

Advancements in abstractive text summarization: a deep learning approach

Wesam Suliman¹, Amer Yaseen², Nuha Hamada³

¹Student Affairs Office, Al Ain University, Abu Dhabi, United Arab Emirates

²Department of Information Technology, Information Technology Software Development Manager, Trust Technical Services L.L.C, Abu Dhabi, United Arab Emirates

³College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates

Article Info

Article history:

Received Jun 13, 2024

Revised Feb 4, 2025

Accepted Mar 15, 2025

Keywords:

Abstractive

Deep learning

Long short-term memory

Natural language processing

Text summarization

ABSTRACT

With the rapid growth of data, text summarization has become vital for extracting key information efficiently. While extractive text summarization models are widely available, they often produce redundant outputs with limited capability of generating human-like summaries. Abstractive summarization (AS), which generates new phrases and rephrases content, remains underexplored due to its complexity. This paper addresses this gap by developing an abstractive deep learning (DL) model using an encoder-decoder architecture supported with an attention mechanism. Trained on the dataset of Amazon food reviews, the model generates contextually rich and semantically accurate summaries. The model's evaluation using bilingual evaluation understudy (BLEU) and recall-oriented understudy for gisting evaluation (ROUGE) metrics demonstrated promising results, with a score of 0.641 for BLEU, 0.520 for ROUGE-1, 0.345 for ROUGE-2, 0.461 for ROUGE-L, and 0.428 for ROUGE-W, indicating coherence and structural integrity. This research highlights the potential of DL in addressing the limitations of classical methods and suggests opportunities for future advancements, such as scaling the model with larger datasets and integrating transformer-based techniques for improved summarization across diverse applications.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Nuha Hamada

College of Engineering, Al Ain University

Abu Dhabi, UAE

Email: nuha.hamada@aau.ac.ae

1. INTRODUCTION

The exponential growth of textual contents and continuous advancement of technology have made text summarization an essential requirement to extract valuable information from the huge volumes of text [1]. News, publications, legal analysis, medical reporting, daily logs, and reviews are just a few examples of domains in which text summarization can save a tremendous amount of time and resources [2]. In these domains, text summarization is crucial to produce a shorter version of the available huge text blocks while preserving the fundamental meaning of that text. However, this task involves several challenges that need to be resolved [3].

Effective text summarization requires a thorough understanding of the context, semantics and the relationships within the text. Traditional rule-based and statistical methods often fail to capture nuanced meanings. These issues result in summaries that lack coherence and completeness [4]. In addition, texts range from well-organized scientific articles to informal social media posts. Such diversity of textual structures

further complicates the development of robust models [5]. Extractive summarization models directly return sentences from the source text. They often produce summaries that are redundant or irrelevant. On the other hand, abstractive models generate new sentences, but they struggle with grammatical correctness and fluency [6], [7]. Balancing the essence of the text with concise and coherent outputs remains a significant challenge.

Recent advancements in natural language processing (NLP) have shown promising results in overcoming the limitations of both abstractive and extractive text summarization. Deep learning (DL) techniques such as recurrent neural networks (RNN) have proven effective in capturing complex contexts. Long short-term memory (LSTM) networks offer additional improvements by handling sequential data [8], [9]. Transformer models have revolutionized the field by handling long-range dependencies. These models use attention mechanisms to better understand textual content [5], [9], [10]. In addition, pre-trained models such as bidirectional encoder representations from transformers (BERT), generative pre-trained transformer (GPT), and text-to-text exchange transformer (T5) have raised the bar higher. They leverage vast amounts of pre-training data to produce high-quality summaries. They achieve this through fine-tuning for specific tasks like abstractive and extractive summarization, topic modeling, and question answering [7], [11], [12]. Unlike extractive methods, DL models excel in abstractive summarization (AS) by constructing a shorter version of text that clearly describes the overall meaning of the original text, leading to more concise, human-like summaries [13].

Despite the many advancements, the field of abstractive text summarization remains underexplored compared to extractive summarization. AS requires significant improvements to match the reliability and fluency of extractive methods [4], [7], [9], [14]. This study addresses this gap by developing an abstractive text summarization model using DL techniques, implemented with Python and Keras application programming interfaces (APIs). The model employs a sequence-to-sequence framework with an attention mechanism, trained on domain-specific data. By addressing the challenges of AS, this research contributes to advancing the field and improving the accessibility of critical information.

The remainder of this paper is structured as follows. Section 2 reviews related work. Section 3 discusses text summarization in the context of NLP and section 4 presents the proposed model and experimental setup. Finally, the results are analyzed, and the study is concluded.

2. RELATED WORK

Text summarization, which condenses large textual content into a concise form while retaining its key information, has been a significant area of research for decades [4], [11]. Early approaches to summarization relied on simple techniques such as word frequency [15], sentence position [16], and keyword-based methods [17]. These methods, while effective in certain cases, often failed to capture deeper semantic relationships within the text. McCulloch and Pitts proposed the original neural network concept in 1948 [18], [19] explained backpropagation and the use of gradient descent to minimize loss. However, different obstacles and challenges were identified, including long training times, model overfitting, and a limited number of hidden layers. Hinton and Salakhutdinov [20] proposed DL and RNN, which were then applied to different NLP domains, including object and voice recognition. Lecun *et al.* [21] proposed a RNN model capable of predicting sequential data and word recognition. As research progressed, more advanced statistical techniques like naïve Bayes and hidden Markov models were introduced, followed by the emergence of multi-document summarization methods in the 1990s, including topic and indicator representation approaches [22].

Modern text summarization methods are broadly categorized into two types: extractive and AS. Extractive methods spot and reuse main terms and phrases directly from the original text, while abstractive methods produce new sentences that paraphrase the source content, capturing its essence in a more human-like manner [23]. Extractive summarization has reached a mature level, while abstractive methods still rely on complex language generation capabilities and remain challenging [24], [25].

