Automated Abstractive Text Summarization Using Multidimensional Long Short-Term Memory

Anwar Khan, Ibrar Ahmad*, Qazi Ejaz Ali, Muhammad Owais Khan, Umair Sana

Department of Computer Science, University of Peshawar, Peshawar, KPK, Pakistan
*Corresponding author: ibrar@uop.edu.pk

Abstract

Text summarization is a technique for condensing and presenting the most crucial information from a larger text in a succinct manner. It is a challenging task for human beings to manually summarize extensive documents due to the time required and cognitive effort involved. The increasing amount of textual data generated in recent years has led to a greater demand for efficient and automated text summarization methods. This study explores the utilization of deep learning and neural network concepts to create an automatic abstractive text summarization system for product reviews using multi-dimensional long short-term memory. Since, reading product reviews can be a time-consuming task, especially given the abundance and diversity of reviews written by numerous individuals. The proposed model was evaluated by comparing the generated summaries to reference summaries using metrics such as Rouge. Further, the performance of the model is compared with the prevalent state-of-the-art techniques. The proposed Multi-Dimensional LSTM model was evaluated using the Amazon food review dataset and compared with four other text summarization methods. The proposed model outperformed the prevalent models in terms of ROUGE-1 and ROUGE-L scores but was slightly outperformed in terms of the ROUGE-2 score.

Keywords—Machine learning, Deep learning, NLP, RNN, LSTM, Multi-LSTM

1 Introduction

Natural Language Processing (NLP) constitutes a significant domain, focusing on the computational analysis of natural languages, frequently employing machine learning methodologies. Text summarization involves the automated condensation of crucial information from a source text into a concise format. NLP techniques are used in multiple tasks, such as automatic translation, information retrieval, and text simplification. The realm of NLP has seen advancements in automatic summarization methodologies, encompassing diverse approaches like abstractive, extractive, single-document, multi-document, genre-based, function-based, and context-based techniques [1]. Summarization broadly categorizes into two main types, abstractive and extractive, each characterized by distinct methods and strategies [2].

Extractive summarization uses statistical attributes such as term frequency, position, relevance to the title, sentence length, proper nouns, text similarities, and headlines to create a summary. This method identifies crucial content from the original text through statistically calculated rankings derived from linguistic features, graph features, clustering, and machine learning [3, 4, 5, 6]. Despite being simpler to implement than abstractive methods, successful extractive summarization utilizes diverse techniques and graph-based strategies, typically highlighting significant sentences from a single document [7].

Conversely, abstractive approaches involve crafting sentences using novel words or phrases to convey the origi-nal concepts from the document. Crafting abstractive summaries poses greater challenges compared to extractive methods due to the necessity for a wider array of natural language processing techniques [8]. The two primary categories within abstractive summarization include structural and semantic methods. Structural methodologies encompass diverse approaches such as rule-based, tree-based, graph-based, and lead body phrase techniques [9-12]. Semantic methods encompass a variety of approaches including multi-model semantic summarization [13], Lex-Rank

ISSN: 2523-0379 (Online), ISSN: 1605-8607 (Print)
DOI: https://doi.org/10.52584/QRJ.2102.01

method-based [11], Text-To-Text framework akin to BERT architecture [14], encoder-decoder based [16], and sequence-to-sequence techniques [17, 18]. Preprocessing tasks such as parsing, recording sentence and word position, stop word removal and stemming play pivotal roles in abstractive summarization [15].

The goal of abstractive summarization is to condense the overall meaning and context into brief and coherent sentences [9, 21, 61]. Methods such as semantic-graph-based reduction, including Graph-based, Semantic graph, and rich-semantic-graph (RSG), are used for summarizing individual input documents [19]. In abstractive multi-document text summarization, semantic-based techniques are utilized to condense information from multiple documents [10].

Recurrent Neural Network (RNN) models emerge as effective tools for generating abstractive summaries. RNNs retain context from prior text inputs and amalgamate them with current inputs to produce cohesive output text. While standard RNNs possess short-term memory, enhanced versions like Long-Short-Term Memory (LSTM) models exhibit more sophisticated functionalities [4].

Another method, multilayered attention-based process and composition of LSTMs (MAPCoL) is used to automatically produce abstractive summaries of massive written content [20, 36].

Although large pretrained-based models outperformed in most cases as observed in recent advancements in summarization but often considered as inefficient. In this study, we proposed an abstractive summarization technique based on multidimensional LSTM with an attention layer that tend to outperform the prevalent massive inefficient encoder-decoder systems.

2 Literature Review

Text summarization in Natural Language Processing (NLP) involves analyzing human language computationally, a field that has recently received significant attention. When classifying text summarization, various factors are considered, including function, category, summary framework, summarizer type, and document quantity. Generally, text summarization can be categorized into two primary groups, extractive and abstractive methods [2].

In extractive summarization, statistical methods are used, considering elements like term frequency, position, text similarities, title relevance, key phrases, sentence count, proper nouns, and proximity. These techniques aim to identify essential components within provided texts by utilizing calculated rankings derived from linguistic or statistical factors to extract segments from the original content [3]. For instance, the sentence-based graph technique assesses the significance of sentences, phrases, or words using predetermined methods to select critical sentences for inclusion in the summary [4, 50, 64, 65, 66, 75].

A cluster-based approach using machine learning is used for extractive summary [6]. Single-page extractive summaries that only include the key phrases from the input [8]. Extractive summarization has several drawbacks, including loss of information, inflexibility, limited creativity, bias, and lack of coherence. These limitations can result in a lack of accuracy and comprehensiveness in the summary and a distorted representation of the original text.

