International Journal for Modern Trends in Science and Technology, 8(07): 176-187, 2022

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0807026
Available online at: http://www.ijmtst.com/vol8issue07.html

Text Summarization Using Deep Learning : An Analysis of sequence-to-sequence models and transformers

Aishvarya Akshaya Vishva Raman[1] | Mansi Kambli[2]

[1]Department of Computer Engineering, KJ Somaiya College of Engineering, Mumbai, Maharashtra, India.
[2]Professor, Department of Computer Engineering, KJ Somaiya College of Engineering, Mumbai, Maharashtra, India.
Email: a.vishvaraman@somaiya.edu

To Cite this Article
Aishvarya Akshaya Vishva Raman and Mansi Kambli. Text Summarization Using Deep Learning : An Analysis of sequence-to-sequence models and transformers. International Journal for Modern Trends in Science and Technology 2022, 8(07), pp. 176-187. https://doi.org/10.46501/IJMTST0807026

Article Info
Received: 05 June 2022; Accepted: 01 July 2022; Published: 09 July 2022.

ABSTRACT

When one opens news sites the first task one does would be to read the headlines and the associated sub-heading below it. Rarely someone would scroll through the entire news content for reading. Thus, the need for news apps and Twitter news feed has emerged that compiles the entire new article content into few lines that describes the actual story in a nutshell. The requirement of text summarization arises for generating abridged news content to the viewers that entails the main points. This paper proposes the different approaches to text summarization using deep learning. First approach is by using sequence-to-sequence model for summarizing the text articles. Further this paper enlists the drawbacks of sequence-to-sequence model and how to reduce the bottleneck by using attention layer. The alternative approach mentioned is by using the transformer model. The transformer network comprises of two architecture model i.e., the encoder and decoder architecture. This paper justifies the approach of using transformer network for text summarization over sequence-to-sequence model over large news datasets over varied set of parameters such as BLEU score, cross-entropy loss and accuracy.

KEYWORDS: *transformer, multi-head attention, LSTM, word2vec, abstractive summarization, RNN, word embeddings, BLEU*

1. INTRODUCTION

Text summarization is regularly sub-categorized into two types which is extractive text-summarization and abstractive text summarization. In extractive text summarization, the essential sentences are excerpted from the entire article or document. The important sentences are selected based on the criteria of statistical appearance. Text-Rank algorithm was devised for this purpose. After the pre-processing step is completed the processing of each input keyword is performed by calculating the weighted frequency of the keyword with the keyword having the maximum frequency that appears in the text article. Based on the calculation the keywords having the maximum weighted frequency are selected for the final summary of that particular article. In abstractive text summarization, it tends to capture the meaning of the entire text article and then build the summary by generating completely new paraphrases. It

is more complex in nature compared to extractive text summarization; often it is similar as to how humans summarize any document. In this paper we will be focusing only on abstractive text summarization using deep learning. Two approaches for abstractive text summarization are elucidated: sequence to sequence modeling using RNN's encoder-decoder architecture and the drawbacks related to it and then transformer architecture and the different transformer models used for summarization.

2. Related Work

A. Word Embeddings

Word embeddings are word representations and often entails how similar are any given words and its representation [1]. Each word of the vocabulary is represented in a vector format (one-hotrepresentation) in the pre-defined vector space. Featurized representations of word embeddings is given by a real value. Consider the vocabulary of ten-thousand words denoted by V= [a, aaron… zulu, <UNK>] where <UNK> denoted unknown word token. "e" denoted the embedding vector; "O" denotes one-hot vector representation of each word in the vocabulary and "E" represents the embedding matrix. Consider the example of the representation below:

Man-> O5391 (Word No: 5391 in V) is given by the one-hot vector representation as [0 0 ……1….0].

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

Table 1. Featurized representation: Word Embedding

This word representation tries to acquire the syntactic and semantic regularities [2] examined as constant vector offsets between different pairs of words sharing a particular relationship. Man: Woman as Boy: Girl; Ottawa: Canada as Nairobi: Kenya; Yen: Japan as Ruble: Russia and so on. For example:

$e_{man} - e_{woman} \approx e_{king} - e_{queen}$; $e_{man} - e_{woman} = [-2\ 0\ 0\ 0]$; $e_{king} - e_{queen} = [-1.92\ \ -0.02\ \ 0.01\ \ 0.01]$. Another way to find word "w" related would be by use of cosine similarity. $e_{man} - e_{woman} \approx e_{king} - e_{w}$; $sim(e_w, e_{king} - e_{man} + e_{woman})$.

$sim(u, v) = u^T v / ||u||_2 ||v||_2$ is the formula for calculating the cosine similarity between two word vectors. For learning word embeddings, the method given by neural language model [3] is to predict the next word in the sentence, given the previous word. It is given by:

$P(w_1^{T}) = \Pi_{t=1}^{T} P(w_t \mid w_1^{t-1})$; where w_t is the t^{th} word and the sub-sequence $w_i^j = (w_i, w_{i+1}, w_{j-1}, w_j)$. The n-gram model constructs table based on conditional probabilities of the given context words (last "n-1" words) for finding the target word. The major drawback for the n-grams model is that it requires a lot of space and RAM for capturing the dependencies between the distant words. A better counterpart for this method would be skip-grams model that works remarkably well.