DL techniques have enriched the text summarization domain by enabling models to understand and produce text with greater fluency and coherence. The development of RNNs and LSTMs marked a significant milestone by addressing issues like vanishing gradients and enabling the modeling of sequential data [21]. The introduction of the sequence-to-sequence (Seq2Seq) architecture further improved summarization by allowing models to convert input sequences into output sequences effectively [10], [26], [27]. The use of attention mechanisms within these frameworks enabled models to shed light on the most important elements of the original text, enhancing the informativeness and relevance of generated summaries [9], [28].

The rise of transformer models has advanced content summarization capabilities. BERT presented bidirectional modeling with improved representation of sentences [11], [12]. Additionally, GPT has demonstrated exceptional capabilities in AS with its capability of producing human-like summaries [29]. T5 has also shown versatility and breakthrough performance in summarization by leveraging pre-training on

different tasks [12]. In spite of the improvements achieved by these models, they still have computational complexity and require high resources which restricts their availability for viable applications.

This study builds upon the transformative capabilities of models such as BERT, GPT, and T5 while addressing their limitations through the usage of a Seq2Seq and attention mechanisms. This approach aims to improve abstractive text summarization performance, especially in domain-specific contexts. As opposed to BERT and GPT, our model is fine-tuned particularly for summarization tasks which facilitates the development of more specialized, productive, and context-aware summarization strategies.

Building on key insights from prior research, including the detailed review presented in [4], this research leverages DL advancements, such as attention mechanisms, to improve the quality and relevance of generated summaries. Moreover, this study highlights the importance of using domain-specific datasets rather than generic ones, which lack the nuance required to address the requirements of specialized fields effectively. With respect to model evaluation, our approach integrates both bilingual evaluation understudy (BLEU) and recall-oriented understudy for gisting evaluation (ROUGE) metrics to ensure a thorough and balanced model evaluation. This comprehensive evaluation approach overcomes the limitations of single-metric assessments and provides a robust framework for measuring the model's quality. By bridging the gap between existing methodologies and the practical requirements of real-world tasks, this study aligns with current best practices in the summarization field and contributes to the ongoing development of summarization technologies.

3. TEXT SUMMARIZATION IN NATURAL LANGUAGE PROCESSING

Text summarization is the process of generating a precise summary from a longer text while preserving the original content and overall meaning [3], [30]. There are mainly two methods for text summarization which are namely extractive summarization and AS. In the following subsections, the main methods of text summarization are explained, and then, other key concepts used in building our model are briefly introduced.

3.1. Extractive summarization

In this method, the most significant sentences or phrases are spotted and extracted from the text. These extracted sentences are then combined to represent the final output text summary, as depicted in Figure 1. Recent advancements have leveraged DL techniques, particularly transformer models, to enhance the performance of extractive summarization. Liu and Lapata [31] introduced BERT-based summarization model (SUM), an extension of BERT for extractive summarization tasks. BERTSUM utilizes BERT's powerful contextual embeddings to better understand sentence-level representations and employs a classifier to identify salient sentences for summarization. This approach significantly outperformed previous extractive methods on benchmark datasets. A graph-based neural model for extractive summarization, where sentences are represented as nodes and their relationships as edges in a graph was proposed in [32]. This method effectively captures the structural information of the text and enhances the coherence of the generated summaries. Narayan *et al.* [33] presented a neural extractive summarization model that uses a Seq2Seq framework with reinforcement learning. The model is trained to select sentences that maximize the ROUGE score. Despite these advancements, existing summarization methods still face several limitations. Extractive methods may still result in summaries that are not coherent or sentences may be selected directly from the source text without any change [34]. In addition, these models may include redundant information, as they focus on individual sentences rather than the overall content [35].

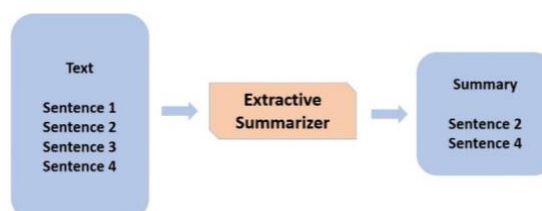


Figure 1. Concept of extractive summarizer

3.2. Abstractive summarization

AS not only generates a shorter version of the input text, but it also may produce new sentences that do not exist in the original text [24]. This concept is depicted in Figure 2 where multiple sentences are fed

into a text summarizer as input while a shorter version of the text along with some new contents are generated as output. The process of text summarization involves more challenging techniques like text rephrasing, paraphrasing, and generation, yet produces more human like summaries [1], [25], [36], implemented pointer-generator networks that utilize both abstractive and extractive methods to ensure the important parts of the text is included in the output summary. Vaswani *et al.* [9] introduced the transformer model that implements self-attention mechanisms for handling the text long-range dependencies in text.

Transformers have become the foundation for many state-of-the-art summarization models, such as bidirectional and auto-regressive transformers (BART), pre-training with extracted gap-sentences for abstractive summarization (PEGASUS), and T5. Lewis *et al.* [37] developed BART, a denoising autoencoder for pretraining sequence-to-sequence models. BART combines the strengths of BERT's bidirectional encoding and GPT's autoregressive decoding, making it highly effective for both extractive and AS tasks. Zhang *et al.* [38] introduced PEGASUS, which pretrains a transformer model by masking whole sentences rather than individual tokens. This approach improves the model's ability to understand sentence-level semantics and leads to better performance in AS tasks. The T5 model was introduced by Raffel *et al.* [12]. It frames all NLP tasks as text-to-text problems. The model implements a unified framework with extensive pretraining and fine-tuning, which enables it to perform well in text generation tasks such as text summarization. Despite these advancements, abstractive models still face significant challenges. For example, these models tend to generate grammatically incorrect or nonsensical sentences when dealing with complex source texts [39]. Another issue is the potential for information loss, which can result in omitting important details [40].

