

ASBO-GRU: Optimization-Enhanced Contextual and Aspect-Aware Neural Summarization

Neelu Verma

Indian Institute of Technology Jodhpur, India

Abstract—Automatic text summarization aims to generate concise and informative summaries while preserving the semantic content and factual consistency of long documents. Despite the remarkable performance of large Transformer-based models, their high computational complexity motivates the development of compact and efficient summarization architectures. In this paper, we propose ASBO-GRU, an optimization-enhanced abstractive summarization framework that integrates BERT-based contextual encoding, aspect-aware feature extraction, a GRU summarization module, and the Adaptive Sandpiper-Based Optimizer (ASBO). The proposed model exploits contextual token representations to identify salient semantic aspects and employs ASBO to optimize the GRU parameters for improved summary generation. Experimental evaluation on benchmark datasets demonstrates that ASBO-GRU consistently outperforms recurrent and competitive pretrained baselines, achieving 47.0% ROUGE-1, 24.4% ROUGE-2, 44.2% ROUGE-L, a 0.928 BERTScore, and 90.3% factual consistency. Ablation and computational analyses further confirm that the integration of contextual representations, aspect-aware learning, and optimization contributes to improved semantic quality while maintaining computational efficiency, making ASBO-GRU a practical alternative to larger pretrained summarization systems.

Index Terms—Text summarization; abstractive summarization; BERT; contextual representation; aspect-aware learning; GRU; Adaptive Sandpiper-Based Optimizer (ASBO); factual consistency.

I INTRODUCTION

The rapid growth of digital information has made it increasingly difficult for users to identify the most relevant information from large collections of textual documents. News articles, scientific publications, technical reports, web documents, and other textual resources may contain substantial amounts of information that cannot be processed efficiently by humans. Automatic text summarization addresses this

challenge by generating concise representations of one or more documents while preserving their most salient information. It has therefore become an important task in natural language processing (NLP), information retrieval, text mining, and knowledge-intensive applications.

Automatic summarization can broadly be categorized into extractive and abstractive approaches. Extractive summarization selects important sentences or textual units directly from the source document, whereas abstractive summarization generates new textual sequences that can paraphrase and reorganize source information. Although extractive systems can preserve source-level wording, their dependence on existing sentences can restrict conciseness and linguistic flexibility. Abstractive systems provide greater flexibility but introduce additional challenges, including information omission, redundancy, grammatical inconsistency, and factual errors.

Early neural summarization systems employed recurrent neural networks (RNNs), long short-term memory (LSTM) networks, gated recurrent units (GRUs), and encoder-decoder architectures. Sequence-to-sequence learning established a general framework for mapping an input sequence to an output sequence [22], while LSTM and GRU architectures improved the ability of recurrent models to represent sequential dependencies [6, 4]. Attention mechanisms subsequently enabled summarization models to focus selectively on informative portions of an input document.

The introduction of pretrained Transformer models substantially changed the summarization landscape. BERT demonstrated that bidirectional contextual pretraining can provide transferable representations for downstream NLP tasks [3, 24]. Subsequent sequence-to-sequence pretrained models, including BART, T5, and PEGASUS, achieved strong results on text generation and summarization benchmarks [13, 18, 28]. More recently, instruction-tuned large language models (LLMs) have demonstrated strong

capabilities for summarization and long-context language understanding. However, recent studies show that LLMs do not always utilize document context uniformly, and their performance can be affected by the position and distribution of relevant information within long inputs [19]. Nevertheless, large pretrained models can require substantial computational resources, and automatic summaries can still

contain unsupported statements, omissions, and redundancy. Factuality and faithfulness remain important concerns in abstractive summarization. Recent work has increasingly moved beyond lexical-overlap evaluation toward claim-level factuality assessment, evidence alignment, and NLI-based verification [16, 12, 5, 21, 9, 25].

These developments motivate investigation of architectures that balance contextual representation quality, sequential modeling capability, optimization efficiency, and computational cost. A lightweight recurrent summarizer can remain useful when a very large generative model is impractical or when model behavior needs to remain relatively compact and interpretable. However, the effectiveness of such a model depends strongly on the quality of its input representation and the optimization of its parameters.