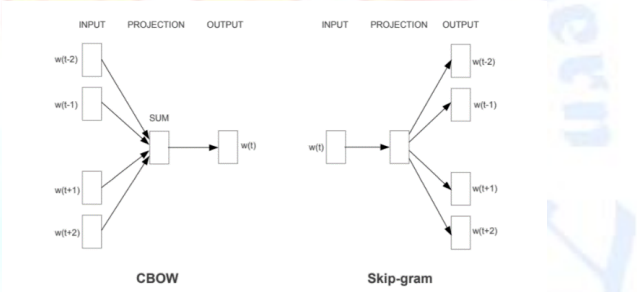

Figure 1. CBOW and Skip-gram model architecture [3]

Consider the sentence: "I want a glass of orange juice to go along with my cereal". In CBOW (Continuous Bag of Words) model it considers the context words to predict the target word. Context words are present to the left and right side of the word to be predicted i.e., the target word. In the above example the context could be "a glass of orange" and "to go along with" to predict the target word "juice".

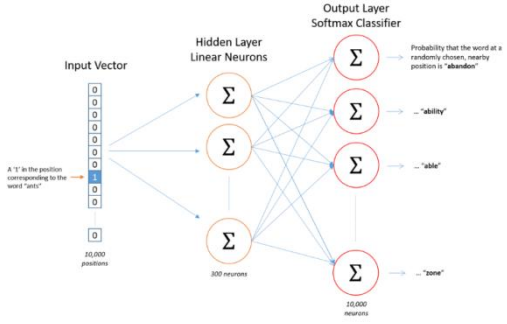

Figure 2. Detailed architecture of the skip-gram model [6]

In skip-gram model the surrounding nearby word is the target word to predict the context in the sentence. In the above sentence consider the target (t) word to be "orange" to predict the context (c) "juice"; "e" denoted the embedding vector; "O" denotes one-hot vector representation. $e_c = E * O_c$ and θ_j is the parameter associated with the output "c" the softmax calculation is given by:

$P(c|t) = e^{\theta_t^T * e_c} / \sum_{j=1}^{10000} e^{\theta_j^T * e_c}$

The loss function for the softmax classification is given by:

$\mathcal{L}(\hat{y}, y) = - \sum_{i=1}^{1000} y_i * \log(\hat{y_i})$. The major drawback with softmax classification is that the summation over the entire vocabulary is carried out in the denominator of the equation. To resolve the drawback negative sampling is carried out on the softmax classification by multiplying the entire equation with $(1/|V|)$. Skip-gram is a type of supervised learning problem. Word2vec based models are not capable of creating word embeddings for the words not present in the corpus/vocabulary that they were trained on.

Further regarding the construction of a new word embedding model called GloVe (Global Vectors), this model acquires the statistics of the entire global word vocabulary corpus [4]. Consider the sentence "I want a glass of orange juice to go along with my cereal"; here X_{ij} is the number of times the term "i" (target word) appears in context of "j". Also, $X_{ij} = X_{ji}$

The goal is to minimize the below function value:

$\sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij}) (\theta_i^T e_j + b_i + b_j` - \log(X_{ij}))^2$ where $f(X_{ij})$ is the weighting term and θ_i and e_j are initialized at the beginning of training and both of which are symmetric. The above basic word embedding methods have a limitation with respect to unlabeled corpora of vocabulary or out of vocabulary words where it is not capable of recognizing the morphological structuring of words context thereby assigning new vector representation for each word. To overcome this problem fastText was introduced by the Facebook research team in 2016 where a new approach is built based on the skip-gram model where every word is represented as the sum of each characters of n-gram model [5]. For example, for the word "where" and if n=3 then it is represented for the trigrams model as sum of each characters as follows: <wh, whe, her, ere, re> It can capture out of vocabulary words after training. Also, the advanced word embedding methods recently developed are BERT (Google in 2018), ELMo (Allen Institute for AI in 2018) and GPT-2 (OpenAI in 2018). Also, the tunable pre-trained models were made available for these methods.

B. Sequence Models

Deep neural networks in general have the capability of representing extreme complex functions [7]. In deep neural networks in general learning requires that the gradients of complicated functions be computed first. For calculating the gradients back-propagation algorithm is performed. It displays the capability of learning different features at several different levels of abstraction starting from the edges (at the shallow layers that are closer to the input) towards the extremely complex features (which are at the deepest of layers that are closer to the output). Consider a sequential data where an input audio clip "X" is asked to map to a textual transcript "Y". In this case a standard network model would not be appropriate to use because the input length and the output length could be different for different examples and it does not share the features learnt through the layers across different text positions. Thus, Recurrent Neural Networks was introduced for sequence data. Its real-time applications include speech recognition, music generation, sentiment classification, DNA sequence analysis, machine translation, video activity recognition, named entity recognition, etc.

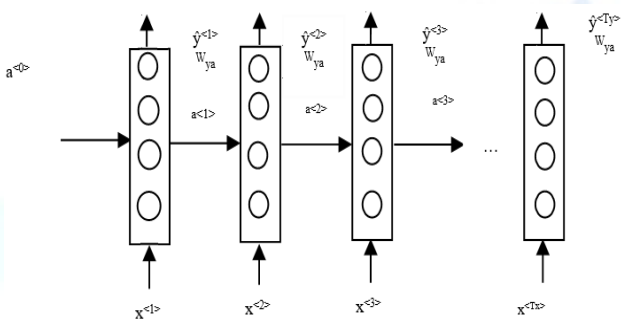

Figure 3. Recurrent Neural Network Architecture

In the above architecture $a^{<0>}$ represents a vector of zeros. Consider the input x to be "Harry Potter and Hermoine Granger invented a new spell" thus the input tokens denoted by $x^{<1>}$: Harry, $x^{<2>}$: Potter, $x^{<3>}$: and $x^{<4>}$: Hermoine….. $x^{<9>}$: spell and $T_x = 9$ indicating 9 words or 9 input tokens. The output tokens are denoted by $y^{<1>}$, $y^{<2>}$, $y^{<3>}$,….. $y^{<t>}$, where $T_y = 9$ in this case if $T_x = T_y$ then it is a case of many-to-many mapping known as bi-directional