In abstractive summarization, new sentences are generated to create a summary, which may not be present in the original text. This is challenging when dealing with abstract concepts. Compared to extractive summarization, abstractive summarization requires advanced natural language processing [9]. Some important stages of abstractive summarization include preprocessing, sentence position, word position, stop-word removal, and stemming [15]. Abstractive summarization encompasses two primary approaches, structured and semantic-based approaches. Structured-based techniques involve a range of techniques such as rule-based, tree-based, lead and body phrase, as well as graph-based techniques [10, 11, 23-25].

Semantic-based approaches include methods such as multi-modal, Lex-Rank Method based unsupervised graph method, info item-based technology, sequence-to-sequence models, Rich Semantic Graph (RSG), and BERT based techniques [24, 16, 18, 19, 39, 40, 58].

Semantic summarization [87] covers various modalities such as text, image, audio, and video. The Lex-Rank Method is an example of text summarization using an unsupervised graph method [24]. The semantic text interpretation model utilizes item-based information for semantic-based abstractive summarization [14, 63]. An encoder-decode based on a neural network for creating summarized text [16]. For abstractive multi-document text summarization, a semantic-based technique [10, 77] makes use of several documents. Abstractive methods also utilize the popular sequence-to-sequence models [18, 67, 83]. Utilizing an ontology-based approach, we craft precise and well-structured summaries [26]. In the realm of abstractive summarization, employing techniques like semantic graphs and simplification methods suggests employing the Rich Semantic Graph (RSG) for effectively summarizing individual documents [19]. An influential approach within abstractive summarization involves neural networks, specifically the Recurrent Neural Network (RNN) [80]. The RNN's adeptness at retaining information significantly enhances the depth of its summaries, leveraging its long-term memory, particularly within the Long Short-Term Memory (LSTM) module [27]. This model employs a graph-based technique to prioritize crucial sentences.

Given the continuous proliferation of information across various sectors such as medical texts [28], news articles [29], financial data [30], and business dialogues aimed at enhancing products and services [31, 70], the demand for summarizing multiple documents is burgeoning. Text summarization broadly encompasses two primary strategies, general summarization and targeted question-answering. General summarization aims to provide an overview, akin to the concise responses seen in search engines like Google or Yahoo! Conversely, targeted question-answering focuses on delivering precise responses [32, 48, 53]. This method seeks relevant information based on specific queries and consolidates related details, often referred to as query-oriented or topic-oriented summaries [33].

The central objective of this research is to distill succinct and lucid summaries from extensive product reviews using deep learning techniques. Despite the remarkable performance of advanced models like BERT, T5, and PEGASUS in linguistic tasks, concerns arise about their potential resource-intensive nature [22, 34, 35]. Additionally, current text summarization systems struggle with recurring information, highlighting the importance of devising effective approaches to curb repetition during the decoding process [37]. The summary of text summarization techniques is given in Table 1.

In literature, summarization can be performed by well-defined architecures and algorithms like artificial neual network [44], [45], Naive Bayes [47], clustering [49, 51], Support Vector Machines [52], Automatic Key Phrase Ex-traction [55], [56], Sentence Extraction [57], Latent Semantic Analysis (LSA) [59, 81], Encoder-Decoder Models [7], Recurrent Neural Networks [68], Long Short-Term Memory [69], Gated Recurrent Unit (GRU) [71], Transformers [72, 73, 79], BERT (Bidirectional Encoder Representations from Transformers) [72], The Text-To-Text Transfer Transformer (T5) [35, 38], BART (Bidirectional and Auto-Regressive Transformers) [74], Reformer [78] and Rhetorical Structure Theory (RST) [62] etc.

Creating short summaries from text is tough and needs special deep learning systems, unlike many other things in language technology [60]. The best systems need lots of training, like Google's PEGASUS, which learned from a huge dataset of 1.5 billion articles, taking up 3.8 terabytes [35]. Using these systems for summarizing text needs lots of computer power and memory. In our research, we're introducing a new way to summarize text. It uses a kind of smart memory system called LSTM and focuses on important parts, aiming to do better than the systems used a lot now, which need a ton of resources.

3 Methodology

Automatic text summarization systems are crucial for efficiently managing large volumes of written documents like reviews in this study [46]. Such systems are developed to summarise in a concise, logical with high accuracy while encompassing all important information of a document. This study starts with a very fundamental type of RNN called one-dimensional Long Short-Term Memory (LSTM). It is followed by an LSTM model called Multidi-mensional LSTM (MD-LSTM) with detailed topological adjustments for carefully and thoroughly examining the review documents.

The proposed methodology for automatic text summarization using a Multidimensional LSTM can be divided into five distinct stages. These stages include data collection, preprocessing, encoding, and attention mechanism (for focusing on important information) followed by generating the final summary. In the data collection phase, we gather the relevant information from sources like Kaggle and Google. It is followed by the step of cleaning and refining the text by removing unnecessary parts and normalizing it. The normalized and preprocessed text is then passed through an encoder and an attention mechanism for recognizing important details. Finally, a decoder equipped with LSTM layers produces the final summary. Overall, the proposed technique aims to collect, prepare, encode, apply attention, and decode the input text to create an accurate and concise summary. The step-by-step approach of the proposed model is illustrated in Figure 1. Data Collection Phase: The data collection phase involves gathering a relevant dataset named "Amazon Fine Foods Reviews" from the Kaggle website. The dataset contains Amazon food reviews spanning over a decade, totaling approximately 500,000 reviews.

