break token between them. Authors of the paper demonstrate that such approach significantly improves the overall translation quality of the encoder-decoder RNN model. This approach seems to be inapplicable to the Transformer model as is, because the Transformer model uses no recurrence and therefore process sequences as bag-of-tokens. With this aspect in mind, it could be seen that the sentence-break tokens used on its own give the model only the information on the number of sentences in the sequence. It does not give the model enough information to determine to which sentence a word belongs to.

If this method is used in conjunction with the plain one-level positional encoding of words as in the original model but applied to the whole document at once, it still provides attention layers with no useful information per se. Keeping in mind the fact that the Transformer model on each iteration process input sequence as bag-of-tokens it could be noted that the presence of the sentence-break tokens in the bag-of-tokens does not help the model to distinguish words from different sentences. However, this time it could be assumed that in the best case if it is needed for the model to distinguish sentences attention layer could enrich

the tokens' encoding with the sentence-belonging information (based on their relative position to the sentence-break tokens) to be used by the following attention layers. In this case, the result is identical to the result of the multi-level positional encoding proposed in Section 1.2, however, it comes at a cost of computation of a single attention layer (which is considered to be more expensive than computing the positional encoding) and could be unstable in terms of the result.

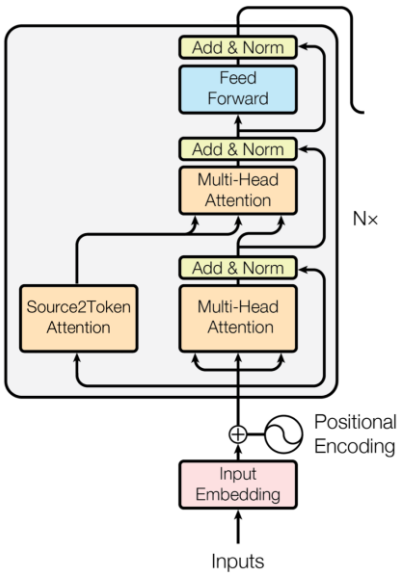

Fig. 1. Encoder stack extended with Source2Token sentence encoding block and additional Multi-Head Attention block [1].

2 HIERARCHICAL ENCODING OF CONTEXT SENTENCES

The original proposal to feed the context to the Transformer model by processing the whole document as a single sequence is computationally expensive because the Self-Attention layer's computational complexity scales quadratically with respect to the sequence length. So to mitigate the overall computational complexity it is proposed to substitute the full-length sentences in the key and value matrices of the Attention layer with sentences' encoding vectors. Sentence encodings could be calculated with source2token self-attention [5].

Following the notation of [1], for a sentence j with m words $\{w_1, \dots, w_m\}$, where each word is represented with an embedding vector of dimensionality d_{model}, the $source2token$ sentence's embedding $s_j \in R^{d_{model}}$ is calculated with a scaled dotproduct attention block.

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

$$s_j = Source2Token(S_j)$$

$$Source2Token(S) = Attention(q, SW^K, SW^V)W^O \#(2)$$

Where $S_j \in R^{m \times d_{model}}$ is a matrix of word embeddings of the sentence j, and learnable parameters are $q \in R^{d_k}, W^K \in R^{d_{model} \times d_\vartheta}, W^O \in R^{d_\vartheta \times d_{model}}$.

Then it is proposed to collect all sentence encodings to form K and V matrices for the Multi-Head Attention blocks of the Transformer. It is thought that the direct information flow from the words in the source sentence is crucial for the translation. Because of this to preserve the original Attention structure the model could be extended with additional Multi-Head Attention block in the encoder and decoder stacks which is fed with K and V matrices constructed of source2token sentence embeddings. It could also be done with additional heads in the existing Multi-Head Attention blocks. A scheme of the Transformer's encoder stack extended with the additional Multi-Head Attention block is provided in Figure 1. A model with this architecture could be trained end-to-end on the translation task. As K and V matrices in the additional Multi-Head Attention block are shared among different sentences and words within sentences, queries in this block could be stacked together as they are stacked in the previous Multi-Head Attention block.