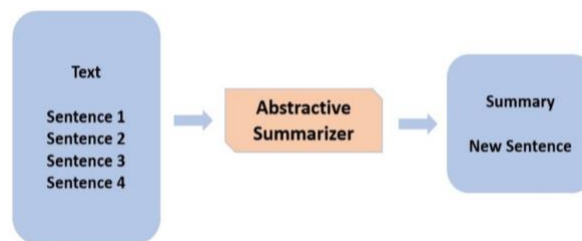


Figure 2. Abstractive summarization

4. METHOD

Human text summarization is a lengthy and time-consuming process. The outcome is subjective and varies widely in quality and accuracy. Therefore, NLP has been applied in several ways to automate the text summarization process [41]. In this paper, a DL text summarization model is developed using Python and Keras AI APIs. Model training was performed using the Amazon food reviews dataset [42]. Textual reviews in this dataset are used to predict a short version of the text without losing key contextual elements or meaning. For training and validation purposes, the dataset is split into 90% for training and 10% for validation. The following sections explain the different techniques and concepts used in our model in more detail.

4.1. Sequence-to-sequence modeling

Seq2Seq is a well-known modeling technique used for processing sequential information. This technique is usually utilized in sentiment analysis [43], named entity recognition (NER) [44] and machine translation problems [45]. In machine translation, the model generates output text in a certain language given that the input text is in another language [46]. In NER, a sequence of words is fed into the model, which in turn generates a sequence of tags for each of the inputted words [46]. In our research, the Seq2Seq model receives a paragraph consisting of a sequence of words and generates a summarized version of the text as a many-to-many sequence problem. The model primarily consists of an encoder and decoder, which are further explained in the following sections.

4.2. Encoding-decoding model

In text summarization models, the input is usually a long sequence of words, while the output is typically a shorter form of the input text. Encoder and decoder components are typically based on either a LSTM or gated RNN due to their capability of handling long term dependencies and solving the problem of vanishing gradient [47]. An encoder LSTM model processes the entire input sequence by feeding one word

into the encoder at each timestep. It captures and processes the contextual information at each timestep throughout the input sequence.

Setting up the model involves two main phases: training and inference. In the training phase, one word is passed at a time to the encoder, which processes the input at each iteration and captures the fundamental information found in the input text. The decoder is another LSTM network that analyzes the target sequence in full and predicts the output sequence. Model training takes place so the model can observe the previous work and be able to predict the next word. In the inference phase, new unseen sentences are fed to the model for testing. The process of decoding starts by feeding the output of the encoder to the model. At each timestep, the decoder generates the next word with the highest probability. This process repeats for all words until the end of the input sequence.

4.3. Attention mechanism

In the encoding-decoding model, the encoder encodes the input sequence completely into a vector of a fixed length. Then, the decoder takes over to predict the output sequence. However, when the length of the input sequence is long, the encoder is required to memorize the whole fixed-length vector, which can cause a potential performance issue [48], [49]. To address this limitation, our model incorporates the attention mechanism, which focuses only on certain parts of the input sequence to predict each word in the output, rather than relying on the entire sequence. This allows the model to dynamically allocate attention to the most relevant portions of the input, improving the accuracy and coherence of the generated summaries. The key concept here is how much weight is given to each word when predicting the next output word.

There are mainly two types of attention mechanisms: global and local attention. Global attention analyzes all encoder hidden states, allowing for a comprehensive view of the entire input sequence, whereas local attention concentrates on a selection of encoder states, making the procedure more computationally efficient [50]. For our model, we chose global attention because it is better suited for AS tasks that require a thorough grasp of the full input context. By observing all hidden states, the model may capture nuanced linkages and dependencies in the text, resulting in summaries that are more coherent, content-rich, and contextually correct. This decision guarantees that our model excels at managing varied and complicated text structures.

4.4. Model architecture

Our model is based on the Seq2Seq transformer framework which has performed well in the field of text summarization and generation [51]. The model architecture consists mainly of three layers which are the encoder, decoder, and the attention mechanism, as depicted in Figure 3. The encoder layer handles the input text and converts it into a series of hidden state rich feature representations. These hidden states encapsulate the semantic and syntactic meanings of the input text. This allows the model to understand complex textual dependencies.

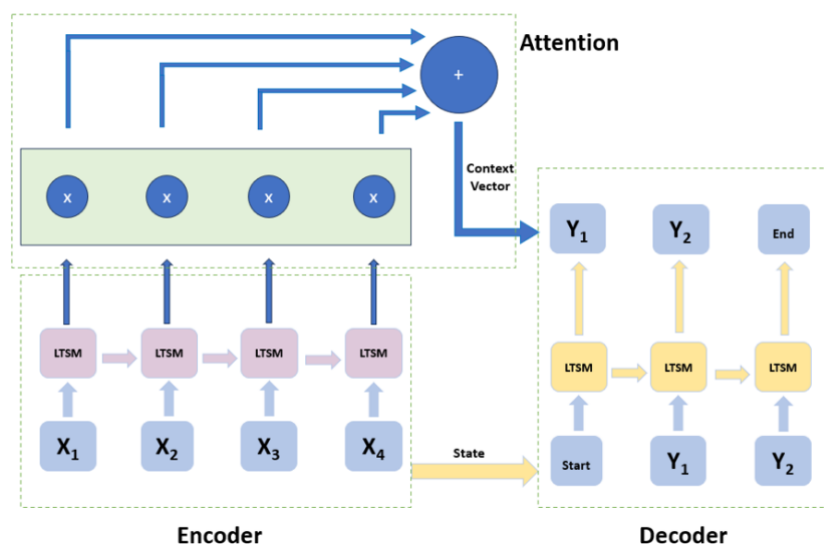


Figure 3. Text summarization model architecture

The context vector, which aggregates the encoder's output, is passed to the decoder which in role generates the summarized text. The decoder is also implemented as an iterative neural network. At each time step, the combination of the context vector, attention mechanism, and the previously generated words are

used to predict the next word in the output sequence. The attention mechanism plays a critical role by dynamically focusing on the most relevant parts of the input sequence for each decoding step which improves both coherence and relevance in the output.