- **Preprocessing**: The preprocessing involves addressing various aspects such as emotions, punctuation, and the consideration of text in both upper and lower cases. These elements can be considered as noise within the text data, and to enable machines to comprehend and work with this data, it is imperative to appropriately convert

TABLE 1: Text Summarization Techniques

<table>
<tr><th>Reference</th><th>Technique Used</th><th colspan="2">Type of Summarization</th></tr>
<tr><td>[3]</td><td>Ranking algorithm</td><td colspan="2" rowspan="4">Extractive</td></tr>
<tr><td>[4]</td><td>Sentence-based graphs</td></tr>
<tr><td>[6]</td><td>Cluster-based approach</td></tr>
<tr><td>[8]</td><td>Key Phrase-based approach</td></tr>
<tr><td>[10]</td><td>Rule-based approach</td><td rowspan="14">Abstractive</td><td rowspan="4">Structured based approaches</td></tr>
<tr><td>[11]</td><td>Tree-based approach</td></tr>
<tr><td>[19]</td><td>Lead and body phrase approach</td></tr>
<tr><td>[23-25]</td><td>Graph-based approach</td></tr>
<tr><td>[88]</td><td>Multi-modal approaches</td><td rowspan="10">Semantic-based approaches</td></tr>
<tr><td>[24]</td><td>Lex-Rank Method</td></tr>
<tr><td>[13]</td><td>BERT</td></tr>
<tr><td>[14]</td><td>Semantic graph-based approach</td></tr>
<tr><td>[16]</td><td>Encoder-decoder approach</td></tr>
<tr><td>[18]</td><td>Sequence-to-sequence models</td></tr>
<tr><td>[19]</td><td>Rich Semantic Graph (RSG)</td></tr>
<tr><td>[27]</td><td>Long Short-Term Memory (LSTM)</td></tr>
<tr><td>[32]</td><td>Multi-Document Query-based Technique</td></tr>
<tr><td>[26]</td><td>Multi-Document Ontology-based method</td></tr>
</table>

the text into a form suitable for further processing. In the study, the preprocessing involves case folding (lower case), tokenization (breaking down the text to words), removing punctuations (eliminate them or replace them with an empty string), removing stop words (as least informative), lemmatization (reducing to lemmas), and normalizing the text into a format suitable for onward stages.

- **Word and Sentence Embeddings:** Word and sentence embeddings convert words and sentences into numerical vectors [82]. Word embeddings try to de-note similar words with a very proximate vector representation. Sentence embeddings aim to pick relevant source sentences, ensure accurate rephrasing, maintain flow, and avoid redundancy.
- **Encoder Stage:** The encoder stage with LSTMs performs a sequential encoding of the input tokens while updating the hidden state at each unit of LSTM. The relationships among words and sentences that are contextual information are efficiently captured by hidden states. Therefore, the final core context is generated by the final hidden state put forward to the attention and decoder stages to draw a coherent and precise abstractive summary.
- **Attention Mechanism:** The attention mechanism phase of the model focuses on multiple parts of the source to keep important details. With an encoder-decoder setup by LSTM, an attention layer preserves the important information for long source sequences. The attention process uses attention weights to focus more on relevant input embeddings that contain significant information. The scaled dot-product attention is used by this model to effectively capture the relationship among different input embeddings to capture the intended context.
- **Decoder Stage:** The decoder starts with an initial state with the aid of attention's context for recognizing the source text's significance empowering it to focus on the most relevant information. In light of this prior information, it starts generating word-by-word predictions based on the context. LSTM layers update this generation throughout until the de-coder reaches an endpoint with the completion of the summary.

4 Results and Discussions

This section provides an elaborate account of the conducted experiments employing various model approaches. It encompasses a comprehensive description of all completed experiments, elucidating the outcomes corresponding to each model. Firstly, it delves into the specifics of the experiment employing one-dimensional

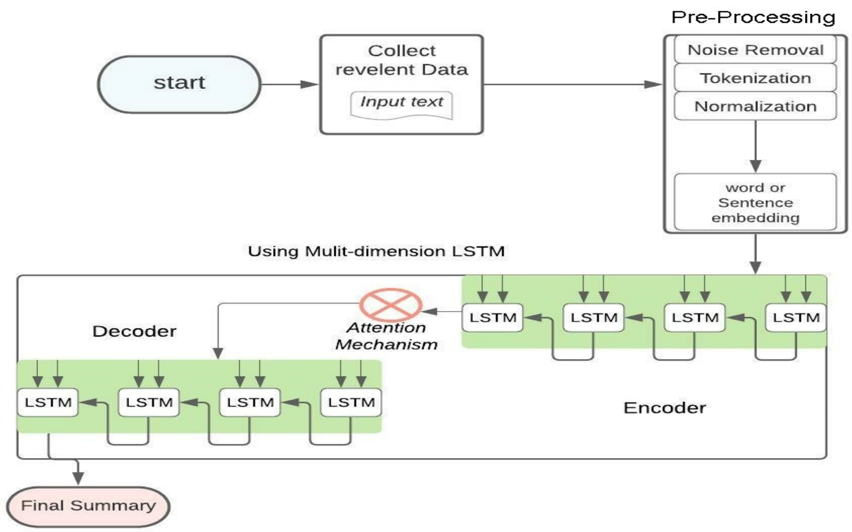


Fig. 1: The Proposed Multidimensional LSTM-based Summarization Methodology

LSTM. Subsequently, it outlines the construction and analysis of the Multilayer-LSTM utilizing the MD-LSTM mechanism.