3 CONDITIONAL CONTEXT AGGREGATION

3.1 Motivation

To include the discourse-level context in the Transformer model it is proposed to process the whole document as a single sequence (please refer to section 1 of this paper). However, since the computational complexity of Self-Attention layer scales quadratically with respect to the sequence's length [1] it could be computationally infeasible on longer documents.

To mitigate the higher computational complexity of processing the whole document at once it was proposed to encode each sentence with a single vector representation and to perform Self-Attention over these sentence-vector representations. This approach has two drawbacks.

1. Word-level precision is lost for attention due to the aggregation.
2. It still scales quadratically with respect to the number of sentences in a document.

To counteract these drawbacks a new assumption regarding the context structure is needed.

Assumption 4. For each word in a document, there are only a few sentences in the same document that are needed to correctly translate it.

With this new assumption, it is suggested that for each word being encoded with Self-Attention it is sufficient to consider words from only the top T most relevant sentences. To select top T most relevant sentences for a given word a similarity function (e.g. dot-product) could be calculated between the linearly transformed word vector and sentences' sorce2token encodings.

3.2 Aggregating Words from The Most Relevant Sentences

The high-level outline of the approach is the following.

1. Encode each sentence in a document with a single vector using source2token self-attention.
2. For each word in the document:

a. Calculate the relevance score between this word and every sentence in the document.

b. Select top T most relevant sentences in the document.

c. Transform its embedding through Attention over words from the top T most relevant sentences.

Relevance function is defined as a scaled dot-

product between a query and keys.

$$Relevance(Q,K) = \frac{QK^T}{\sqrt{d_k}} \#(3)$$

KeepTopT function is defined identically to [6] with the notation adopted (k→t) to prevent overlap with the one currently being used.

$$\boldsymbol{KeepTopT(\vartheta,t)_i} = \{\boldsymbol{\vartheta_i \;\; if \;\; \vartheta_i \; is \; in \; the \; top \; t \; elements \; of \; \vartheta} \; \boldsymbol{-\infty \;\; otherwise}$$

To allow the gradient flow to the relevance computation block through a Self-Attention block over words from selected top T sentences it is proposed to augment the attention scores of selected words with an addition of the relevance score of their sentences (based on which they were selected) before Softmax application. In this case, constructing the K and V input matrices for the Attention block (Equation 1) with words only from selected top T sentences is mathematically identical in terms of the final result to placing all words of the document in the K and V matrices and adding the corresponding relevance scores to the attention scores before Softmax (utilizing the fact that the relevance scores for the irrelevant sentences equal to minus-infinity). Of course, in practice it is supposed to calculate attention scores only for the words from top T selected sentences.

Bringing it all together, an Attention head j in the Multi-Head Attention block in the encoder stack of the original Transformer model could be redefined as follows to incorporate conditional attention over the whole document. It is assumed that there is a document with n sentences with m words in each.

$$S = [Source2Token(S_1) \; \vdots \; Source2Token(S_n)\;]_{n\times d_{model}} where \;\; S_i = X[(i-1)m+1 : im, :] \#(5)$$

$$M = \begin{bmatrix} \gamma_{0,0,0}\,\gamma_{0,0,1}\,\gamma_{1,0,0}\,\gamma_{1,0,1} & \cdots\, \gamma_{0,n,m} & \cdots\, \gamma_{1,n,m} \\ \vdots\vdots & \gamma_{n,0,0}\,\gamma_{n,0,1} & \ddots\vdots\cdots\, \gamma_{n,n,m} \end{bmatrix}_{n\times d_{model}} where \gamma_{i_1,i_2,i_3} = \{1, if \; i_1 equals \; to \; i_2 \; 0, otherwise \; \#(6)$$

$$ConditionalAttention_j(X) = Softmax\big[Relevance\big(XW_j^{Q_X}, XW_j^{K_X}\big) + KeepTopT\big[Relevance\big(XW_j^{Q_S}, SW_j^{K_S}\big), t\big]M\;\big]\big(XW_j^{V_X}\big) \#(7)$$