Contextual representation is particularly important because salient information cannot always be identified from word frequency alone. BERT provides contextual token representations that incorporate surrounding linguistic information [3]. In addition, aspect-aware processing can provide an explicit mechanism for identifying semantically important terms or concepts. In this work, aspect information is therefore used as a salience-oriented feature rather than treating aspect extraction merely as a sentiment-analysis task.

Optimization constitutes an important component of the proposed framework, particularly for improving the parameter configuration of neural architectures. Optimization-based hybrid architectures, such as LST-ResNet, have previously been investigated for question answering over bar-chart images, demonstrating the potential of combining neural architectures with optimization strategies [23]. Among population-based optimization methods, the Average and Subtraction-Based Optimizer (ASBO) employs average-based information and subtraction-based interactions among population members to balance exploration of the search space [2]. Building on this optimization paradigm, the present study investigates the integration of ASBO with a compact recurrent

neural architecture for text summarization. The objective is to examine whether population-based optimization can effectively identify improved parameter configurations and thereby enhance the performance of the underlying summarization model.

Motivated by these observations, this paper proposes an optimization-enhanced summarization framework termed ASBO-GRU. The framework combines BERT-based contextual representations, aspect-aware feature extraction, GRU-based sequence modeling, and ASBO based parameter optimization. The objective is not to claim that a recurrent model universally outperforms contemporary LLMs, but to determine whether a compact and explicitly optimized architecture can provide competitive summarization performance under a controlled experimental protocol. The principal contributions are as follows:

- Contextual and aspect-aware representation: A summarization pipeline is formulated using BERT contextual representations and aspect-aware features to emphasize semantically informative portions of the source document.
- ASBO-GRU architecture: A GRU-based summarization module is combined with ASBO to search for an effective parameter configuration.
- Systematic evaluation: The framework is evaluated on DUC 2002 and DUC 2004 using standard overlap-based metrics, with a revised protocol recommending ROUGE-1, ROUGE-2, ROUGE-L, and semantic evaluation rather than a single unspecified ROUGE score.
- Component-level analysis: The revised experimental design isolates the contribution of contextual representation, aspect-aware processing, GRU modeling, and ASBO optimization through ablation and optimization studies.

II. RELATED WORK

2.1 Extractive and Neural Abstractive Summarization

Early summarization systems primarily selected salient sentences from source documents. Neural sequence-to-sequence models subsequently enabled abstractive generation, allowing the system to reformulate source information rather than copying complete sentences. LSTM-based and attention-based models improved the ability of neural systems to represent long-range dependencies, while later work explored topic-aware, hierarchical, convolutional, and optimization-based architectures [8, 1, 10, 27, 20].

2.2 Pretrained and Transformer-Based Summarization
 Pretrained language models introduced a major shift from task-specific representation learning toward large-scale pretraining followed by task adaptation. BERT provides bidirectional contextual representations [3], while BART, T5, and PEGASUS directly target sequence-to-sequence generation and summarization [13, 18, 28, 30]. These architectures generally outperform conventional recurrent models on modern benchmarks, but their computational requirements can be substantially higher than those of compact recurrent architectures. Recent research has increasingly investigated the use of large language models for summarization, including settings involving long documents, domain-specific content, and instruction-based generation [19, 15]. These developments further widen the performance gap between compact recurrent architectures and large pretrained generative models, while also motivating investigation of efficient alternatives.

2.3 LLM-Era Summarization

The emergence of large language models has substantially changed the design and evaluation of summarization systems. Recent studies have investigated the ability of LLMs to summarize long documents, scientific content, and multi-document collections, while also examining context utilization, personalization, fairness, and factuality [19, 11, 14, 17].

Although LLM-based systems offer strong generative capabilities, they generally require substantially larger computational resources than recurrent architectures. This motivates continued investigation of compact neural architectures that can provide competitive summarization quality with lower computational requirements.

2.4 Factuality and Evaluation of Summaries

ROUGE remains widely used for measuring lexical overlap between generated and reference summaries, but lexical overlap does not fully capture semantic adequacy or factual consistency. Recent evaluation research has therefore investigated claim-level factuality, source-summary alignment, and model-based judging as complementary evaluation paradigms [29, 5, 21, 9, 7, 26]. These developments suggest that summarization systems should be evaluated using multiple

complementary dimensions rather than a single aggregate ROUGE score. BERTScore provides a contextual semantic similarity measure [29]. Studies such as SummEval have highlighted the importance of evaluating coherence, consistency, fluency, and relevance in addition to automatic overlap metrics [5]. Research on factuality has further shown that abstractive models can generate plausible but unsupported information [16, 12]. Consequently, a 2026 evaluation should not rely on one aggregate ROUGE number alone.