4.1 Hardware Specification

The proposed methodology can operate on various platforms, including the hardware specifications provided, which include 12 GB RAM, a 4th generation core i7 processor, and a 2GB graphics card. Additionally, the methodology does not have any specific hardware requirements, but it is recommended to use machines with sufficient RAM and graphics processing capabilities to achieve optimal performance. The provided hardware specifications are well-suited for executing the method, and they should be capable of meeting the GPU and RAM requirements.
However, for this study, the algorithm is being executed on Google Colab, which is a cloud-based service that eliminates the need for local hardware resources. With Google Colab, all the processing and training of the algorithm are performed on Google's servers, making it a cost-effective and convenient solution for executing the algorithm. All that is required is a stable internet connection and a device to connect to the internet, such as a laptop or PC.

4.2 Training with Parameters Optimization

The model's training hyper-parameters encompass various elements such as learning rate and its decay rate, step size, initial learning rate, number of epochs, batch size, and additional training and optimization factors. Table 2 provides the configuration details for specific parameters, along with short descriptions of their dimensions.

Further elaboration is needed for techniques like dropout, degrade, and maximum grad norms, which will be detailed below,

4.2.1 Dropout

Overfitting occurs when a model becomes too specialized in learning a particular dataset, resulting in high accuracy but difficulty in generalizing to new test data, causing a considerable drop in validation accuracy. This issue is widespread in machine learning. Dropout, a neural network technique, acts as a form of regularization to combat overfitting. In this study, the dropout used is 0.4. Normally, it is recommended for abstractive summarization models as it provides a balance between regularization and model capacity.

4.2.2 AdaGrad

AdaGrad (Adaptive Gradient Descent) is well-known for its adaptive learning rates, which adjust for each parameter based on past gradients, and is used in this study instead of SGD (Stochastic Gradient Descent). SGD is a popular optimization technique for reducing cost functions but it uses the derivatives of the network weights to update them at a single learning rate. AdaGrad performs efficiently when the data is not dense. It reduces the learning rate automatically avoiding extra effort for extensive manual learning rate tuning. These automatic tuning and optimization of hyperparameter tuning and model training save time and effort.

TABLE 2: Training Model Overview

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[None, 30)]	0	
embedding (Embedding)	(None, 30, 100)	844000	input_1[0][0]
lstm (LSTM)	[None, 30, 300),	(N 481200)	embedding[0][0]
input_2 (Input Layer)	[(None, None)]	0	
lstm_1 (LSTM)	[None, 30, 300),	(N 721200)	lstm[0][0]
embedding_1 (Embedding)	(None, None, 100)	198900	input_2[0][0]
lstm_2 (LSTM)	[None, 30, 300),	(N 721200)	lstm_1[0][0]
lstm_3 (LSTM)	[None, None, 300),	481200	embedding_1[0][0] lstm_2[0][1] lstm_2[0][2]
attention_layer (AttentionLayer)	(None, None, 300),	180300	lstm_2[0][0] lstm_3[0][0]
concat_layer (Concatenate)	(None, None, 600)	0	1stm_3[0][0] attention_layer[0][0]
time_distributed (TimeDistribution)	(None, None, 1989)	1195389	concat_layer[0][0]
Total Params: 4823389 Trainable params: 4823389 Non-trainable params: 0			

4.2.3 *Maximum Gradient Norms*

In abstractive text summarization, maximum gradient norms play a main role in the training of models featuring LSTM-based encoders, attention mechanisms, and decoders. These norms are essential for managing the size of gradients during backpropagation for dealing with the risk of exploding gradients. A typical range to consider for the predetermined maximum gradient norm (max gradient norm) hyperparameter in models with LSTM-based encoders, attention mechanisms, and decoders is between 1.0 and 5.0. In this study, we set it to a value of 1.0. During training, the L2 norm is kept under observations, if it exceeds the set maximum gradient norm, then we increase its value from 1.0 to 2.0. One can check suitable values by using cross-validation, aiding in selecting the optimal value for the specific summarization task.

4.3 A Dataset of Amazon Food Reviews

The dataset comprises Amazon reviews [86] focused on high-quality food items. Spanning over a decade, it encompasses a comprehensive collection of 500,000 reviews until October 2019. These evaluations encompass details regarding the products and their users, ratings, a textual review, and a summary.

Additionally, the dataset includes feedback from various categories on Amazon. Each review, averaging 75 words, contains a user-generated title that acts as a summary of the entire review. These summaries are concise. Figure 2(A)(B) illustrates the distribution of text and summary lengths in Amazon reviews.
Before initiating the model construction phase, it's crucial to perform essential preprocessing tasks. Incorporating inaccurate or irrelevant content details can significantly impact the model's effectiveness. Therefore, eliminating unnecessary symbols, characters, and similar elements from the text that don't contribute to the model's primary objectives is imperative.

4.3.1 *Dropping Useless Values*

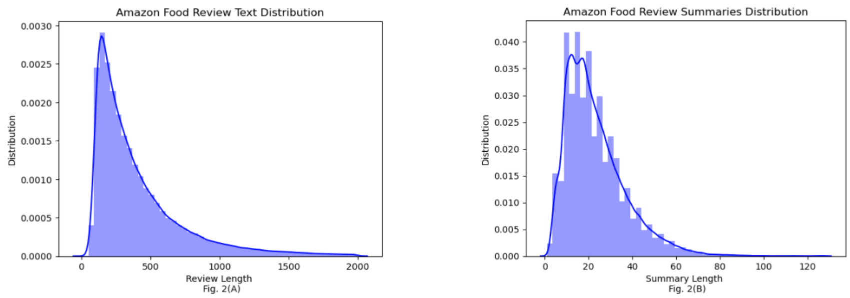
These datasets undergo a cleanup process using the drop command to remove irrelevant values and entities that are not necessary for modeling in the datasets. Before initiating the model construction phase, it's crucial to execute this essential preprocessing task.