Our architecture differs from traditional sequence-to-sequence models in that it features performance optimization. This is achieved by including stacked layers in both the encoder and decoder, allowing the model to capture more complex dependencies within the input data. In addition, the attention mechanism facilitates better alignment of positional and semantic information in the output summary.

Model training on the Amazon food reviews dataset took place using the approach of supervised learning. Loss function was continuously optimized in both the encoder and decoder to ensure that output text is closely matching the reference summaries. For evaluation, BLEU score was calculated with a value of 0.641 indicating that the model achieved a high degree of alignment with human-generated summaries. This score reflects the model's ability to effectively capture lexical and semantic elements of the input text. Furthermore, the model's performance can be enhanced, making it a scalable and reliable framework for abstractive text summarization tasks by using larger datasets and higher computational resources.

4.5. Environment setup and data preprocessing

In this project, PyCharm IDE was used to develop the Python implementation code. A project virtual environment folder was set up, and all prerequisite libraries were downloaded, including Keras as the main library. The review of the fine food dataset was copied to the working directory as well. The dataset consists of around 600,000 reviews, which include different structures of data such as ratings, user information, and user text reviews. For the purpose of model training, 10% of the data was used to avoid long training times or overloading the computer used to develop and train the model.

Firstly, the dataset was initialized and loaded. Then the data structure was reviewed. The raw dataset was found to be unstructured and contained a lot of unnecessary data. Therefore, basic data processing operations were performed to prepare the data and make it ready for training the model. Data processing started with removing duplicate and empty rows. Then, other processing tasks were applied to handle unnecessary symbols and characters, including, i) converting all text to lowercase, ii) removal of any HTML tags; iii) applying contraction mapping, such as converting “can’t” to “cannot”, iv) removing [’s], v) removing text between parentheses (), vi) removing special characters and punctuation marks, and vii) removal of stop words.

Prior to developing the model, some model parameters needed to be derived from the dataset, which would affect both the training and prediction phases. The maximum length of the summary and the maximum length of the text review are examples of such parameters. To derive suitable values for these two parameters, a function was defined to plot the dataset sequence distribution of summary and review parts. The plot shown in Figure 4 is a pair of histograms that visually depict the distribution of word counts in two sets of text data: the original reviews (or full text) and their corresponding summaries. Each histogram shows the frequency of word counts for these texts, helping to understand their length distribution across the dataset. The x-axis, labeled “word count” represents the number of words per review or summary, while the y-axis, labeled “Frequency” represents how many entries (reviews or summaries) match each word count range.

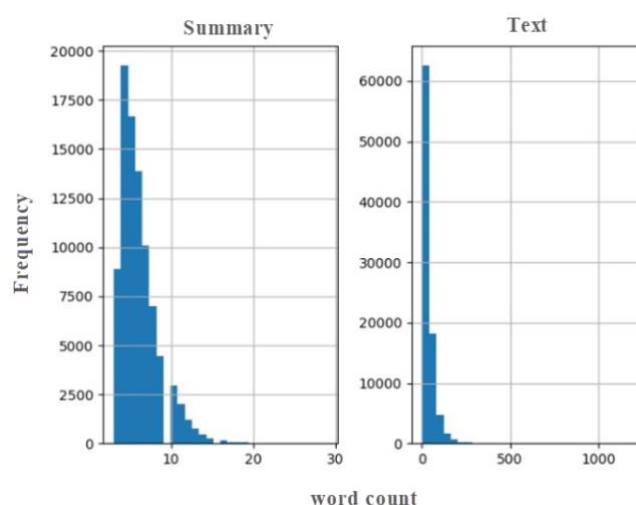


Figure 4. Text sequence distribution graphs

From the histograms, we can observe common patterns, such as how much longer the original reviews might be compared to the summaries. A wider spread along the x-axis for reviews would suggest that reviews generally contain more words and vary significantly in length, while a narrower spread for summaries would imply, they are consistently shorter and potentially more standardized. Accordingly, the maximum length of reviews was set to 80 words, as most reviews in the dataset are around this length. Similarly, the maximum length of summaries was set to 10 words which is the typical length of summaries in the data. These settings allow the model to focus on the most relevant information with standardized input sizes leading to an improved model accuracy.

4.6. Building the model

The text summarization model developed in this research leverages a multi-layer architecture to effectively process and generate output summaries. The model starts with two input layers: one for the input text sequence and another for the target sequence used during training. These inputs are passed through separate embedding layers which in role, transform words into dense vector representations. This step ensures that the semantic relationships between words are captured and hence, providing a strong basis for further processing.

There are three stacked LSTM layers at the core of the model. These layers play a vital role in capturing long-term dependencies and contextual relationships within the input text. Ma *et al.* [52] using multiple LSTM layers enhances the model's ability to represent text sequences by learning progressively complex features at each layer. This hierarchical arrangement allows the model to better capture nuanced patterns in the data resulting in more accurate and comprehensive text representations. Following the LSTM layers, an attention layer is incorporated which dynamically focuses on the most relevant parts of the input sequence during decoding. This mechanism generates a weighted context vector that prioritizes the most critical information in the input text which is significantly improving the relevance of the generated summaries.

Finally, the outputs from the attention mechanism are concatenated with the decoder LSTM's hidden states and passed through a dense layer wrapped in a time-distributed layer. This final component predicts the next word in the sequence at each time step. By combining embedding, multi-layer LSTMs, attention, and dense layers, the model achieves an architecture capable of generating meaningful, fluent, and content-rich summaries, while ensuring better representation of text sequences through the multi-layer LSTM structure as described in [52].