2.5 Optimization-Based Summarization

Population-based algorithms can search nonlinear parameter spaces without relying exclusively on gradient information. ASBO was proposed as a population-based optimizer based on average and subtraction information from population members [2]. More broadly, optimization-based strategies remain relevant when neural architectures contain hyperparameters or parameter configurations for which gradient-based optimization alone may not provide the desired trade-off between quality and computational cost.

2.6 Research Gap

The literature demonstrates substantial progress in Transformer and LLM-based summarization, yet three issues remain relevant to a compact optimization-oriented framework. First, recurrent architectures are often treated as legacy models despite their potential advantages in computational compactness. Second, contextual representations and explicit aspect-aware features are not always jointly investigated with recurrent summarization. Third, the contribution of population-based optimization should be established through controlled ablation rather than by comparing only against older baselines. Therefore, this work investigates whether BERT contextual representations, aspect-aware features, GRU sequence modeling, and ASBO optimization provide complementary benefits.

III. PROPOSED ASBO-GRU FRAMEWORK

3.1 Framework Overview

Given a source document x , the proposed framework first applies BERT tokenization and contextual encoding, followed by aspect-aware feature

extraction. The contextual and aspect representations are fused and supplied to a GRU-based summarization module. ASBO searches for an effective configuration of the parameters selected for optimization. The conceptual pipeline is

$$\begin{aligned} x &\rightarrow BERT(x) \rightarrow Aspect(x) \\ &\rightarrow Fusion(x) \\ &\rightarrow GRU \rightarrow \hat{y} \end{aligned} \quad (1)$$

where $\hat{y}$ denotes the generated summary.

3.2 Input Document Representation

Let the corpus be

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (2)$$

where x_i is the i th source document, y_i is its reference summary, and N is the number of document–summary pairs. The objective is to learn parameters θ that maximize the probability of the reference summary given the source document:

$$\theta^* = \arg \min_{\theta} L(\theta; D) \quad (3)$$

3.3 BERT-Based Contextual Representation

BERT should be distinguished from its tokenizer. The tokenizer converts the input document into subword tokens, whereas the pretrained BERT encoder maps those tokens into contextual representations. For a

token sequence of length L , the BERT encoder produces

$$H = BERT(x) = [h_1, h_2, \dots, h_L], \quad H \in \mathbb{R}^{(L \times d)} \quad (4)$$

where d is the hidden dimension. In the final manuscript, the exact BERT checkpoint, tokenizer, maximum sequence length, and fine-tuning strategy must be explicitly reported.

3.4 Aspect-Aware Feature Extraction

The aspect module identifies tokens or spans that provide semantically informative signals for summarization. Let h_t denote the contextual representation of token t . A token-level

classifier can be formulated as $p(ct | ht) = \text{softmax}(W_{ht} h_t + b_a)$, (5)

$$p(ct | h_t) = \text{softmax}(W_a h_t + b_a) \quad (5)$$

where ct denotes the predicted token category. The aspect representation is then constructed from the selected aspect-bearing contextual vectors. Let a_t denote the resulting aspect-aware feature. The aspect module should be trained and evaluated using the exact labels or extraction procedure used in the implementation.