RNN (BRNN) often used for machine translation. The forward propagation in RNN is given by first $a^{<0>}= 0$, $a^{<t>}= g(W_{aa}*a^{<t-1>}+ W_{ax}*x^{<t-1>}+ b_a)$ where g is represented as the tanh / ReLU function. The output token $\hat{y}^{<t>}$is calculated by the formula: $\hat{y}^{<t>}= g(W_{ya}* a^{<t>}+ b_y)$. The backpropagation is given by:

$L^{<t>}(\hat{y}^{<t>}, y^{<t>}) = - y^{<t>}\log (\hat{y}^{<t>}) – (1 – y^{<t>}) \log (1 - \hat{y}^{<t>})$

The loss function is given by:

$L (\hat{y}, y) = \sum_{t=1}^{Ty} L^{<t>}(\hat{y}^{<t>}, y^{<t>})$. RNN could also be used for sequence generation for the new articles as input. Consider the input to be a large corpus of English text and the output as:

Cats average 15 hours of sleep a day. <EOS>

$y^{<1>}$ $y^{<2>}y^{<3>}$ ……

$\hat{y}^{<t>}$ denotes that it predicts the output token what is the chance of the next output word given the previous output words.

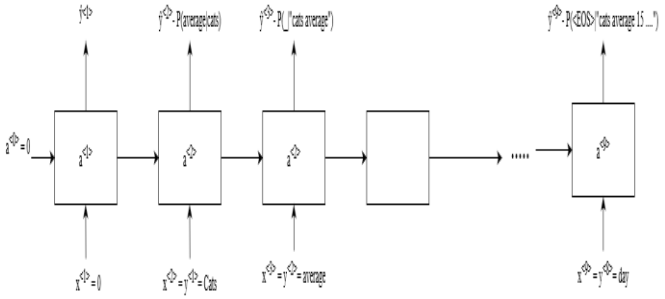

Figure 4. Text sequence generation using RNN

In the above figure the working of sequence generation is captured. In RNN's it could capture dependencies of sequence within a short range and takes up less RAM space compared to the other n-gram models. However, language (sentence) has the possibility of having long-term dependencies. The drawback is one needs to have the entire sequence of data before making predictions. Also, it suffers from the problem of vanishing gradient as one backpropagates from the final layer to the first layer, the weight matrix is being multiplied in each step thus the gradient descent can decrease exponentially down to zero. To overcome the problem of vanishing and exploding gradients (where the weights and activation units take up Nan value while training the model), LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Units) models were introduced for performing several tasks related to sequence data such as machine translation, sentiment analysis and so on.

Apart from RNN's another reasonable approach for dealing long sequences could be to utilize a recursive version of the Recurrent Neural Network model [8] where the learnable parameters at each layer are propagated through the entire recursive network. GRU displays the proficiency to capture relevant information for long sequence of word in the sentence [9]. GRU also decides how the information in the hidden state could be updated. The activation (candidate activation) function computed in GRU at any recursive level "t" gives the option of the function calculated for the left child or the right child [8] of the recursive model. This exercises the overall structure of the model to accustom to the changing input information providing flexibility to the model thereby modulating the flow of information throughout the structure. Thus, the need to have a separate memory cell is prevented.

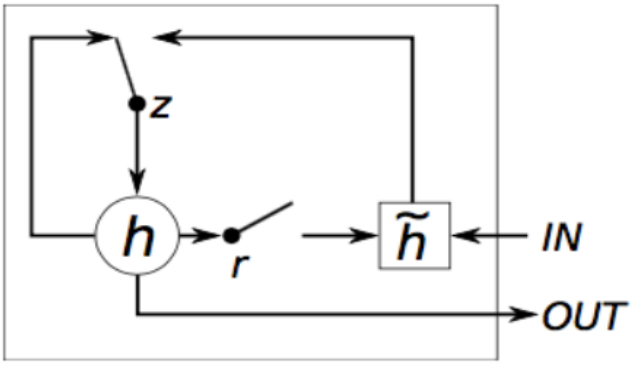

Figure 5. Gated Recurrent Unit Model Architecture [9]

The equations for update gate, reset gate and the activation functions for GRU are formulated below:

$\tilde{h} = \tilde{c}^{<t>}= \tanh(W_c [\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$ (candidate activation)

$u = \Gamma_u= \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$ (update gate)

$r = \Gamma_r= \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$ (reset gate)

$h = c^{<t>}= \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$(activation function)

The information present in the update gate of GRU will not be overwritten but rather it will be preserved which advocates for long-term temporary dependencies in case of machine translation, sequence generation. LSTM (Long Short-Term Memory) is slightly more powerful than GRU and more generalized. The significant difference between LSTM and GRU is in the disclosure of information present in the memory cell unit [9]. In the case of LSTM, the amount of information present in the memory could be controlled from being exposed to the different gates in LSTM through output gate. However,

in GRU that is not the case and the entire information present in the memory is completely revealed to the other gates in the network.

The equations for LSTM are illustrated as followed:

$\tilde{h} = \tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$ (candidate activation)

$u = \Gamma_u = \sigma(W_h[a^{<t-1>}, x^{<t>}] + b_u)$ (update gate)

$f = \Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$ (forget gate)

$o = \Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$ (output gate)

$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$

$a^{<t>} = \Gamma_o * \tanh(c^{<t>})$

LSTM makes use of two gates (update gate and forget gate) to compute the final value of the hidden unit. "c" indicated the memory cell unit. Thus, LSTM and GRU units are incorporated within RNN encoder-decoder architecture. LSTM gates discard the information that is not relevant, add more new information that is at priority and later produce the output. Applications of LSTMs include next character predictions, chatbots, music composition, image captioning and speech recognition.