4.7. Model training

During model training, the validation loss is continuously monitored, and the training process will stop if an increase in validation loss is detected. The batch size for model training was set to 512, reflecting the number of samples for each gradient update. The model validation process was applied to the remaining 10% of the holdout set. The following code snippet shows the model training statement. The code snippet represents the training process for the text summarization model using the fit function in Keras. The model is trained in a Seq2Seq format, where the input consists of the original text (input_sequences) and a shifted version of the target text (target_text[:, :-1]) as the decoder input. The target output (target_text.reshape(target_text.shape[0], target_text.shape[1], 1)[:, 1:]) is reshaped to align with the model's expected output format, using the shifted target text (excluding the first element). The training is performed over a predefined number of epochs (ep_count) with a batch size of 512 to ensure efficient gradient updates. Early stopping (early_stopping) is applied as a callback to halt training if the validation loss stops improving, thus preventing overfitting. Validation data, consisting of val_in and val_out, undergoes the same preprocessing steps as the training data to ensure consistency. This setup ensures the model learns to predict the next word in the sequence effectively, leveraging both input text and previously generated words during training.

```
set=model.fit(
    [input_sequences, target_text[:, :-1]],
    target_text.reshape(target_text.shape[0], target_text.shape[1], 1)[:, 1:],
    epochs=ep_count,
    callbacks=[early_stopping],
    batch_size=512,
    validation_data=(
        [val_in, val_out[:, :-1]],
        val_out.reshape(val_out.shape[0],
            val_out.shape[1], 1)[:, 1:]
        )
    )
```

During model training, the validation loss is monitored to determine when to stop the process. Training halts automatically if the validation loss begins to increase, indicating potential overfitting. In Figure 5, which plots both training and validation loss over 10 epochs, we see that the model's error decreases on both datasets initially, demonstrating effective learning.

At the beginning, both training (solid line) and validation loss (dotted line) decrease, indicating improvement on both the training and validation datasets. However, starting from epoch 4, the validation loss continues to slightly decrease, and after epoch 8, it increases, while the training loss continues to decrease. This divergence suggests that the model is starting to overfit to the training data, meaning it is learning details that are specific to the training set but not generalizable to new data.

By implementing early stopping, we prevent the model from continuing to train past the point of lowest validation loss (around epoch 8). Stopping at this stage allows the model to maintain better generalization, balancing accuracy on the training set with performance on unseen data. This technique ensures the model does not overfit, preserving its effectiveness in real-world applications.

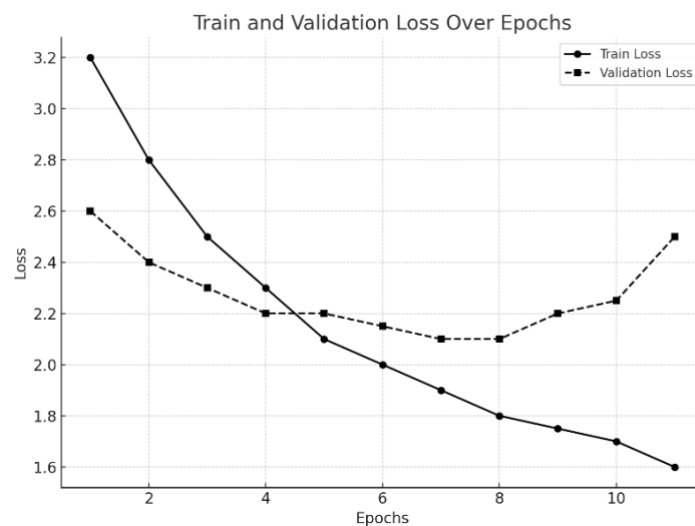


Figure 5. Training and validation loss over epochs

4.8. Model evaluation

4.8.1. BLEU score

The BLEU score was calculated in this experiment to evaluate the model's performance. BLEU is a very commonly used metric in evaluating machine translation models which can also be used for text summarization as well [4], [53]. A BLEU score of 1.0 represents a perfect match, while a score of 0.0 indicates a perfect mismatch. Typically, a score of 1.0 is not possible as the generated summary can differ from the reference text while it is still valid and precise.

In this experiment, the model achieved a BLEU score of 0.641. Such a score indicates a strong performance and a high degree of similarity to the reference summaries. It also demonstrates that the model is effective at capturing and replicating essential content from the input sequences. In addition, the result reflects the ability of the model to identify key phrases and semantic relationships within the input text and accurately generate summaries that preserve the core meaning.

On the other side, the BLEU score of 0.641 also suggests a need for improvement in handling longer or more complex text structures. Further fine-tuning of the model and using larger training dataset could potentially improve this score and lead to more accurate summarization results. This score provides a solid foundation for further research and optimization of AS models.

4.8.2. ROUGE score

ROUGE is performance evaluation metric which is specifically designed for text summarization tasks. It provides an in-depth analysis of the overlap between the generated summaries and reference summaries [54], [55]. ROUGE includes different sub-metrics that evaluate different dimensions of summary quality by comparing the generated summaries with reference summaries.

ROUGE-1 captures model's ability to include the most critical words from the input text by measuring the unigram (single-word) overlap. ROUGE-2 reflects the model's capability to understand the short-term contextual relationships within the text by measuring the bigram (two-word sequence) overlap. ROUGE-L emphasizes sentence-level structural coherence by focusing on the longest common subsequence (LCS) between the generated and reference summaries. Additionally, ROUGE-W rewards summaries that preserve longer meaningful spans of text by prioritizing the longest contiguous subsequence matches. In this

experiment, the model achieved ROUGE scores of 0.520 for ROUGE-1, 0.345 for ROUGE-2, 0.461 for ROUGE-L, and 0.428 for ROUGE-W. These scores demonstrate the model's effectiveness in generating relevant and structurally coherent summaries.

ROUGE-1 score of 0.520 indicates that the model effectively captures essential content and lexical overlap with the reference summaries. The ROUGE-2 score of 0.345 reflects the model's ability to capture contextual relationships through bigram overlaps, which is a critical factor for ensuring meaningful summaries. The ROUGE-L score of 0.461 demonstrates the model's strength in maintaining sentence-level structure and producing coherent summaries. Finally, the ROUGE-W score of 0.428 highlights the model's capacity to retain longer and contiguous segments of information.