4.3.2 *Contractions Mapping of Words*

For our data, we'll execute the preprocessing steps outlined in the methodology section. These include converting everything to lowercase, mapping contractions, removing HTML tags, deleting, eliminating text within parentheses, getting rid of all punctuation and special characters, removing stop words, and discarding short words.

4.3.3 *Deletion of Empty Rows*

The 'dropna' function in data processing serves to remove rows or columns containing missing or 'NaN' (Not a Number) values. In machine learning datasets,



(a) (b)

Fig. 2: Amazon Reviews with Stop-Words and without Stops-Words Graphs

these missing values can disrupt model training and prediction accuracy. By employing 'dropna', the dataset is cleaned, resulting in a more organized and consistent dataset conducive to machine learning modeling. Nevertheless, it's crucial to be mindful of the implications of data removal, as excessive elimination might result in the loss of valuable information.

5 Model Training

This study focuses mainly on the Multi-Dimensional Long Short-Term Memory (MD-LSTM), an attention-based design that replaces the traditional RNN layers in sequential models. The experiment includes training models using a batch size of 128 for 15 epochs and assessing their performance on a separate holdout set, which accounts for 10% of the dataset for each epoch.

5.1 Model Testing

The testing process in our study involves assessing the effectiveness of our model using the ROUGE evaluation method [41, 42, 43, 84, 85]. This method, known as Recall-Oriented Understudy for Gisting Evaluation (ROUGE), is a standard technique used to compare automatically generated summaries against human-created ones [87]. To do this, the ROUGE toolbox measures how many words in the system-generated summary match those in the human-created summary within our holdout set. ROUGE-1 score focuses on the presence of individual words in both sys-tem-generated and human summaries, while ROUGE-2 considers overlaps between pairs of words.

We have chosen to use a multi-dimensional LSTM, which is different from the typical LSTM approach used in most studies. Many researchers combine LSTM with other deep-learning models to get better results. In our study, we compare the performance of our model with some widely used baseline models, using the Amazon fine food review dataset. This dataset is established and widely recognized in the field. We then assess our model's performance using ROUGE-1, ROUGE-2, and ROUGE-L scores, which respectively evaluate summaries based on single words and pairs of words.

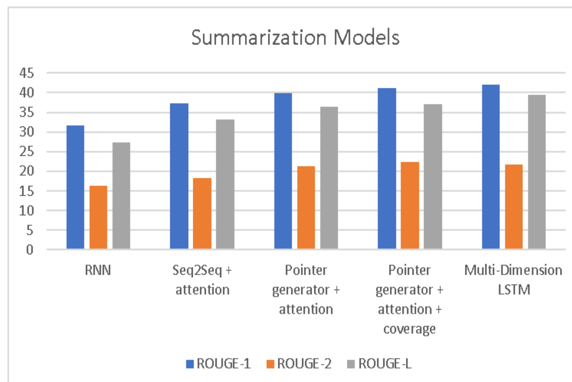


Fig. 3: Comparison of Summarization Models

Table 3 provides a detailed comparison of the ROUGE scores generated by four different text summarization methods, RNN, Seq2Seq + attention, Pointer generator + attention, and Pointer generator + attention + coverage. All these methods leverage the Multi-Dimensional LSTM strategy we've implemented. This allows us to see how our approach stacks up against these established methods in the domain of text summarization.

Table 3 illustrates the top attained scores for ROUGE-1, ROUGE-2, and ROUGE-L across all models under consideration. Notably, the Multi-Dimensional model exhibits higher ROUGE-1, ROUGE-2, and ROUGE-L scores compared to the conventional model, except for the ROUGE-2 score, where the Pointer generator + attention + coverage model achieves a slightly higher score of 0.73. These findings are visually represented in Figure 3 using a bar chart.

In this scenario, we evaluate the proposed system against three well-known classical methods, RNN, a

TABLE 3: Outcomes Assessment of the Proposed Model Against Established Techniques

References	Summarization Models	R-1	R-2	R-L
[67]	Recurrent Neural Network	31.6	16.24	27.32
[16]	Seq2Seq with attention	37.33	18.31	33.15
[76]	Pointer generator with attention	39.8	21.2	36.4
[89]	Pointer generator + attention + coverage	41.21	22.4	37.1
[In this study]	Multi-Dimension LSTM	41.98	21.67	39.84