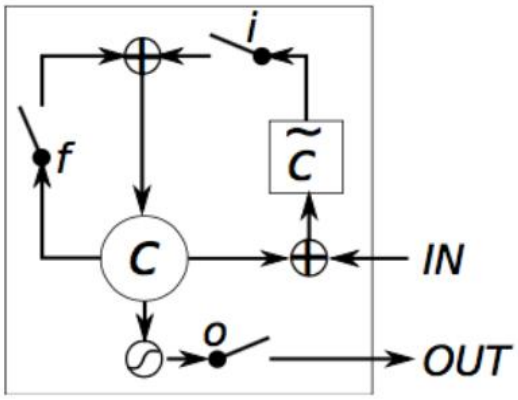

Figure 6. LSTM Unit Architecture [9]

C. Encoder-Decoder Model

In the encoder-decoder architecture used for statistical machine translation LSTM and GRU are used that maps variable length sentences [7] into a fixed-length dimensional representation of vector.

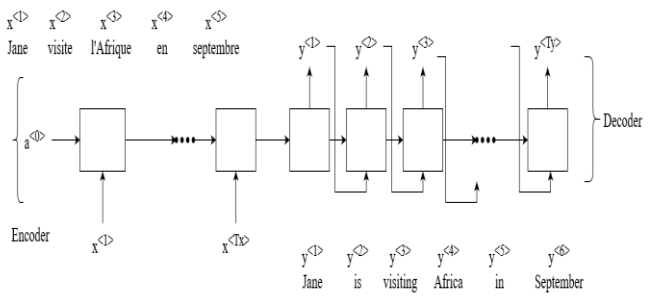

Figure 7. Encoder-Decoder Model example

The statistical machine translation is given by a conditional language model to predict the probability of the output English translation given the input French sentence: "Jane visite l'Afrique en Septembre" i.e., $P(y^{<1>}, y^{<2>}, \ldots y^{<Ty>} | x^{<1>}, x^{<2>}, \ldots x^{<Tx>})$. To find the most likely translation of the input sentence (French sentence in this case) an algorithm was devised that could maximize the conditional probability value for finding the value of output "y" as mentioned before.

arg max $y^{<1>}, y^{<2>}, \ldots y^{<Ty>} P(y^{<1>}, y^{<2>}, \ldots y^{<Ty>} | x)$

The most trivial algorithm was devised called as Beam Search [7]. Beam search is more optimal compared to greedy search algorithm. In greedy search, the most likely first word is picked based on the conditional language model. This approach will not be prefect for the long run in the case for longer sequences. Greedy decoding is appropriate for shorter sequences.

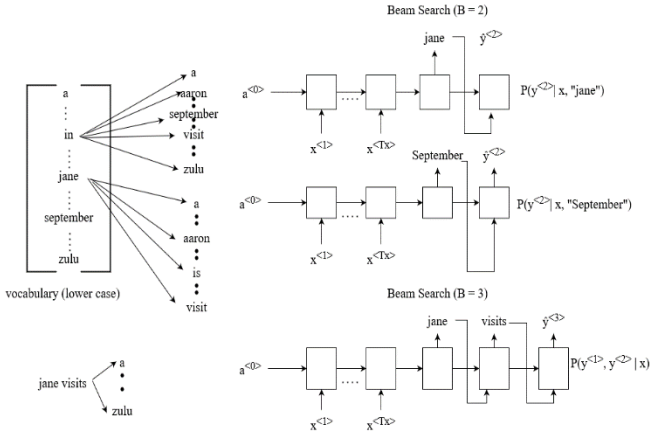

Figure 8. Beam Search Decoding for Conditional Language Model

Based on the beamwidth the search complexity of beam search algorithm would be B * |V| where |V| denotes the number of words in the vocabulary which is far less than the search complexity of greedy decoding that occupies

$|V|^{\text{no of translations possible}}$. If beamwidth is equal to one, then it works similar to greedy search decoding. The possible shortcoming of using beam search is its computation speed decreases as the length of the beam increases.

For evaluation of machine translation, human evaluation could be time-consuming and exhaustive. Hence a mechanized evaluation of machine translation was introduced called BLEU (Bi-lingual Evaluation Understudy) [11]. This BLEU score method highly associates with human evaluation. It inspects the

candidate translations with the corresponding reference (human) translations. Also, BLEU score does not capture the sematic regularities or the structure of the sentence. If the input sentence in French taken into account is "Le chat est sur le tapis". Its corresponding machine translation is "The cat on the mat".

Consider the following scenario where:

Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

Machine translation Output: The cat the cat on the mat.

Count: calculates the count of the given term if present in the translated output.

$Count_{clip}$: calculates the count of the given term if present in either of the candidate (reference) translations.

	Count	$Count_{clip}$
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1

Table 2. BLEU Score example on bigrams

p_n (n-grams) = $\sum_{n\text{-grams } \epsilon\ y} Count_{clip} / \sum_{n\text{-grams } \epsilon\ y} Count$

For the above stated example $p_1 = 4/6$. The combined BLEU score is given by:

BP * exp (1/4 * $\sum_{n=1}^{N} p_n$) where BP is denoted by brewy penalty. BP is given by:

BP = 1 (if translated output length > reference output length).

BP = exp (1 – translated output length/reference output length) (otherwise). The drawback of encoder-decoder models mainly was it depended on a fixed-length memory i.e., only a fixed amount of information is passed from the encoder to the decoder model. As the size of the input sequence increases the performance of the model decreases. The encoder-decoder architecture performs well for shorter sentences, as the number of sentences multiply (more than 30 or 40) the BLEU score also drops.

D. Attention Model

To overcome the shortcomings of the encoder-decoder model a bi-directional RNN with hidden units was introduced for machine translation [13].

