Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/42/20230687

Comparative study of sequence-to-sequence models: From RNNs to transformers

Jiancong Zhu

The Henry Samueli School of Engineering, University of California, Irvine, 2501 Alton Pkwy, Irvine, 92606, The United States

jiancoz@uci.edu

Abstract. In this comprehensive exploration of sequence-to-sequence models in Natural Language Processing (NLP), we have traced the trajectory of their evolution and contributions. Starting from foundational Recurrent Neural Networks (RNNs) to the revolutionary capabilities of Long Short-Term Memory (LSTM), In this comprehensive exploration of sequence-to-sequence models in Natural Language Processing (NLP), we have meticulously traced the trajectory of their evolution and impactful contributions. From the foundational Recurrent Neural Networks (RNNs) to the revolutionary capabilities of Long Short-Term Memory (LSTM), as well as the transformative innovations brought forth by Transformers and BERT, this review eloquently highlights the monumental advancements that have fundamentally reshaped our understanding and generation of language. The crux of our comparative analysis lies in its ability to spotlight the distinctive strengths and limitations inherent in each model. Through an intricate examination, we uncover their nuanced applications across a diverse spectrum of NLP tasks. Particularly noteworthy is the pivotal role played by Transformers and the transformative Bidirectional Encoder Representations from Transformers (BERT). The paper concludes with a summary and outlook of the entire paper.

Keywords: Sequence-to-Sequence, Recurrent Neural Networks, LSTMs, Transformers.

1. Introduction

The domain of Natural Language Processing (NLP) has, over the past few decades, transformed from a niche academic field to a cornerstone of modern technological applications. Central to this transformation has been the development and refinement of sequence-to-sequence (Seq2Seq) tasks. These tasks, pivotal in applications ranging from machine translation to automated chatbots, involve the conversion of one sequence, often a sentence or paragraph in one language, into another sequence, possibly its translation in another language or a summarized version of the original.

In today's digital age, the sheer volume of textual data generated every second is staggering. From tweets and blog posts to scholarly articles and e-books, the digital universe is awash with textual information. This deluge of data has not only underscored the importance of Seq2Seq tasks but also highlighted the challenges inherent in processing such data. Language, with its nuances, idioms, and cultural contexts, is inherently complex. Its sequences are dynamic, varying in length and structure, making it a challenging domain for computational processing.

DOI: 10.54254/2755-2721/42/20230687

The motivation for this study is rooted in the pressing need to unravel the complexities of Seq2Seq tasks in the context of NLP. As the volume and diversity of textual data continue to surge, understanding how to effectively process and manipulate sequences becomes paramount. This study seeks to dissect the underlying mechanisms of Seq2Seq models, shedding light on their evolution, strengths, and limitations. By doing so, we aim to provide insights into their applications across a spectrum of real-world scenarios.

This paper adopts the following structure: Section 2 offers a detailed examination of the models utilized in Seq2Seq tasks. It explores the progression from Recurrent Neural Networks (RNNs) to more sophisticated models such as Transformers and Bidirectional Encoder Representations from Transformers (BERT). This section provides a comprehensive overview and evolution of these models, highlighting their significance in Seq2Seq tasks. Section 3 presents a detailed comparative analysis of these models, highlighting their respective contributions and limitations. In Section 4, we draw conclusions from our analysis and discuss potential directions for future research in Seq2Seq tasks.

2. Evolution and Variants of Seq2Seq Models

2.1. RNNs and its Variants

- ***RNNs***

RNNs have been a cornerstone in the field of sequence modeling due to their inherent ability to remember past information and use it to influence future predictions. Traditional RNNs process sequences element by element, maintaining a hidden state that captures information about the processed parts of the sequence [1].

- ***Basic Architecture and Working Principle***

At the heart of the RNNs architecture is the concept of a hidden state, which captures information from previous time steps. At each time step, the network updates its internal state, which enables it to retain a type of memory about the sequence.

Mathematically, the recurrent mechanism of an RNN can be represented as:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}xt + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

Where:

- h_t denotes the hidden state at time step t.
- x_t represents the input at time step t.
- W_h, W_{xh}, and W_{hy} are weight matrices.
- b_h and b_y are bias terms.
- σ is an activation function, often the hyperbolic tangent [2].