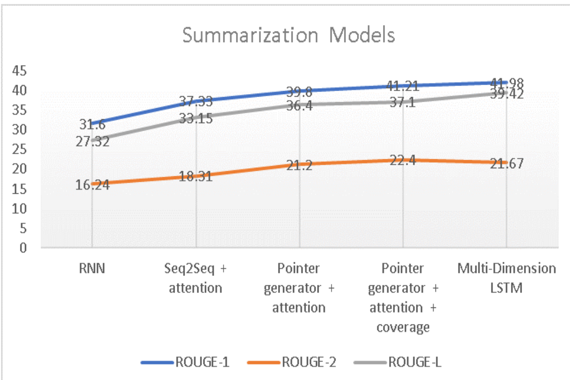


Fig. 4: Comparison of Summarization Models

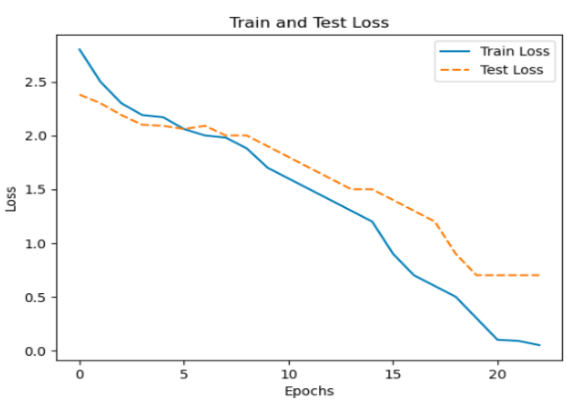


Fig. 5: Comparison of the Training and Validation Losses

sequence-to-sequence model incorporating an attention layer, a pointer generator using an attention mechanism, and the Multi-Dimensional LSTM. The results of this assessment are depicted in Figure 4.

Figure 4 depicts the training and validation loss computed across different epochs. Meanwhile, Figure 5 illustrates that both losses consistently decline as expected until epoch 19, after which they gradually begin to rise. This pattern beyond epoch 19 suggests a potential for overfitting.

5.2 Review Summarization Analysis

The review summary of the proposed system is generated based on customer reviews, and the original summary versus generated summaries are evaluated to assess their quality and excellence.
Table 4 displays sample summaries generated by the proposed models, labeled as R1, R2, and R3, randomly chosen from newly created summaries. The table presents a comparison between the original reviews and the generated text summaries. These reviews have undergone processing. R1's review has been condensed into a brief sentence, improving the model's context by system-generated concise summarization. Similarly, R2 and R3 have undergone similar treatments. Notably, the summary generated for R3 is superior as it explicitly mentions containing excellent information, unlike the original summary, which merely states 'Great food'.

6 Conclusion

The advancements in this area aim to improve text summarization methods. However, current technologies have limitations that could be addressed by introducing more effective solutions. Using multidimensional LSTM with attention mechanisms shows promise in significantly improving text summarization compared to other deep learning methods.
Testing our model using the Amazon Fine Food Review dataset from Kaggle revealed that the multidimensional LSTM outperformed regular LSTM-based models. Yet, there's a need to assess critical elements like factual accuracy, fluency, coherence, and relevance in summaries crafted by human experts. Developing enhanced evaluation algorithms that surpass current measures to capture the most essential qualities of summaries is imperative.

References

[1] H. P. Luhn, "The automatic creation of literature abstracts," IBM Journal of Research and Development, vol. 2, no. 2, pp. 159-165, Apr. 1958.

TABLE 4: Comparison of Original & Generated Summaries

S. No.	Review	Original summary	Generated summary by MDLSTM
R1	First tried glass straight strong aromatic coffee smell good dark color strong coffee flavor hint sweetness next heated cup microwave consistent results surprisingly good canned coffee	Tasty hot or cold	Great coffee at great price
R2	Meat flavor feel could little also slices thin burn quick don't pay attention cucumber sause place amazing though	Not the best not the worst	Not as good as expected
R3	Hard find free gluten soy corn daughter eat shaped cereal find one yummier others better ingredients hard find strong happy see amazon happy buy pack	Yummy	Great gluten free snack mix

[2] D. Suleiman and A. Awajan, "Deep learning based abstractive text summarization, approaches, datasets, evaluation measures, and challenges," Mathematical Problems in Engineering, vol. 2020, Feb. 2020.

[3] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms," Cognitive Computation, vol. 10, no. 4, pp. 651-669, Jul. 2018.

[4] I. Mani and M. T. Maybury, "Advances in automatic text summarization," Camb MA, vol. 293, Aug. 1999.

[5] S. Yadav and D. Shah, "News summarization using text mining," International Research Journal of Engineering and Technology, vol. 5, no. 11, pp. 202-206, Nov. 2018.

[6] M. Subramaniam and V. Dalal, "Test model for rich semantic graph representation for Hindi text using abstractive method," International Research Journal of Engineering and Technology (IRJET), vol. 2, no. 2, pp. 113-116, Feb. 2015.

[7] D. Suleiman and A. A. Awajan, "Deep learning based extractive text summarization, approaches, datasets and evaluation measures," in Proc. IEEE, 2019, pp. 204-210.

[8] A. Nenkova and K. McKeown, "A survey of text summarization techniques and Mining text data," Springer, pp. 43-76, 2012.

[9] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques, a survey," Artificial Intelligence Review, vol. 47, no. 1, pp. 1-66, Jan. 2017.

[10] D. Bartakke, S. Sawarkar, and A. Gulati, "A semantic based approach for abstractive multi-document text summarization," International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 7, pp. 13620-13629, Jul. 2016.

[11] E. D. Liddy, "Natural language processing," surface.edu, 2001.

[12] A. R. Fabbri et al., "Multi-news, A large-scale multi-document summarization dataset and abstractive hierarchical model," arXiv preprint arXiv:1906.01749, 2019.

[13] Y. Liu and M. Lapata, "Text summarization with pretrained encoders," arXiv preprint arXiv:1908.08345, 2019.

[14] A. Alzuhair and M. Al-Dhelaan, "An approach for combining multiple weighting schemes and ranking methods in graph-based multidocument summarization," IEEE Access, vol. 7, pp. 120375-120386, 2019.

[15] D. Patel, S. Shah, and H. Chhinkaniwala, "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique," Expert Systems with Applications, vol. 134, pp. 167-177, Dec. 2019.