Where $S \in R^{n\times d_{model}}$ is a matrix containing all sentences' encodings; $M \in R^{n\times nm}$ is an auxiliary matrix mapping of sentences to words; $X \in R^{nm\times d_{model}}$ is the input matrix of words' embeddings; Softmax and KeepTopT functions are applied to the input matrices row-by-row; $W_j^{Q_X}, W_j^{K_X}, W_j^{Q_S}, W_j^{K_S} \in R^{d_{model}\times d_k}, W_j^{V_X} \in R^{d_{model}\times d_\vartheta}$ are learnable parameters.

Attention head in the Multi-Head Attention block of the encoder in the original Transformer model could be substituted with the *ConditionalAttention* block directly. The resulting model could be trained end-to-end with t > 1. It has been shown that this occasionally-sensitive behavior of a gating unit is enough for end-to-end training [7] [6].

Table 1. Computational complexity for different layer types. n is the number of sentences, m is the average number of words in each sentence.

Layer Type	Complexity per Layer
Self-Attention (vanilla — sentence-by-sentence — no context)	$O(n \cdot m^2)$
Self-Attention (over the whole document at once)	$O([n \cdot m]^2) = O(n^2 \cdot m^2)$
Self-Attention (over the sentence encodings)	$O(n^2 \cdot m + \text{encodings})$
Conditional Self-Attention (over top t most relevant sentences)	$O(n^2 \cdot m + n \cdot t \cdot m^2 + \text{encodings})$
Hierarchical Conditional Self-Attention (over top t most relevant sentences)	$O(n \cdot m \cdot t \cdot [\log n + m] + \text{encodings})$
Source2Token Self-Attention (generating encodings for each sentence)	$O(n \cdot m)$

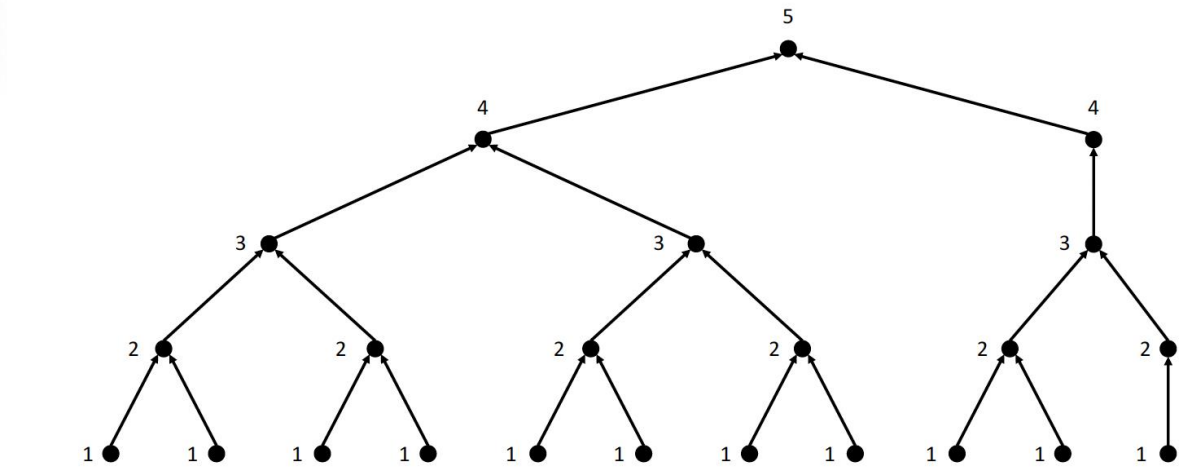

Fig. 2. Binary tree representation of depth 5 of a document consisting of eleven sentences.