Due to the vanishing and exploding gradient problems encountered by traditional RNNs, researchers have introduced a novel solution called the Independently Recurrent Neural Network (IndRNN). This innovative architecture was designed to overcome the limitations that hindered the effectiveness of traditional RNNs in handling long sequences.

Unlike traditional RNNs where all neurons in a layer are interconnected, in IndRNNs, neurons in the same layer operate independently. This independent operation allows for more effective regulation, preventing gradient-related issues and enabling the network to capture long-term dependencies.

The architecture of IndRNNs is visually depicted in the figure 1 below:

DOI: 10.54254/2755-2721/42/20230687

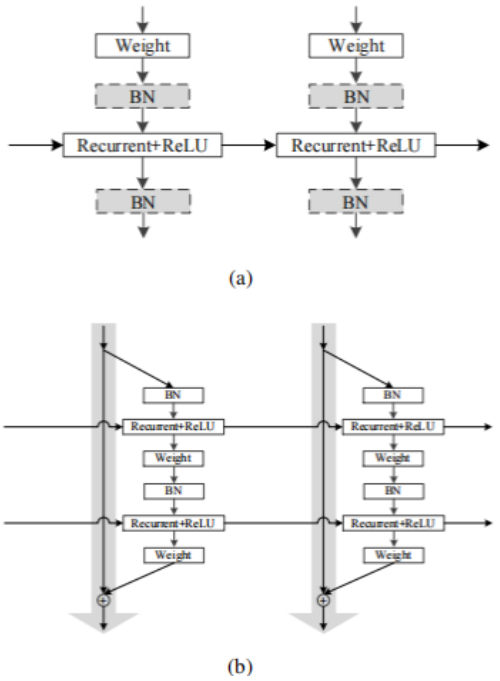

Figure 1. depicts the (a) basic architecture and (b) residual architecture of IndRNN [2].

Bidirectional Recurrent Neural Network RNNs extend the traditional RNN(BRNN) by processing sequences from both directions (forward and backward). This allows them to capture both past and future context, making them particularly useful for tasks where understanding the entire context is crucial, such as named entity recognition or part-of-speech tagging [3]. A BRNN is uniquely structured, comprising two distinct RNNs. One of these RNNs processes the sequence in a forward pass, from the beginning to the end, while the other operates in a backward pass, processing from the end to the start. At each time step, the outputs from both RNNs are merged, typically through concatenation, and this combined output is then channelled to subsequent layers. This bidirectional processing empowers BRNNs to adeptly capture intricate patterns in sequences, as they can assimilate context from both preceding and succeeding elements seamlessly.

While traditional RNNs consist of a single layer of recurrence, Deep RNNs stack multiple RNN layers on top of each other. This added depth allows them to capture more complex patterns and hierarchies in the data, leading to improved performance on a variety of tasks [4]. Deep Recurrent Neural Networks (RNNs) utilize multiple stacked layers of recurrent units to process input sequences. Starting at the bottom layer, the input sequence is initially processed, and as the data progresses through each layer, it operates at an increasingly abstract level. This hierarchical learning approach allows the network to capture both fine-grained details and broader patterns within the data. While the added depth enhances the network's capacity to model intricate sequences, it also introduces training challenges, particularly the risk of vanishing and exploding gradients.

2.2. LSTMs and Their Variants

Long Short-Term Memory (LSTM) networks, a specialized type of RNN, were created to excel at identifying patterns spanning considerable time durations. At the heart of the LSTM's architecture lies its cell state, functioning akin to a conveyor belt for information transfer, maintaining data integrity. This cell state's behavior is influenced by three gates: the Input Gate, employing a sigmoid activation function to ascertain the quantity of new information for preservation; the Forget Gate, responsible for determining the segment of the cell state to remember or discard; and the Output Gate, which employs the cell state and input to dictate the subsequent hidden state. The cooperative functionality of these gates empowers the LSTM to uphold long-term dependencies, a trait pivotal for its proficiency in tasks

DOI: 10.54254/2755-2721/42/20230687

such as predictive sequence analysis and natural language processing. Additionally, the intricate gating mechanism serves to alleviate the vanishing gradient predicament, a prominent challenge in conventional RNNs. LSTMs adeptly manage the flow of information through their gates and cell state, ensuring robust performance while preserving the core essence of the original text. LSTMs regulate information flow through their gates and cell state:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{3}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\mathrm{h_t} = (1 - \mathrm{z_t}) \odot h_{t-1} + z_t \tag{5}$$