[16] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," arXiv preprint arXiv:1509.00685, 2015.

[17] T. Shi et al., "Neural abstractive text summarization with sequence-to-sequence models," ACM Transactions on Data Science, vol. 2, no. 1, pp. 1-37, Feb. 2021.

[18] R. Nallapati et al., "Abstractive text summarization using sequence-to-sequence rnns and beyond," arXiv preprint arXiv:1602.06023, 2016.

[19] I. F. Moawad and M. Aref, "Semantic graph reduction approach for abstractive Text Summarization," IEEE, pp. 132-138, 2012.

[20] M. M. Rahman and F. H. Siddiqui, "Multi-layered attentional peephole convolutional LSTM for abstractive text summarization," ETRI Journal, vol. 43, no. 2, pp. 288-298, Apr. 2021.

[21] C. Sunitha, A. Jaya, and A. Ganesh, "A study on abstractive summarization techniques in Indian languages," Procedia Computer Science, vol. 87, pp. 25-31, 2016.

[22] G. Sun et al., "Automatic Text Summarization Using Deep Reinforcement Learning and Beyond," Information Technology and Control, vol. 50, no. 3, pp. 458-469, 2021.

[23] P. F. Brown et al., "A statistical approach to machine translation," Computational linguistics, vol. 16, no. 2, pp. 79-85, 1990.

[24] J. Eisner, "Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)," Book Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.

[25] D. Wang et al., "Integrating document clustering and multidocument summarization," ACM Transactions on Knowl-

edge Discovery from Data (TKDD), vol. 5, no. 3, pp. 1-26, Aug. 2011.

[26] M. J. Mohan et al., ”A study on ontology based abstractive summarization,” Procedia Computer Science, vol. 87, pp. 32-37, 2016.

[27] N. Moratanch and S. Chitrakala, ”A survey on abstractive text summarization,” IEEE, pp. 1-7, Jul. 2016.

[28] K. Sarkar, ”Using domain knowledge for text summarization in medical domain,” International Journal of Recent Trends in Engineering, vol. 1, no. 1, pp. 200, 2009.

[29] Y. J. Kumar and N. Salim, ”Automatic multi document summarization approaches,” Citeseer, 2012.

[30] E. Cardinaels, S. Hollander, and B. J. White, ”Automatic summarization of earnings releases, attributes and effects on investors' judgments,” Review of Accounting Studies, vol. 24, no. 3, pp. 860-890, Sep. 2019.

[31] A. Tamura et al., ”Extractive summarization method for contact center dialogues based on call logs,” Book Extractive summarization method for contact center dialogues based on call logs, pp. 500-508, 2011.

[32] M. Litvak and N. Vanetik, ”Query-based summarization using MDL principle,” Book Query-based summarization using MDL principle, pp. 22-31, 2017.

[33] D. Sahoo et al., ”Aspect based multi-document summarization,” IEEE, pp. 873-877, Mar. 2016.

[34] J. Devlin et al., ”Bert, Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[35] C. Raffel et al., ”Exploring the limits of transfer learning with a unified text-to-text transformer,” arXiv preprint arXiv:1910.10683, 2019.

[36] A. Cohan et al., ”A discourse-aware attention model for abstractive summarization of long documents,” arXiv preprint arXiv:1804.05685, 2018.

[37] M. Abdolahi and M. Zahedh, ”Sentence matrix normalization using most likely n-grams vector,” IEEE, pp. 0040-0045, Oct. 2017.

[38] J. L. P. Espejel, ”Automatic abstractive summarization of long medical texts with multi-encoders Transformer and general-domain summary evaluation with wikiSERA,” Université Paris-Nord-Paris XIII, 2021.

[39] A. See et al., ”Get to the point, Summarization with pointer-generator networks,” arXiv preprint arXiv:1704.04368, 2017.

[40] R. Mitkov, ”The Oxford handbook of computational linguistics,” Oxford University Press, 2004.

[41] C.-Y. Lin, ”Rouge, A package for automatic evaluation of summaries,” Book Rouge, A package for automatic evaluation of summaries, pp. 74-81, 2004.

[42] L. A. Cabrera-Diego and J.-M. Torres-Moreno, ”Summtriver, A new trivergent model to evaluate summaries automatically without human references,” Data & Knowledge Engineering, vol. 113, pp. 184-197, Apr. 2018.

[43] A. Cohan and N. Goharian, ”Revisiting summarization evaluation for scientific articles,” arXiv preprint arXiv:1604.00400, 2016.

[44] W. S. McCulloch and W. Pitts, ”A logical calculus of the ideas immanent in nervous activity,” The bulletin of mathematical biophysics, vol. 5, no. 4, pp. 115-133, Dec. 1943.

[45] R. M. Alguliyev et al., ”COSUM, Text summarization based on clustering and optimization,” Expert Systems, vol. 36, no. 1, pp. e12340, Feb. 2019.

[46] J. K. Yogan et al., ”A review on automatic text summarization approaches,” Journal of Computer Science, vol. 12, no. 4, pp. 178-190, 2016.

[47] J. Kupiec et al., ”A trainable document summarizer,” Book A trainable document summarizer, pp. 68-73, 1995.

[48] N. Ramanujam and M. Kaliappan, ”An automatic multidocument text summarization approach based on Naive Bayesian classifier using timestamp strategy,” The Scientific World Journal, vol. 2016, Dec. 2016.

[49] C. D. Manning and H. Schütze, ”Foundations of statistical natural language processing,” MIT Press, 1999.