Circles represent encodings; arrows represent the flow of computation; numbers denote the order of nodes generation. The circles denoted by 1 represent Source2Token encodings of sentences in the document. The circle denoted by 5 represents the root encoding of the document that is encoding the whole content of the document in a single vector.

3.3 Hierarchical Sentence Selection

As it was noted in the section 3.1 the computational complexity of calculations of the attention scores over all sentences' encodings scales quadratically with respect to the number of sentences. It is used to select the most relevant sentences from the document for a given word. To further reduce the computational complexity of the ConditionalAttention block it is proposed to construct a binary tree representation of the document's content to search over it for relevant sentences. To navigate and branch computations over the tree the Relevance (Equation 3) and KeepTopT (Equation 4) functions are used

respectively. To allow a gradient-flow from the resulting embedding transformation through the tree to the gating units the cumulative relevance score is propagated from top to bottom of the tree and added to the attention scores of the words of the selected sentences before the Softmax activation.

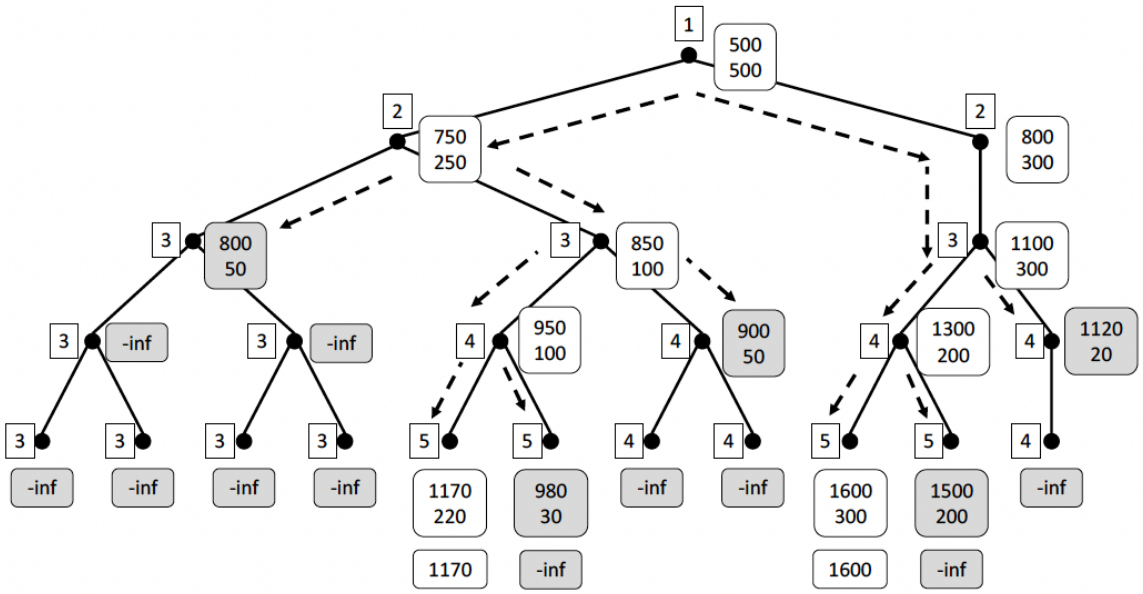

Fig. 3. Illustration of the Traverse Tree procedure keeping top 2 relevant sentences applied to a binary tree representation of depth 5 of a document consisting of eleven sentences. Circles represent encodings; arrows represent the flow of computation; numbers in rectangles denote the order of computations. Numbers in rectangles with rounded edges represent computed relevance scores — the bottom one is relevance score for the encoding contained in the node; the upper one is cumulative relevance score that is summed together scores of all nodes on the path from the rootNode to the node. Filled with grey rectangles represent scores dropped by KeepTopT function. The bottommost rectangles contain the cumulative relevance scores of the sentences that are added to the attention scores of the words in them.