The Peephole LSTM, an advanced variant of the standard LSTMs introduced by Gers et al. [5], is distinguished by its unique architecture that incorporates peephole connections. These connections grant the gate activations, namely the input, forget, and output gates, direct visibility into the cell state. In terms of its working principle, while a standard LSTM's gates make decisions based on the input and the previous hidden state, the Peephole LSTM's gates also consider the current cell state. This means, for instance, the forget gate can decide to retain or discard information based on the value of the cell state itself, offering a more nuanced memory retention mechanism. Such an architectural tweak allows the network to make more context-aware decisions, especially beneficial in tasks where the timing or duration of events is pivotal. By integrating the cell state into the gating decisions directly, the Peephole LSTM provides a richer context, potentially enhancing performance in specific sequence modeling challenges.

The Gated Recurrent Unit (GRU), innovatively presented by Cho and his colleagues [6], emerges as a streamlined rendition of the LSTM model. It effectively addresses the long-standing issue of vanishing gradients pervasive in conventional RNNs. The structural essence of the GRU revolves around a dual-gate mechanism: the reset gate and the update gate.

Reset Gate: The reset gate plays a crucial role in determining the extent to which the prior hidden state should be disregarded. By subjecting the sum of the prior hidden state and the present input to a sigmoid function, the reset gate facilitates this decision-making process:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{6}$$

Modification Gate: The modification gate serves the purpose of allowing the model to decide the extent to which the existing hidden state should be modified using the candidate from the new hidden state. This gate operates by blending the prior hidden state with the present input, a process executed via a sigmoid function:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{7}$$

Hidden State Candidate: The hidden state candidate arises from blending the present input with the prior hidden state, subject to adjustment by the reset gate:

$$\mathrm{h_t} = \tanh(\mathrm{W} \cdot [\mathrm{r_t} \odot h_{t-1}, x_t] + \mathrm{b}) \tag{8}$$

End State Fusion: The end state fusion involves a linear combination between the former concluding state and the potential concluding state, regulated by the adjustment gate.

$$\mathrm{h_t} = (1 - \mathrm{z_t}) \odot h_{t-1} + z_t \odot \mathrm{h_t} \tag{9}$$

The simplified architecture of the GRU has enhanced its computational efficiency compared to the conventional LSTM, all while preserving its capacity to capture prolonged relationships within sequential data. This model has effectively found its application across diverse tasks in sequence modeling, encompassing machine translation, speech recognition, and time-series prediction.

Depth Gated LSTM (DGLSTM), an advanced LSTM variant, was developed to counteract the challenges inherent in the sequential nature of RNN-based models. As delineated in the research by Zhang et al. (2020), DGLSTM represents an entire sentence as a graph, with individual words as word-

DOI: 10.54254/2755-2721/42/20230687

level nodes and an added unique sentence-level node [7]. Unlike traditional LSTMs, which update word states sequentially, DGLSTM updates all word states simultaneously through a message-passing mechanism. This design not only captures local n-grams effectively but also remains sensitive to long-range dependencies. The sentence-level node in DGLSTM is particularly adept at tasks requiring an understanding of the semantic correlation between different elements, such as slot and intent in spoken language understanding. By modeling sentences as graphs and updating word states in tandem, DGLSTM offers a robust solution to the limitations of traditional RNNs, ensuring efficient capture of both local contexts and overarching linguistic structures.

2.3. Bert and its Variants

BERT, marks a significant advancement in the field of natural language processing. Unlike conventional models that analyze text sequentially in a single direction (either left-to-right or right-to-left), BERT introduces a novel approach by simultaneously considering both directions. This enables BERT to capture contextual information from both preceding and succeeding words for each word in a sequence. The foundation of BERT's architecture lies in the Transformer model, which employs attention mechanisms to assign varying degrees of importance to different words within a sequence. BERT's fundamental concept involves pre-training on an extensive corpus using tasks like the Masked Language Model (MLM) and Next Sentence Prediction (NSP). BERT's architecture comprises multiple Transformer blocks, each incorporating multi-head self-attention mechanisms and feed-forward neural networks. A pivotal breakthrough in BERT's design is the incorporation of positional encodings into input embeddings, serving to indicate the relative position of words within a sequence. The attention mechanism, a cornerstone of the Transformer model and consequently BERT, is mathematically described by the following equation:

$$\text{Attention(Q,K,V)} = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})\text{V}, \tag{10}$$

where Q, K, V are the query, key, and value matrices, respectively, and $d_k$is the dimension of the keys. This equation captures the essence of attention, computing a weighted sum of values with weights determined by the query's compatibility with the corresponding key. BERT's bidirectional approach, combined with the Transformer architecture, has set new benchmarks in various NLP tasks, understanding context from both word directions. The essence of the attention mechanism is captured by this equation, which calculates a weighted sum of values based on the compatibility between a query and its corresponding key.

RoBERTa, an optimized derivative of the BERT model, was specifically developed to address some of the challenges and limitations associated with BERT. Both models are built upon the transformer architecture, a groundbreaking structure in the realm of deep learning, which employs attention mechanisms to extract and interpret contextual information from input data. This architecture enables models to discern context and intricate relationships between words or sub-words within a given sentence. Another significant divergence is RoBERTa's approach to training data. Unlike BERT, RoBERTa was trained on a considerably larger dataset, and for extended durations. This rigorous training regimen, combined with the vast amount of data, has enabled RoBERTa to consistently outperform BERT in a variety of benchmark tasks. Architecturally, while RoBERTa retains the essence of BERT's multi-layer bidirectional transformer design, it introduces certain refinements.

DistilBERT, conceived as an efficient iteration of BERT, emerged as a response to the computational and memory requirements posed by the original BERT model. At its core, DistilBERT employs the concept of knowledge distillation, a method in which a more compact model, referred to as the 'student', is educated to mirror the actions of a larger and more intricate model, known as the 'teacher'. Architecturally, DistilBERT integrates a range of adjustments to achieve its streamlined nature while upholding its performance. Notably, it adopts a reduced number of transformer layers – precisely six layers – in contrast to the BERT-base. An additional prominent deviation is observed in the omission of pooling within DistilBERT, a trait inherent in the traditional BERT, designed to attain a consistent-sized

DOI: 10.54254/2755-2721/42/20230687

representation of inputs that inherently vary in length. This mechanism empowers the model to assess the significance of distinct words in a sentence concerning a designated word, thereby shaping its understanding [8].

ALBERT, which stands for A Lite BERT, presents a transformative iteration of the original BERT model. Its meticulous design aims to strike a harmonious equilibrium between model size and performance efficiency. This equilibrium is achieved through two primary innovations: factorized embedding parameterization and cross-layer parameter sharing. In the context of factorized embedding parameterization, ALBERT introduces a clear distinction between the dimensions of the embedding layer and those of the hidden layers. By doing so, the model retains a robust embedding capacity capable of encompassing a vast vocabulary. Simultaneously, this design ensures that the hidden layers remain streamlined, leading to a considerable reduction in computational demands. Conversely, the cross-layer parameter sharing strategy constitutes a novel approach adopted by ALBERT. This strategy involves the strategic reuse of the same set of parameters across multiple layers of the model. This approach not only leads to a significant reduction in the overall parameter count but also serves as a regularizing mechanism, effectively addressing concerns related to overfitting that often arise in more expansive models [9].

3. Comparative analysis of Seq2Seq Models

Table 1 is a comparative analysis of various Seq2Seq models, highlighting their strengths and weaknesses.

Table1. A comparative analysis of various Seq2Seq models, highlighting their strengths and weaknesses.

Model	Strengths	Weaknesses
IndRNN	Efficiently handles long sequences without vanishing gradient problems.	Limited understanding of complex context.
Bidirectional RNN	Captures both past and future context effectively.	Computational complexity for bidirectionality.
Deep RNN	Incorporates multiple hidden layers for enhanced representation learning.	Proneness to overfitting with deep architectures.
Peephole LSTMs	Improved LSTM performance by allowing gates to see cell state.	Complexity in learning gate interactions.
GRUs	Simple architecture with efficient memory storage and gating mechanisms.	May struggle with modeling long-range dependencies.
Depth Gated LSTMs	Adaptable to various sequence-dependent problems due to depth mechanism.	Complexity in training and optimization.
RoBERTa	Proficient in context understanding and relationships with transformer attention.	Heavy computational requirements for training.
DistilBERT	Offers significant speed and resource advantages with retained performance.	Slightly reduced model performance compared to RoBERTa.
ALBERT	Efficient design with comparable or superior performance in handling large-scale text data.	Limited model interpretability.