[50] R. M. Aliguliyev, ”A new sentence similarity measure and sentence based extractive technique for automatic text summarization,” Expert Systems with Applications, vol. 36, no. 4, pp. 7764-7772, May 2009.

[51] K. Shivakumar and R. Soumya, ”Text summarization using clustering technique and SVM technique,” International Journal of Applied Engineering Research, vol. 10, no. 12, pp. 28873-28881, Aug. 2015.

[52] C. J. C. H. Watkins and P. Dayan, ”Q-learning,” Machine Learning, vol. 8, no. 3-4, pp. 279–292, May 1992.

[53] C. Schilder and R. Kondadadi, ”Fastsum, Fast and accurate query-based multi-document summarization,” Book Fastsum, Fast and accurate query-based multi-document summarization, pp. 205-208, 2008.

[54] N. Begum et al., ”Automatic text summarization using support vector machine,” International journal of innovative computing information and control, vol. 5, no. 7, pp. 1987-1996, Sep. 2009.

[55] K. S. Hasan and V. Ng, ”Automatic keyphrase extraction, A survey of the state of the art,” Book Automatic keyphrase extraction, A survey of the state of the art, pp. 1262-1273, 2014.

[56] H. H. Alrehamy and C. Walker, ”SemCluster, unsupervised automatic keyphrase extraction using affinity propagation,” Springer, pp. 222-235, 2017.

[57] R. Mihalcea and P. Tarau, ”Textrank, Bringing order into text,” Book Textrank, Bringing order into text, pp. 404-411, 2004.

[58] G. Erkan and D. Radev, ”Lexpagerank, Prestige in multi-document text summarization,” Book Lexpagerank, Prestige in multi-document text summarization, pp. 365-371, 2004.

[59] ure and latent semantic analysis,” Book Generic text summarization using relevance measure and latent semantic analysis, pp. 19-25, 2001.

[60] P. Anttila, ”Automatic Text Summarization,” Book Automatic Text Summarization” Diplomová práce, Univerzita v Turku, Fakulta budoucích technologií, pp, 2018.

[61] W. C. Mann and S. A. Thompson, ”Rhetorical structure theory, Toward a functional theory of text organization,” Text-interdisciplinary Journal for the Study of Discourse, vol. 8, no. 3, pp. 243-281, Sep. 1988.

[62] V. R. Uzêda et al., ”A comprehensive comparative evaluation of RST-based summarization methods,” ACM Transactions on Speech and Language Processing (TSLP), vol. 6, no. 4, pp. 1-20, Aug. 2010.

[63] A. Louis et al., ”Discourse Indicators for Content Selection in Summarization,” 2010.

[64] R. Kiros et al., ”Skip-thought vectors,” arXiv, 2020.

[65] A. Conneau et al., ”Supervised learning of universal sentence representations from natural language inference data,” arXiv, 2017.

[66] D. Cer et al., ”Universal sentence encoder,” arXiv, 2018.

[67] I. Sutskever et al., "Sequence to sequence learning with neural networks," Advances in neural information processing systems, 2014.
[68] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, no. 2, pp. 179-211, Apr. 1990.
[69] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
[70] K. Wołk et al., "Survey on dialogue systems including slavic languages," Neurocomputing, vol. 477, pp. 62-84, Feb. 2022.
[71] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.
[72] A. Vaswani et al., "Attention is all you need," Advances in neural information processing systems, pp. 30, 2017.
[73] J. Zhang et al., "Pegasus, Pre-training with extracted gap-sentences for abstractive summarization," Book Pegasus, Pre-training with extracted gap-sentences for abstractive summarization (PMLR), pp. 11328-11339, 2020.
[74] M. Lewis et al., "Bart, Denoising sequence-to sequence pre-training for natural language generation, translation, and comprehension," arXiv, 2019.
[75] A. Radford et al., "Language models are unsupervised multi-task learners," OpenAI blog, vol. 1, no. 8, pp. 9, 2019.
[76] Y. Liu and M. Lapata, "Hierarchical transformers for multi-document summarization," arXiv, 2019.
[77] P. J. Liu et al., "Generating wikipedia by summarizing long sequences," arXiv, 2018.
[78] N. Kitaev et al., "Reformer, The efficient transformer," arXiv, 2020.
[79] J. Zhang et al., "Pegasus, Pre-training with extracted gap-sentences for abstractive summarization," PMLR, pp. 11328-11339, 2020.
[80] H. Lin and V. Ng, "Abstractive summarization, A survey of the state of the art," pp. 9815-9822, Dec. 2019.
[81] A. Nenkova and K. McKeown, "Automatic summarization," Foundations and Trends® in Information Retrieval, vol. 5, no. 2–3, pp. 103-233, Nov. 2011.
[82] B. Tozyılmaz, "Abstractive text summarization on WikiHow dataset using sentence embeddings," Middle East Technical University, 2019.
[83] I. Sutskever et al., "Sequence to sequence learning with neural networks," Advances in neural information processing systems, pp. 27, 2014.
[84] E. Lloret et al., "The challenging task of summary evaluation, an overview," Language Resources and Evaluation, vol. 52, no. 1, pp. 101-148, Mar. 2018.
[85] C.-Y. Lin, "Rouge, A package for automatic evaluation of summaries," pp. 74-81, 2004.
[86] J. McAuley and J. Leskovec, "From amateurs to connoisseurs, modeling the evolution of user expertise through online reviews," WWW, 2013.
[87] A. Jangra et al., "A Survey on Multi-modal Summarization," J. ACM, vol. 37, no. 4, Article 111, 37 pages, Jul. 2018.