These scores collectively demonstrate the model's robust performance in producing accurate, coherent, and contextually rich summaries. However, there is still a room for improvement in enhancing the bigram and structural coherence as suggested by the ROUGE-2 and ROUGE-W scores. Future efforts could focus on refining the attention mechanism or employing reinforcement learning techniques to optimize the summarization process further.

4.8.3. Comparative analysis

Evaluating the model's performance using both BLEU and ROUGE provides a well-rounded framework that captures different aspects of summary quality. A BLEU score of 0.641 indicates the model's strong alignment with reference summaries, while ROUGE offers additional insight by measuring lexical and structural coherence. Specifically, the ROUGE scores (ROUGE-1: 0.520, ROUGE-2: 0.345, ROUGE-L: 0.461, and ROUGE-W: 0.428) reflect the model's effectiveness in preserving key information, capturing short-term relationships, and maintaining logical sentence flow. This dual-metric approach affirms the model's ability to generate coherent and accurate summaries, aligning with established benchmarks in NLP research. At the same time, the results shed light on potential areas for improvement, such as better handling of complex sequences and long-term contextual relationships. Overall, this evaluation highlights the model's strengths while paving the way for further refinement and practical applications.

4.9. Model results

Figure 6 presents three random samples of reviews from the Amazon food reviews dataset, along with their original summaries and the predicted summaries generated by our text summarization model. The results demonstrate that the model performs effectively in capturing the essence of the reviews while maintaining brevity and coherence. For instance, in the first review, the model successfully identifies the key sentiment of a "rich coffee flavor" while addressing the packaging issue, closely aligning with the original summary. Similarly, in the second and third examples, the predicted summaries reflect the main ideas such as the nutritional value of protein bars and the relaxing properties of herbal tea while excluding extraneous details. These results highlight the model's ability to generalize across diverse contexts, accurately identifying critical aspects of the original text. However, minor discrepancies, such as slightly less descriptive phrasing in the predicted summaries, suggest opportunities for further fine-tuning to enhance specificity and accuracy.

<p>Review: "I purchased this coffee a few weeks ago, and I must say, the flavor is fantastic! It has a rich, bold taste, and the aroma fills my kitchen every morning. However, I noticed that the packaging was not very eco-friendly, which is something the company could improve."</p> <p>Original Summary: "Rich coffee flavor with eco-packaging concerns."</p> <p>Predicted Summary: "Bold coffee taste but packaging needs improvement."</p>
<p>Review: "The protein bars are a convenient snack for my busy schedule. They have a great balance of sweetness and nutrition, and I love that they use natural ingredients. However, I wish the texture was less crumbly, as it can be a bit messy to eat on the go."</p> <p>Original Summary: "Tasty and healthy protein bars but crumbly texture."</p> <p>Predicted Summary: "Protein bars: nutritious but crumbly."</p>
<p>Review: "I recently bought this herbal tea, and it has become my favorite! The calming blend helps me relax after a long day, and the natural ingredients are a big plus. My only concern is the slightly high price compared to similar products."</p> <p>Original Summary: "Relaxing herbal tea with a slightly high price."</p> <p>Predicted Summary: "Favorite herbal tea but pricey."</p>

Figure 6. Text summarization model output

5. DISCUSSION AND RECOMMENDATION

After completing the model training and reviewing the results, the model evaluation scores are found to be promising as they indicate a strong performance and high capability of generating relevant summaries which are closely matching the meaning of the original text. This showcases the model's potential for real world applications such as text summarization in the customer reviews, academic research, and commercial domains. On the other side, there is a considerable room for improvement to optimize the model's training process and capabilities.

The most important aspect of model's improvement lies in increasing the size of the training data in order to improve the accuracy and generalizability of the model, but that comes with the cost of increased training time. In this experiment, the plan was to train the model using 90,000 records. However, the training time proved to be a limiting factor so we had to compromise the model settings and reduce the dataset size to 10,000 records. That led to reducing the training time to six hours which is reasonable compared with the time needed to train the full dataset. Despite that reducing the size of dataset yielded satisfactory results, it proves that using larger datasets would improve the model's performance and extend its generalization capability across diverse text patterns and nuances.

6. CONCLUSION AND FUTURE WORK

In this project, a neural network text summarization model was developed which basically has three stacked LSTM layers. The Amazon food reviews dataset was used to train the model. Model training was stopped at epoch 8 at which the validation loss started to increase. The model's performance was evaluated using BLEU and ROUGE scores with a BLEU score of 0.641 and ROUGE-1, ROUGE-2, and ROUGE-L scores of 0.520, 0.345, and 0.461, respectively. These results indicate the model's ability to replicate critical elements of the input text while maintaining structural integrity. Despite its promising results, the model has some limitations. Training the model on a smaller dataset due to computational constraints restricted its generalizability across diverse text structures. This limitation can be addressed by increasing the dataset size but would require higher hardware resources and longer training time. Additionally, the model occasionally produces inconsistent or redundant summaries, particularly with longer texts, highlighting areas for improvement in conciseness and fluency. Future work could focus on leveraging reinforcement learning techniques or advanced attention mechanisms to refine and optimize the summarization process. Exploring transformer-based architectures like T5 or GPT variants may also enhance the model's capability to generate more nuanced and high-quality summaries. These advancements could further bridge the gap between current AS methods and human-level performance.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Wesam Suliman	✓			✓	✓	✓	✓	✓	✓	✓				✓
Amer Yaseen	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Nuha Hamada		✓		✓	✓		✓			✓	✓	✓	✓	

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are openly available online at <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>, reference number [42].