- **Bidirectional RNNs vs. GRUs**

Bidirectional RNNs (BiRNNs) and GRUs are both extensions of the basic RNN architecture, aiming to capture long-term dependencies in sequences. According to the study on biometric electrocardiogram classification [10], BiRNNs demonstrated superior performance in capturing both past and future context, making them particularly effective for tasks like speech recognition and sentiment analysis. On the other hand, GRUs, with their gating mechanisms, have shown to mitigate the vanishing gradient problem, leading to better performance on tasks with longer sequences.

DOI: 10.54254/2755-2721/42/20230687

- **ALBERT vs. Depth Gated LSTM**

In the rapidly evolving domain of natural language processing, both ALBERT and Depth Gated LSTM have emerged as influential models. A study by Holger Schwenk et al. delves into the performance of various deep learning models in text classification tasks [11]. ALBERT, with its efficient design, demonstrates comparable or even superior performance to traditional models, emphasizing its prowess in handling large-scale text data. On the other hand, the Depth Gated LSTM, with its unique depth mechanism, showcases adaptability and efficiency across a spectrum of sequence-dependent problems.

- **LSTMs vs. RoBERTa**

LSTMs, with their intricate cell state and gating mechanisms, have been a cornerstone in sequence modeling for years. However, the advent of transformer-based models like RoBERTa has shifted the paradigm. According to the study, Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies [12], LSTMs excel in tasks requiring memory of long-term dependencies, especially when the syntax is crucial. On the other hand, RoBERTa, with its attention mechanism, outperforms in tasks requiring understanding of context and relationships between different parts of a text, such as question-answering and sentiment analysis. The study further elaborates on the LSTM's capability to capture syntactic dependencies, emphasizing its significance in language processing tasks.

While RoBERTa exhibits several strengths, it is important to acknowledge that there are certain scenarios where LSTMs still maintain their relevance. One such scenario is in tasks that involve syntactic analysis and dependency parsing, where LSTMs have demonstrated proficiency in capturing intricate linguistic structures. Additionally, in tasks where interpretability and explainability are crucial, LSTMs provide a more transparent view of their decision-making process compared to transformer models like RoBERTa. Therefore, while RoBERTa may surpass LSTMs in certain context-based tasks, LSTMs remain a valuable choice for tasks that demand in-depth syntactic understanding and transparent decision-making processes.

- **RoBERTa vs. Bidirectional RNNs**

The investigation into deep bidirectional LSTM RNNs for acoustic modeling in speech recognition delves deep into the capabilities of Bidirectional LSTMs [13]. The study provides a comprehensive overview of various training aspects of BLSTMs and their application in automatic speech recognition (ASR). The research found that deep bidirectional LSTMs outperformed feedforward neural networks for ASR. The bidirectional nature of these networks allows them to capture patterns from both past and future data points in a sequence, making them highly effective for tasks like speech recognition. When comparing RoBERTa and Bidirectional RNNs, it's evident that while both models are designed to capture context from both directions of a sequence, their primary applications differ. RoBERTa, with its transformer architecture, is more suited for tasks that require understanding the context between non-adjacent words in a sentence, making it ideal for NLP tasks. In contrast, Bidirectional RNNs, with their recurrent nature, excel in tasks that require understanding the temporal dynamics of a sequence, such as speech recognition or time series forecasting.

However, it's worth noting that the advancements in transformer models like RoBERTa are beginning to overshadow RNNs in many sequence-based tasks due to their ability to handle longer sequences and capture intricate patterns in data. But, as the study by Zeyer et al. suggests, Bidirectional RNNs still hold their ground in specific applications like ASR, where understanding the temporal dynamics is crucial [13].