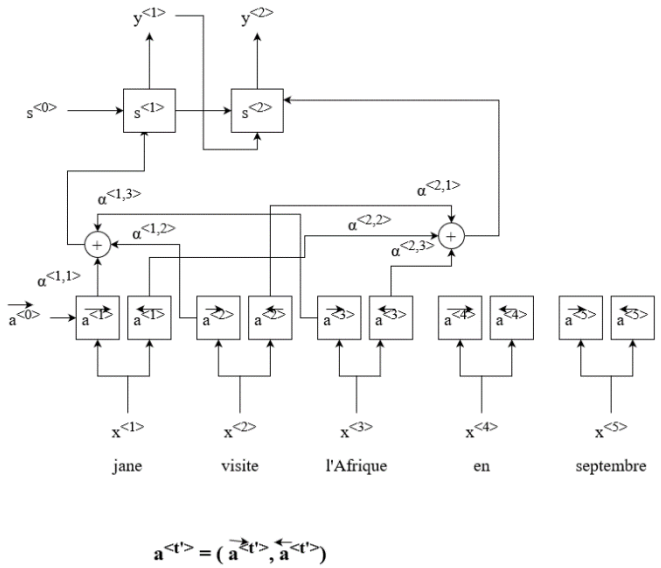

Figure 9. Example of bi-directional RNN with hidden layers

The hidden state is denoted by "c". The units in the hidden layer are calculated by pointwise addition with weighted sum. This weighted sum is dependent on the previous hidden state in the decoder. The context vectors $c^{<t>}$ are given as inputs to the post-attention RNN. The context vectors are computed as:

$c^{<1>} = \sum_{t'} \alpha^{<1,\,t'>} a^{<t'>}$ and $c^{<2>} = \sum_{t'} \alpha^{<2,\,t'>} a^{<t'>}$. Attention layer enables the model to focus on the important information. $\alpha^{<t,\,t'>}$ denotes the amount of attention $y^{<t>}$ should pay to $a^{<t'>}$.

$\alpha^{<t,\,t'>} = \exp(e^{<t,\,t'>}) / \sum_{t'=1}^{Tx} \exp(e^{<t,\,t'>})$. Here, $e^{<t,\,t'>}$ is computed using a small neural network because it's values is required for the network to determine where "attention" need to be focused. The BLEU score of sequential models using attention mechanism is comparatively more than the sequential encoder-decoder models.

The drawback of attention type model (Neural Machine Translation model) and sequential models is that it computes the attention for the input sequence one token at a time. No parallel computing is possible in the case of RNN. If the complexity increases for the sequential encoding the performance reduces. In case of neural machine translation model [3] in the encoder part the hidden states layers computed is utilized as keys and values and in the pre-attention decoder part, the hidden states are used as queries. Teacher forcing [12] is a method used in neural machine translation for improving the training performance of the model. For each predicted output token generated (translated

output) it is compared with the actual output token if it is a match then the score for the correct match is considered. The actual output token is passed as the next input token in the decoder.

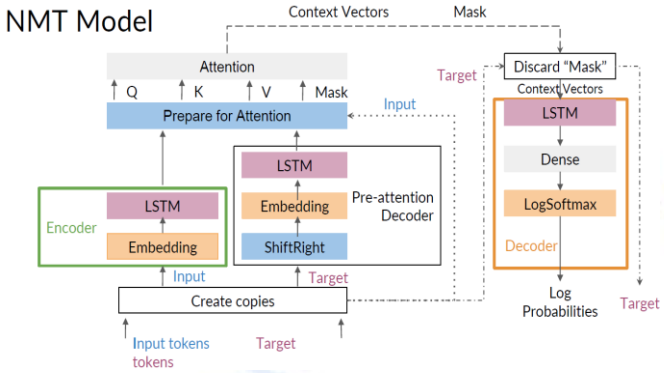

Figure 10. Neural Machine Translation architecture [12]

E. Transformers

The drawbacks faced in neural machine translation model is solved in transformers. Transformer architecture enables on to run extensive computations for longer input sequences in parallel. Transformers also prevent the problem of vanishing gradients. Transformer architecture amalgamates the usage of attention-based representations and a convolutional neural network way of processing [14]. Convolutional Neural networks can accept a larger input of sequence of words and the representations for the sequences could be computed in parallel in transformer architecture with a goal for reducing sequential computing.

Transformers avoids the use of recurrence or recurrent neural networks and instead relies heavily on attention mechanism [14]. An attention function could be stated as alignment of a query and a set of key-value pairings to an output. The attention intuition mechanism includes self-attention and multi-head attention. Self-attention also familiarized as intra-attention is an attention mechanism that computes "n" representations for "n" words/tokens in an input sequence. Self-attention structure could be applied in different applications such as reading comprehension, abstractive summarization, textual entailment (whether one textual fragment could be inferred from another) and learning the sentence representations self-sufficient of the given task. In self-attention the maximum path length between any two input and output positions through all the layers in the input and output sequences is comparatively less than the path length required for recurrent neural network which would be beneficial for long-term dependencies for the input sequences preventing loss of information. Multi-head attention process is a loop over the entire self-attention process. The scaled dot-product attention is computed multiple times parallelly.

The attention function is formulated as:

A (q, k, v) = ∑i (exp (q * k<i>) / ∑j exp (q * k<j>)) * v<i>.

Query (q) indicates the question. The scaled dot-product attention method consists of inputs as keys of dimension dk, query of dimension dk and values of dimension dv. Dot product is computed of matrix Q (query) with the transpose of matrix K (key). If the values of the dimension dk increases the dot product of the resultant matrix could possibly grow in weight, hence it its divided by the square root of the dimension dk and further softmax function is applied to it. This helps in optimizing the time and space complexity of the function by scaling out. It is finally multiplied with the value matrix.