3.3.1 Constructing a Binary Tree Representation of Content.

To search for the most relevant sentences in the document it is proposed to construct a binary tree representation of the document's content. The outline of the ConstructBT process is the following.

1. Encode sentences with the Source2Token Self-Attention (please refer to the equation 2).
2. Divide encodings into pairs. (for an odd number of encodings leave one encoding in a dummy pair of a single member)
3. For each pair produce an aggregated single vector encoding by applying a function Merge(e_l,e_r) to the members of the pair. In the case of a dummy pair just copy the encoding further — creating a node with a single child node.
4. Go to step 2 applying it to the encodings produced on the previous step until there is only one encoding at the top (root encoding).

So that all sentences' encodings lie on the same (bottommost/last) level of the resulting tree. The example of the described binary tree structure is presented in Figure 2. This approach outputs a tree with at most 2n nodes for a document with n sentences, thus the computational complexity of such aggregation of the document's content is O(n) with respect to the number of sentences.

Merge function could be implemented in a number of ways, for example by averaging the input encodings, source2token self-attention, or multi-dimensional source2token self-attention [5]. Here it is proposed to use a Source2Token Self-Attention block (equation 2) with weights sharing throughout the tree (except for the initial sentences encoding procedure).

3.3.2 Selecting the Most Relevant Sentences from the Tree.

For a word embedding xi being transformed it is proposed to use the following procedure called TraverseTree to select top T most relevant sentences from the tree representation of the document's content. Similarly to the discussed above in the Section 3.2 method the intermediate result here would be the relevance score for each sentence in the document with all of them except for top T are being equal to -∞.

TraverseTree procedure (illustrated in Figure 3):

1. Start by calculation of a relevance score for the rootNode.
2. Process each level of the tree following the children of the nodes processed on the previous level.

a. Compute relevance scores for the nodes on the current level. Except for the nodes with already defined relevance scores of -∞ (could be there after the step 2c).

b. Apply KeepTopT function to the computed relevance scores on the current level keeping top T scores.

c. Propagate relevance scores of -∞ through the tree down to the bottommost level.

With this approach the relevance scores for the top T sentences for a given word could be calculated with O (t · log n) operations on encodings instead of O(n) with the earlier proposed approach (Section 3.2). Integration of these weights in the Attention block is straightforward.

$$HierarchicalCondAtt_j(X) = Softmax\big[Relevance\big(XW_j^{Qx}\big) + TraverseTree[X, ConstructBT(S), t]M\big]\big(XW_j^{Vx}\big)$$

Where $S \in R^{n \times d_{model}}$ is a matrix containing all sentences' encodings; $M \in R^{n \times nm}$ is an auxiliary matrix mapping of sentences to words; $X \in R^{nm \times d_{model}}$ is the input matrix of words' embeddings; $Softmax$ function is applied to its input row-by-row; $TraverseTree$ function is applied row-by-row to its input X; $W_j^{Qx}, W_j^{Kx} \in R^{d_{model} \times d_k}, W_j^{Vx} \in R^{d_{model} \times d_\vartheta}$ are learnable parameters.

Computational complexities of the mentioned designs of attention blocks are listed in the Table 1.

DISCUSSION

The proposed extensions to the Transformer model for Neural Machine Translation (NMT) demonstrate the potential to significantly improve the quality of translations, particularly when handling longer texts that require cross-sentence

context. By redesigning the positional encoding mechanism and introducing hierarchical encoding and conditional attention, the model is better equipped to preserve semantic and structural relationships across sentences within a document. This addresses a critical gap in current NMT systems, which often struggle with maintaining coherence, resolving ambiguities, and ensuring consistent meaning when translating documents rather than isolated sentences.