REFERENCES

- [1] A. P. Widyassari *et al.*, "Review of automatic text summarization techniques & methods," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1029–1046, 2022, doi: 10.1016/j.jksuci.2020.05.006.
- [2] J. K. Adeniyi, S. A. Ajagbe, A. E. Adeniyi, H. O. Aworinde, P. B. Falola, and M. O. Adigun, "EASESUM: an online abstractive and extractive text summarizer using deep learning technique," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1886–1897, 2024, doi: 10.11591/ijai.v13.i2.pp1888-1899.
- [3] M. Y. Day and C. Y. Chen, "Artificial intelligence for automatic text summarization," *Proceedings - 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018*, pp. 478–484, 2018, doi: 10.1109/IRI.2018.00076.
- [4] S. Sharma, G. Aggarwal, and B. Kumar, "A survey on the dataset, techniques, and evaluation metric used for abstractive text summarization," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 3, pp. 681–689, 2024, doi: 10.12928/TELKOMNIKA.v22i3.25512.
- [5] R. Mihalcea and P. Tarau, "TextRank: bringing order into texts," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004*, 2004, pp. 404–411.
- [6] M. D. Okpor, "Machine translation approaches: issues and challenges," *IJCSI International Journal of Computer Science Issues*, vol. 11, no. 5, pp. 159–166, 2014.
- [7] H. Lin and J. Bilmes, "A class of submodular functions for document summarization," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA. Association for Computational Linguistics, 2011, pp. 510–520.
- [8] G. Erkan and D. R. Radev, "LexRank: graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004, doi: 10.1613/jair.1523.
- [9] A. Vaswani *et al.*, "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, United States, 2017, pp. 5999–6009.
- [10] S. Wu, D. Zhang, N. Yang, M. Li, and M. Zhou, "Sequence-to-dependency neural machine translation," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 698–707, 2017, doi: 10.18653/v1/P17-1065.
- [11] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, vol. 1, pp. 4171–4186, 2019.
- [12] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.
- [13] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," *SIGIR 1998 - Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336, 1998, doi: 10.1145/290941.291025.
- [14] A. Alomari, N. Idris, A. Q. M. Sabri, and I. Alsmadi, "Deep reinforcement and transfer learning for abstractive text summarization: A review," *Computer Speech and Language*, vol. 71, 2022, doi: 10.1016/j.csl.2021.101276.
- [15] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159–165, 2010, doi: 10.1147/rd.22.0159.
- [16] P. B. Baxendale, "Machine-made index for technical literature-an experiment," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 354–361, 2010, doi: 10.1147/rd.24.0354.
- [17] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969, doi: 10.1145/321510.321519.
- [18] 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, 1943, doi: 10.1007/BF02478259.
- [19] J. Li, J. H. Cheng, J. Y. Shi, and F. Huang, "Brief introduction of back propagation (BP) neural network algorithm and its improvement," *Advances in Intelligent and Soft Computing*, pp. 553–558, 2012, doi: 10.1007/978-3-642-30223-7_87.
- [20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.
- [21] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [22] C. C. Aggarwal and C. X. Zhai, *Mining text data*, Springer New York, 2013, doi: 10.1007/978-1-4614-3223-4.
- [23] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review*, vol. 47, no. 1, pp. 1–66, 2017, doi: 10.1007/s10462-016-9475-9.
- [24] A. K. M. Masum, S. Abujar, A. M. I. Talukder, A. K. M. S. A. Rabby, and S. A. Hossain, "Abstractive method of text summarization with sequence to sequence RNNs," in *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, 2019, pp. 1–5, doi: 10.1109/ICCCNT45670.2019.8944620.
- [25] J. Li, C. Zhang, X. Chen, Y. Cao, P. Liao, and P. Zhang, "Abstractive text summarization with multi-head attention," *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8851885.
- [26] X. Yin and X. Wan, "How do Seq2Seq models perform on end-to-end data-to-text generation?," in *P Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland. Association for Computational Linguistics, 2022, vol. 1, pp. 7701–7710, doi: 10.18653/v1/2022.acl-long.531.
- [27] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104–3112, 2014.
- [28] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *The Thirty-Second AAAI Conference on Artificial Intelligence 2018 (AAAI-18)*, pp. 2604–2611, doi: 10.1609/aaai.v32i1.11873.