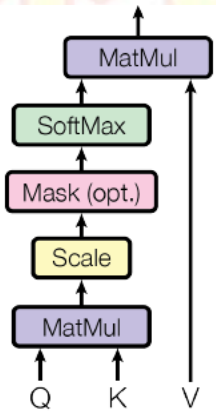

Figure 11. Scaled dot-product attention [14]

The scaled dot-product attention is much more optimal than the additive attention.

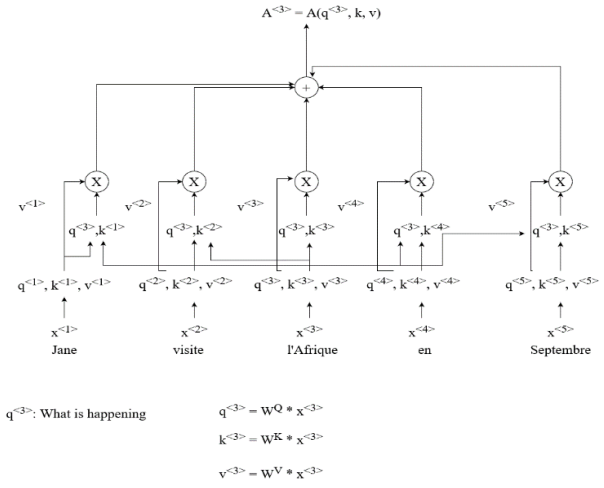

Figure 12. Working of self-attention mechanism

Similarly, the same could be computed for $A^{<1>}$, $A^{<2>}$ and so on based of the query q. For multi-head attention the computation is given as Multihead (Q, K, V) = concat($head_1$, $head_2$,….$head_n$) W_o, where.

$head_i$ = Attention ($W_i^Q Q$, $W_i^K K$, $W_i^V V$). The transformer architecture is composed of an encoder-decoder structure having multi-head self-attention mechanism.

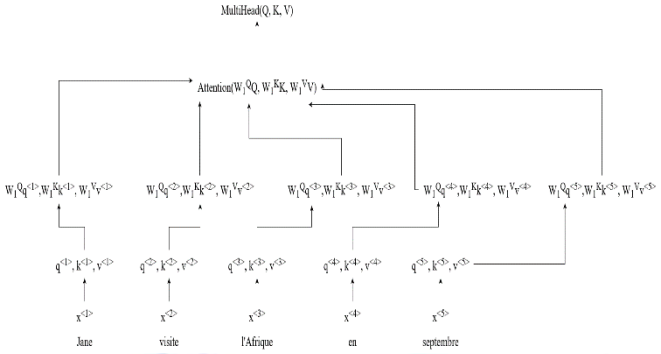

Figure 13. Working of Multi-head attention

In multi-head attention mechanism, the linear projections are projected "h" times to the dimensions d_k (keys, queries) and d_v (values).

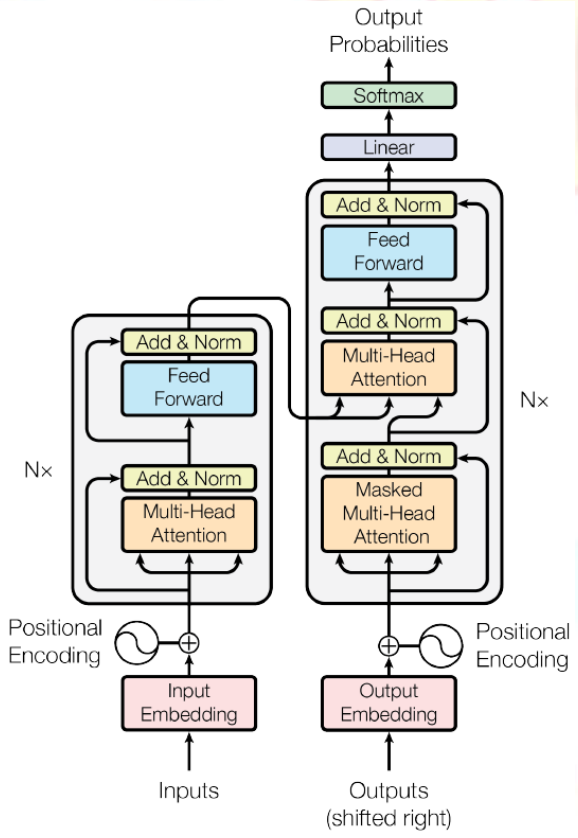

Figure 14. Transformer Architecture Model [14]

The step of positional encoding is paramount in the model for translation purpose because the position and the order of words are required for construction of sentences in any language given. Also, it contributed extra information to the entire model. For any positional encoding algorithm, it is essential that it outputs a unique encoding (word's position in a sentence) for every time-step, the distance between any two time-steps should be uniform for all the sentence lengths. Most important parameter would be the ability of the model to generalize to longer sequences based on the positional encoding algorithm working.

In general, the transformers could be used for several applications in NLP (Natural Language Processing) domain such as text-summarization, auto-complete, named entity recognition, question answering, language translation, chat-bots, sentiment analysis, market intelligence, text classification, character recognition, spell checking and so on. The state-of-the-art transformers recently introduced include GPT-2 (Generative Pre-training for Transformer), BERT (Bi-directional Encoder Representations from Transformer) and T5 (Text-to-text transfer transformer).

3. Literature Survey

Since text summarization utilizes several approaches ranging from NLP domain, sequential models and transformers, a comparative analysis of several approaches using a combination of different models could be inferred from below.