Comparison with Existing Approaches

Our approach builds upon the foundation of prior work in document-level NMT and context-aware translation mechanisms. Traditional methods, such as those using sentence concatenation with special tokens [4], offer some improvement in translation quality but are limited by the lack of explicit modeling of cross-sentence dependencies. Our hierarchical encoding strategy and the introduction of multi-level positional encoding go beyond simple concatenation by explicitly modeling the structural units within a text, allowing the model to distinguish between different levels of context (e.g., sentence, paragraph) more effectively.

Similarly, while previous efforts such as those by Voita et al. [3] and Zhang et al. [5] have explored the integration of discourse-level context through memory networks and hierarchical attention, our approach offers a more direct and computationally efficient solution. The conditional attention mechanism introduced in this work allows for selective aggregation of context from the most relevant sentences, reducing computational complexity while maintaining the necessary granularity of word-level attention. This method is particularly advantageous for large-scale translation tasks where computational resources are a limiting factor.

Limitations and Future Work

Despite the promising results, there are several limitations to our approach that warrant further exploration. First, the hierarchical encoding strategy assumes a clear and consistent structure within documents, which may not always be the case in real-world text. Texts with irregular or ambiguous sentence structures may pose challenges for the model's ability to effectively encode and utilize cross-sentence context. Additionally, while our approach reduces computational complexity compared to processing entire documents as single sequences, the hierarchical and conditional attention mechanisms still introduce additional overhead that may impact performance in low-resource settings.

Future work could focus on optimizing the computational efficiency of the proposed methods, perhaps by exploring alternative approaches to sentence encoding or by integrating dynamic context aggregation techniques that adjust the level of detail based on the complexity of the input text. Additionally, extending the model to handle multilingual contexts or specialized domains (e.g., legal, medical) could further enhance its applicability and robustness.

Implications for NMT Systems

The enhancements presented in this article have broader implications for the development of NMT systems, particularly in domains where the preservation of cross-sentence coherence and meaning is critical. By enabling more accurate and context-aware translations, these methods could improve the usability of NMT systems in professional and academic settings, where the integrity of translated documents is paramount. Furthermore, the ability to handle longer and more complex texts opens up new possibilities for applications in automated summarization, content generation, and cross-lingual information retrieval.

CONCLUSION

Extension of the Transformer model to incorporate cross-sentence context more efficiently suggests an improvement in quality in the field of Neural Machine Translation. Combination of hierarchical encoding, redefined positional encoding, and conditional attention mechanisms is a path forward for improving the accuracy and coherence of document-level translations. While challenges remain, the methods proposed in this work offer a foundation for a future research and development in NMT.

REFERENCES

1. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need.," CoRR abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762, 2017.
2. J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, "Convolutional Sequence to Sequence Learning," CoRR abs/1705.03122 (2017). arXiv:1705.03122 http://arxiv.org/abs/1705.03122, 2017.
3. H. Choi, K. Cho and Y. Bengio, "Context-DependentWord Representation for Neural Machine Translation," CoRR abs/1607.00578 (2016). arXiv:1607.00578 http://arxiv.org/abs/1607.00 [2]578 , 2016.
4. J. Tiedemann and Y. Scherrer, "Neural Machine Translation with Extended Context," CoRR abs/1708.05943 (2017). arXiv:1708.05943 http://arxiv.org/abs/1708.05943, 2017.
5. T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan and C. Zhang, "DiSAN: Directional Self-Attention Network for RNN/CNN-free Language Understanding," CoRR abs/1709.04696 (2017). arXiv:1709.04696 http://arxiv.org/abs/1709.04696, 2017.
6. N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton and J. Dean, "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer," CoRR abs/1701.06538 (2017). arXiv:1701.06538 http://arxiv.org/abs/1701.06538, 2017.
7. Y. Bengio, N. Léonard and A. C. Courville, "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation," CoRR abs/1308.3432 (2013). arXiv:1308.3432 http://arxiv.org/abs/1308.3432, 2013.