- **The Relationship Between RNNs, LSTMs, and BERT**

In the realm of sequence modeling and natural language processing, the evolution from RNNs to LSTMs, and eventually to transformer-based models like BERT, signifies a continuous pursuit of capturing intricate patterns in data. RNNs, with their foundational architecture, laid the groundwork for sequence modeling, but their bidirectional variants (BiRNNs) enhanced the capability to grasp both past and future contexts. LSTMs, with their specialized cell states and gating mechanisms, further refined this approach, excelling in tasks that demand memory of long-term dependencies and intricate syntactic structures. Their adaptability is evident in the emergence of various LSTM-based models, such as Depth

DOI: 10.54254/2755-2721/42/20230687

Gated LSTM, which offers unique depth mechanisms. However, the introduction of transformer architectures, epitomized by models like BERT and its efficient variant ALBERT, has revolutionized the field. These models, with their attention mechanisms, have set new benchmarks, especially in tasks requiring a deep understanding of context and bidirectional relationships within texts. Yet, it's crucial to note that while BERT's prowess is undeniable, its computational demands can be a limitation. For instance, in real-time applications like chatbots or scenarios with limited computational resources such as mobile devices, models like GRUs and LSTMs might be more practical. In essence, the journey from RNNs to BERT encapsulates the field's progression from capturing sequential data's basic patterns to understanding the intricate nuances of language and context.

4. Conclusion

Throughout this study, we have embarked on a comprehensive analysis to understand the intricacies of various sequence-to-sequence models, from the foundational RNNs to the transformative capabilities of models like BERT and RoBERTa. The evolution of these models underscores the rapid advancements in the field of NLP and their increasing importance in real-world applications.

From the basic architecture of RNNs and their bidirectional and deep variants, we observed the initial attempts to capture sequential dependencies in data. The introduction of LSTMs and GRUs marked a significant leap, addressing the vanishing gradient problem and enhancing the model's ability to remember long-term dependencies. The transformer architecture, epitomized by BERT and its variants like RoBERTa, brought about a paradigm shift, emphasizing the importance of attention mechanisms and context understanding.

Our comparative analysis highlighted the strengths and weaknesses of each model. While RNNs and their variants excel in tasks requiring sequential memory, transformer-based models like BERT shine in understanding context and relationships in text. However, it's essential to note that no single model is universally superior. The choice of model largely depends on the specific requirements of the task at hand.

Looking ahead, the field of NLP is ripe for further innovations. As we continue to generate vast amounts of textual data, the demand for more efficient, accurate, and context-aware models will only grow. Future research could delve deeper into hybrid models, combining the best features of RNNs and transformers. Additionally, with the rise of unsupervised and self-supervised learning, future models might require less labeled data, making them more accessible and versatile.

In conclusion, the journey from RNNs to BERT is a testament to the relentless pursuit of excellence in the NLP community. As we stand on the cusp of new breakthroughs, it's exciting to envision the future directions this field might take, promising even more sophisticated models and applications that can truly understand and generate human language.

References

[1] Cho, Kyunghyun, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint* (2014)

[2] Li, Shuai, et al. Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper *RNN*. *Computer Vision and Pattern Recognition*, (2018)

[3] Schuster, Mike, and Kuldip K. Paliwal. Bidirectional recurrent neural networks. IEEE *Transactions on Signal Processing* 45.11 (1997)

[4] Pascanu, Razvan, Caglar Gulcehre, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint* (2013)

[5] Gers, Felix A., Jurgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation* 12.10 (2000).

[6] Cho, Kyunghyun, et al. Learning phrase representations using RNN encoder-decoder for statistical *machine* translation. arXiv preprint arXiv: https://arxiv.org/abs/1406.1078 (2014)

[7] Zhang, Linhao, et al. Graph LSTM with Context-Gated Mechanism for Spoken Language Understanding. *Association For the Advancement of Artificial Intelligence* (2020)

DOI: 10.54254/2755-2721/42/20230687

[8] Sanh, Victor, et al. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *NeurIPS Workshop* 2019.

[9] Lan, Zhenzhong, et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv preprint arXiv:1909.11942,* 2019

[10] Lynn, Htet Myet, et al. A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks. *IEEE Access*, 7, 145395-405 (2019).

[11] Schwenk, Holger, et al. Very Deep Convolutional Networks for Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics:* 1, 1107-1116, (2017).

[12] Linzen, Tal, et al. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4, 521-35, (2016).

[13] Zeyer, Albert, et al. A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. 2017. https://dx.doi.org/10.1109/ICASSP.2017.7952599