Ahmed Elmogy and Belal Mahmoud [15] put forth a deep learning approach for generation of text. They practiced one of the most prominent deep learning architectures celled bi-directional recurrent neural network. It is a sequence-to-sequence model that used LSTM (long short-term memory) instead of GRU or RNN. The datasets that were used in this case for training and evaluating the model were Alice In Wonderland and Twilight stories. They considered a relatively small vocabulary for testing and evaluation purpose. The embedding methods they used were word2vec and one-hot encoding, however the latter embedding method was applied in only the Alice In Wonderland dataset. Before applying the embedding methods, the data pre-processing steps were carried out for both corpuses (tokenizing sentences into a set of words, converting into lower case, replacing the short form of words with the original ones).

Based on the results performed it was implied that word2vec embedding method gave a better end accuracy than the one-hot encoding method (for larger sequences) [15]. It was also inferred that in case of a tiny

corpus word2vec generated unreadable text due to overfitting.

Dima Suleiman and Arafat A. Awajan [16] purported in this paper a survey regarding their reviews for extractive text summarization using deep learning approaches. The deep learning approaches specified were Restricted Boltzmann machine, variational autoencoders and recurrent neural networks. The evaluation metric of text summarization for all the approaches was ROUGE (Recall Oriented Understudy of Gisting Evaluation) that is used to compare the candidate with human translations. The datasets utilized for all the three approaches in a whole were listed as DUC2002, DUC2006, DUC2007, CNN/Daily Mail. It was inferred from the survey that SummaRuNNer a bi-directional RNN incorporated with GRU (Gated Recurrent Unit) displayed the highest ROUGE-1 and ROUGE-2 score out of all the mentioned deep learning approach for DUC2002 dataset and a ROUGE-L score was highest for CNN/Daily Mail dataset. The embedding method used for all the cases was word2vec only.

Nikhil S. Shirwandkar and Dr. Samidha Kulkarni [17] proposed an approach for extractive text summarization which is an amalgamation of Restricted Boltzmann Machine and Fuzzy Logic with respect to a single document. The text summary generated from RBM and the summary generated from fuzzy logic is joined and further processed using some set operations for obtaining the final resultant summary.

RBM is a neural network that contains a bi-directional connection between any two given nodes, the resultant network is said to represent a bipartite graph. Two hidden layers were taken into consideration for the proposed approach. For the purpose of extracting features from the re-processed sentences, tf-isf (inverse sentence frequency) score was computed. The summary generated through RBM was selected as top 50% of the whole based on the decreasing order of sentence score. Further, the summary generated through fuzzy logic was selected based on the feature score. The final summary that was generated was based on the positional sentence sorting for the most common and the rare sentences obtained from both the methods. Based on the results performed it was concluded that the proposed method had a higher ROUGE score compared to the singular RBM methodology.

Afsaneh Rezaei, Sina Dami and Parisa Daneshjoo [18] introduced two text summarization systems based on deep learning mainly variational autoencoders and deep belief network (based on RBM) for multiple-documents extractive text summarization purpose. The deep belief network was a several stacks of Restricted Boltzmann Machine composed together. In the proposed approach the implementation was done for 9-layer network having 7-hidden layers within it. The proposed variational autoencoder was also mentioned to be built of 9-layers. In the initial pre-processing steps the feature extraction of sentences is done using TF-IDF score and the position of the sentences were also taken into consideration. The dataset used for implementation of the two systems was DUC2007. The evaluation metric taken into account was ROUGE-1 and ROUGE-2. Based on the results evaluated it was concluded that proposed variational autoencoders had a greater ROUGE-1 and ROUGE-2 score than the proposed DBN.

Yisong Chen and Qing Song [19] put forth a methodology for resolving the issue of deviating from the main topic in abstractive text summarization. The combination of text-rank and BART model was proposed as a methodology for the extraction and generation of summary from the given news text article. Text-rank algorithm is graph-based that was designed based on the concept of voting/ recommendation. If a vertex is connected to another vertex in the graphical structure a vote is cast for the same. The BART model is based in the concept of a denoising-autoencoder that prevents the entire network from learning the identity function. The BART model architecture is based on the transformer-model neural machine translation. The model is primarily utilized for training the sequential models. Once the summary was generated from the text-rank algorithm and the BART model the summarization sentences generated are united for giving out the resultant final summary. CNN/Daily Mail dataset was being tested for the proposed model. Of the entire dataset only approximately 13,760 news articles were used for carrying out the experiment and analysis. The methods using which the summarization process was carried out were TextRank, BART, TextRank with BART. It was implicated that the combination of TextRank and BART gave out a higher value of

evaluation metrics for ROUGE-1, ROUGE-2 and ROUGE-L.

Kethan Pabbi and Sindhu C [20] elaborated about abstractive opinion summarization (from articles) and the method mentioned for obtaining abstractive summarization was a bi-directional LSTM (long short-term memory) for improving the performance and increase the long-term dependencies of the model an attention layer was also introduced. The attention layer in general computes the context vectors and the hidden units. The dataset used as mentioned was Amazon Fine Food Reviews. The evaluation metrics that were prioritized for model performance over the dataset were ROUGE-1, ROUGE-L and BLEU scores. Based on the results it was implied that ROUGE-1 had had higher value than ROUGE-L and BLEU for the results calculated. For the mentioned dataset.

4. Methodology

The proposed methods for text summarization are based on two approaches: encoder-decoder model using LSTM and the transformer model architecture with attention mechanism. ADAM optimizer was used for both the approaches for training and evaluating the model.