- [29] X. V. Lin *et al.*, “Few-shot learning with multilingual generative language models,” in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022*, Dec. 2022, pp. 9019–9052, doi: 10.18653/v1/2022.emnlp-main.616.
- [30] D. R. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” *Computational Linguistics*, vol. 28, no. 4, pp. 399–408, 2002, doi: 10.1162/089120102762671927.
- [31] Y. Liu and M. Lapata, “Text summarization with pretrained encoders,” in *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, 2019, pp. 3730–3740, doi: 10.18653/v1/d19-1387.
- [32] M. Yasunaga, R. Zhang, K. Meelu, A. Pareek, K. Srinivasan, and D. Radev, “Graph-based neural multi-document summarization,” *CoNLL 2017 - 21st Conference on Computational Natural Language Learning, Proceedings*, pp. 452–462, 2017, doi: 10.18653/v1/k17-1045.
- [33] S. Narayan, S. B. Cohen, and M. Lapata, “Ranking sentences for extractive summarization with reinforcement learning,” *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 1747–1759, 2018, doi: 10.18653/v1/n18-1158.
- [34] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, “Neural abstractive text summarization with sequence-to-sequence models,” *ACM/IMS Transactions on Data Science*, vol. 2, no. 1, pp. 1–37, 2021, doi: 10.1145/3419106.
- [35] W. Guo, B. Wu, B. Wang, and Y. Yang, “Two-stage encoding extractive summarization,” *Proceedings-2020 IEEE 5th International Conference on Data Science in Cyberspace, DSC 2020*, pp. 346–350, 2020, doi: 10.1109/DSC50466.2020.00060.
- [36] A. See, P. J. Liu, and C. D. Manning, “Get to the point: summarization with pointer-generator networks,” *ACL 2017-55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 1073–1083, 2017, doi: 10.18653/v1/P17-1099.
- [37] M. Lewis *et al.*, “BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, 2020, doi: 10.18653/v1/2020.acl-main.703.
- [38] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, “PEGASUS: pre-training with extracted gap-sentences for abstractive summarization,” in *37th International Conference on Machine Learning, ICML 2020*, 2020, pp. 11265–11276.
- [39] M. Yang, C. Li, Y. Shen, Q. Wu, Z. Zhao, and X. Chen, “Hierarchical human-like deep neural networks for abstractive text summarization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2744–2757, 2021, doi: 10.1109/TNNLS.2020.3008037.
- [40] H. nan Wang *et al.*, “Deep reinforcement learning: a survey,” *Frontiers of Information Technology and Electronic Engineering*, vol. 21, no. 12, pp. 1726–1744, 2020, doi: 10.1631/FITEE.1900533.
- [41] N. Moratanch and S. Chitrakala, “A survey on extractive text summarization,” *International Conference on Computer, Communication, and Signal Processing: Special Focus on IoT, ICCCSPP 2017*, 2017, doi: 10.1109/ICCCSP.2017.7944061.
- [42] Stanford Network Analysis Project (SNAP), “Amazon fine food reviews,” *Kaggle*, 2017, [Online]. Available: <https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>.
- [43] R. Satapathy, Y. Li, S. Cavallari, and E. Cambria, “Seq2Seq deep learning models for microtext normalization,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2019, 2019, doi: 10.1109/IJCNN.2019.8851895.
- [44] X. Zhao, Y. Zhang, and X. Yuan, “Dependency-aware attention model for emotion analysis for online news,” in *Advances in Knowledge Discovery and Data Mining*, Cham: Springer International Publishing, Mar. 2019, pp. 172–184, doi: 10.1007/978-3-030-16148-4_14.
- [45] J. T. Z. Wei, K. Pham, B. Dillon, and B. O’Connor, “Evaluating syntactic properties of Seq2seq output with a broad coverage HPSG: a case study on machine translation,” in *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop*, 2018, pp. 298–305.
- [46] S. Zhao, E. Deng, M. Liao, W. Liu, and W. Mao, “Generating summary using sequence to sequence model,” in *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020*, 2020, pp. 1102–1106, doi: 10.1109/ITOEC49072.2020.9141919.
- [47] K. Cho *et al.*, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014, pp. 1724–1734, doi: 10.3115/v1/d14-1179.
- [48] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, “Convolutional sequence to sequence learning,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017, vol. 70, pp. 1243–1252, doi: 10.5555/3305381.3305510.
- [49] P. M. Hanunggul and S. Suyanto, “The impact of local attention in LSTM for abstractive text summarization,” in *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 2019, pp. 54–57, doi: 10.1109/ISRITI48646.2019.9034616.
- [50] J. Liu, G. Wang, P. Hu, L. Y. Duan, and A. C. Kot, “Global context-aware attention LSTM networks for 3D action recognition,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 3671–3680, doi: 10.1109/CVPR.2017.391.
- [51] Z. Liang, J. Du, and C. Li, “Abstractive social media text summarization using selective reinforced Seq2Seq attention model,” *Neurocomputing*, vol. 410, pp. 432–440, 2020, doi: 10.1016/j.neucom.2020.04.137.
- [52] S. Ma, Z. Xing, C. Chen, C. Chen, L. Qu, and G. Li, “Easy-to-deploy API extraction by multi-level feature embedding and transfer learning,” *IEEE Transactions on Software Engineering*, vol. 47, no. 10, pp. 2296–2311, 2021, doi: 10.1109/TSE.2019.2946830.
- [53] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, “BLEU: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, United States, 2002, pp. 311–318.
- [54] C. Y. Lin, “ROUGE: a package for automatic evaluation of summaries,” in *Text Summarization Branches Out, pages 74–81, Barcelona, Spain. Association for Computational Linguistics*, Barcelona, Spain, 2004, no. 1, pp. 74–81.
- [55] T. He *et al.*, “ROUGE-C: a fully automated evaluation method for multi-document summarization,” in *2008 IEEE International Conference on Granular Computing, GRC 2008*, 2008, pp. 269–274, doi: 10.1109/GRC.2008.4664680.

BIOGRAPHIES OF AUTHORS






Wesam Suliman    is a dedicated Senior Student Affairs Coordinator and researcher, specializing in teaching English to speakers of other languages (TESOL). She holds a master's degree in education with a focus on TESOL and is currently pursuing a Ph.D. in the same field. She has extensively researched various aspects of TESOL, particularly ESL learners' communication and negotiation skills. Her work also delves into the grammatical-based learning (GBL) approach, contributing valuable insights into effective language instruction methodologies. Wesam's expertise extends to the application of natural language processing (NLP) techniques, including translation and summarization, enhancing the learning experience for ESL students. She is enthusiastic about integrating these advanced techniques to improve communication and comprehension in diverse student populations. She can be contacted at email: Wesam.Suliman@aau.ac.ae.



Amer Yaseen    is an accomplished IT Software Development Manager specializing in the development of modern web applications empowered with AI capabilities. He holds a master's degree in informatics and data science, and his professional interests are deeply rooted in the research and advancement of data processing techniques within the machine learning and AI domains. He is passionate about integrating AI technologies with web and smart device applications to revolutionize user experiences. His work focuses on leveraging cutting-edge AI to develop innovative solutions that elevate functionality and interactivity, providing users with enhanced, seamless, and intelligent interactions. With a strong background in both web development and data science, he is capable of driving significant progress in the way users interact with digital platforms and shifting their experiences to new heights. He can be contacted at email: amyasien@gmail.com.



Nuha Hamada    is working at Al Ain University, UAE. She has completed her Ph.D. in Functional Analysis-Hilbert spaces from the University of Baghdad. Her Ph.D. thesis addressed the Jordan derivation on the algebra of all bounded linear operators on separable infinite dimensional complex Hilbert space. Her research interests include cyclic phenomena, optimization problems, and machine learning. In addition to this work, she has contributed to the area of quantitative analysis in management, chaos theory from a decision-making perspective, applying statistical techniques to investigate difficulties in learning and find some cultural factors that affect the process of teaching and learning and studying how LMS can support teaching and learning. She has filed many applications to the Patent Office in Egypt and WIPO. She can be contacted at email: nuha.hamada@aau.ac.ae.