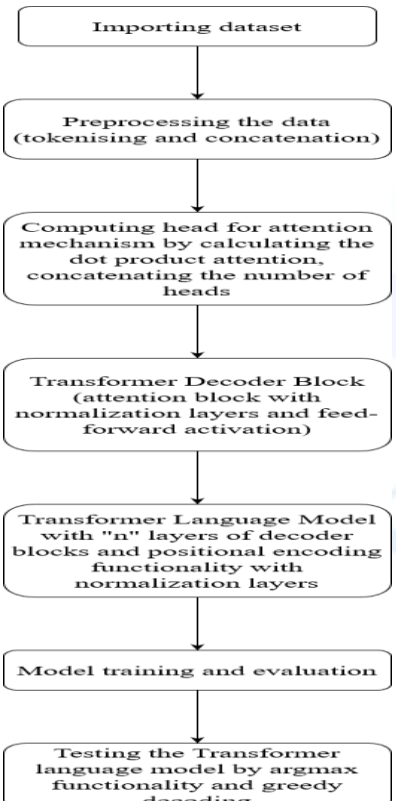

Figure 15. Proposed methodology for text summarization using Transformers

The dataset used for testing and training the models for the two approaches was CNN/Daily Mail. It comprised of several news articles and the highlights that indicated the target summary of the new article to me matched during the evaluation time.

5. Experimental Results And Analysis

Input sentence: It's the posing craze sweeping the U.S. after being brought to fame by skier Lindsey Vonn, soccer star Omar Cummings, baseball player Albert Pujols - and even Republican politician Rick Perry. But now four students at Riverhead High School on Long Island, New York, have been suspended for dropping to a knee and taking up a prayer pose to mimic Denver Broncos quarterback Tim Tebow. Jordan Fulcoly, Wayne Drexel, Tyler Carroll and Connor Carroll were all suspended for one day because the 'Tebowing' craze was blocking the hallway and presenting a safety hazard to students. Scroll down for video. Banned: Jordan Fulcoly, Wayne Drexel, Tyler Carroll and Connor Carroll (all pictured left) were all suspended for one day by Riverhead High School on Long Island, New York, for their tribute to Broncos quarterback Tim Tebow. Issue: Four of the pupils were suspended for one day because they allegedly did not heed to warnings that the 'Tebowing' craze at the school was blocking the hallway and presenting a safety hazard to students.

Summarized sentence (using transformer model): Jordan Fulcoly, Wayne Drexel, Tyler Carroll and Connor Carroll were suspended for one day. Four students were suspended for one day because they allegedly did not heed to warnings that the 'Tebowing' craze was blocking the hallway and presenting a safety hazard to students. <EOS>.

Steps	TrainingCross Entropy	Evaluation Accuracy
1	69.19%	50.31%
500	50.71%	81.40%
1000	35.92%	79.53%
1500	35.24%	85.12%

Table 3. Transformer Model Evaluation Metrics

Steps	Training Accuracy	Evaluation Accuracy
2	62.22%	63.13%
4	64.17%	63.92%
6	65.09%	65.31%
12	66.57%	65.27%
14	68.56%	64%

Table 4. Sequential Model Evaluation Metrics

For sequential models using LSTM the performance deteriorates after 14 epochs. It works efficiently for only a small quantity of test data or a new article of few words. Based in the results captured it could be concluded that transformer model shows a better performance and greater evaluation metrics score over the sequential model for large-sized datasets. The implementation was done over Google Colab.

6. Conclusion and Future Scope

For abstractive text summarization the transformer model architecture would be preferable for training, testing and evaluating large-sized datasets because of the long-term dependency factor. Sequential models work efficiently for small-sized datasets, however the performance decreases as the size of the dataset and the information increases. Further, one could implement beam search decoding for testing the text articles input within the model. Ahead of transformers, pointer generator network could also be considered for text summarization.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

References

[1] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

[2] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

[3] YoshuaBengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. J. Mach. Learn. Res. 3, null (3/1/2003), 1137–1155.

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

[5] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the association for computational linguistics, 5, pp.135-146.

[6] Mccormickml.com. 2022. Word2Vec Tutorial - The Skip-Gram Model·Chris McCormick. [online] Available at: <http://mccormickml.com/2016/04/19/word2vectutorial-the-skip-gram-model/>.

[7] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27 (2014).

[8] Kyunghyun Cho, Bart van Merriënboer, DzmitryBahdanau, and YoshuaBengio. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

[9] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. 2014 Dec 11.

[10] Cho, Kyunghyun, Bart Van Merriënboer, CaglarGulcehre, DzmitryBahdanau, FethiBougares, Holger Schwenk, and YoshuaBengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL '02). Association for Computational Linguistics, USA, 311–318. https://doi.org/10.3115/1073083.1073135.

[12] Coursera. 2022. [online] Available at: <https://www.coursera.org/learn/attention-models-in-nlp>.

[13] Bahdanau, Dzmitry, Kyunghyun Cho, and YoshuaBengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[14] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and IlliaPolosukhin. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[15] Elmogy, Ahmed, Belal Mahmoud, and Mohamed Saleh. "A Deep Learning Approach for Text Generation." 2019 29th International Conference on Computer Theory and Applications (ICCTA). IEEE, 2019.

[16] Suleiman, Dima, and Arafat A. Awajan. "Deep learning based extractive text summarization: approaches, datasets and evaluation measures." 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). IEEE, 2019.

[17] Shirwandkar, Nikhil S., and Samidha Kulkarni. "Extractive text summarization using deep learning." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[18] Rezaei, Afsaneh, SinaDami, and Parisa Daneshjoo. "Multi-document extractive text summarization via deep learning approach." 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI). IEEE, 2019.

[19] Chen, Yisong, and Qing Song. "News text summarization method based on bart-textrank model." 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). Vol. 5. IEEE, 2021.

[20] Pabbi, Kethan, and C. Sindhu. "Opinion Summarisation using Bi-Directional Long-Short Term Memory." 2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, 2021.