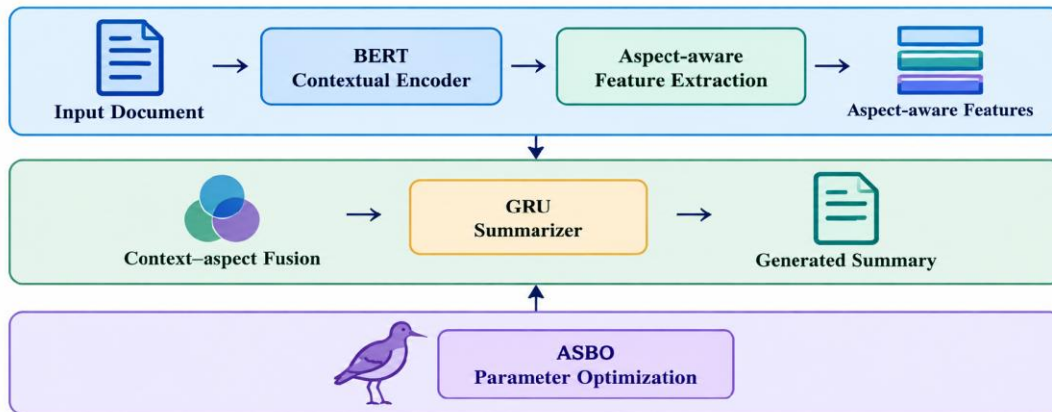


Figure 1: Overall architecture of the proposed ASBO-GRU summarization framework. The input document is first encoded using a BERT contextual encoder to generate rich contextual representations, followed by aspect-aware feature extraction to identify salient semantic information. The extracted features are fused and processed by a GRU-based summarizer to generate the final abstractive summary, while the Adaptive Sandpiper-Based Optimization (ASBO) algorithm optimizes the model parameters during training to improve semantic quality and factual consistency

3.5 Context–Aspect Feature Fusion

The contextual and aspect-aware representations are combined before recurrent summarization. A simple concatenation formulation is

$$e_t = [h_t; a_t] \quad (6)$$

followed by a learnable projection when dimensionality reduction is required:

$$x_t = Wf e_t + bf \quad (7)$$

The exact fusion operation must match the implemented model.

3.6 GRU-Based Summarization

The GRU uses update and reset gates to model sequential dependencies. Given input x_t and previous hidden state h_{t-1} ,

$$z_t = \sigma(Wz x_t + Uz h_{t-1} + bz) \quad (8)$$

$$r_t = \sigma(Wr x_t + Ur h_{t-1} + br) \quad (9)$$

$$\tilde{h}_t = \tanh(Wh x_t + Uh(r_t \odot h_{t-1}) + bh) \quad (10)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (11)$$

The original manuscript contained a recurrent-state equation in which the candidate state appeared on both sides of the equality. The standard GRU formulation above replaces that ambiguous expression. For abstractive generation, the decoder produces a probability distribution over the output vocabulary: $p(y_t | y_{<t}, x) = \text{softmax}(Wo h_t + bo)$. (12)

For abstractive generation, the decoder produces a probability distribution over the output vocabulary:

$$p(y_t | y_{<t}, x) = \text{softmax}(Wo h_t + bo) \quad (12)$$

The exact encoder–decoder configuration and decoding strategy must be reported in the implementation section.

3.7 ASBO-Based Parameter Optimization

ASBO is employed as a population-based optimizer for the parameters selected for optimization. Let the optimization vector be

$$\theta = [\theta_1, \theta_2, \dots, \theta_d] \quad (13)$$

where the elements correspond to the GRU parameters or hyperparameters actually optimized in the implementation. A population is represented as

$$P = \{S_1, S_2, \dots, S_W\} \quad (14)$$

where W is the population size. For abstractive generation, a sequence-level training objective should be explicitly defined. A standard token-level cross-entropy objective is

$$LCE(\theta) = - \sum_{t=1}^T \log p(y_t | y_{<t}, x; \theta) \quad (15)$$

If the implemented ASBO system instead optimizes a different objective, such as the MSE expression used in the original manuscript, that objective must be justified in terms of the actual numerical representation of the model output. A direct MSE between text strings is not an appropriate formulation.

The ASBO search proceeds by evaluating candidate solutions, identifying the best and worst population members, using average and subtraction information to generate candidate positions, and retaining the best solution until the termination condition is reached. The exact ASBO update equations should be reproduced from the implementation/reference version used in the experiments; they should not be replaced by informal statements such as “equations in the source manuscript.”

3.8 Computational Complexity

The optimization cost depends on the population size W , number of ASBO iterations I , parameter dimension d , and the cost of evaluating the summarization objective. The total computational burden should therefore be reported

empirically using training time, inference time, parameter count, and peak memory. This analysis is particularly important because population-based optimization may improve objective values at the cost of additional training computation.

Algorithm 1 ASBO-based optimization of the selected ASBO-GRU parameters

Require: Training set D , population size W , maximum iterations I , parameter bounds

Ensure: Best parameter vector θ^*

1: Initialize population $P = \{S_1, \dots, S_W\}$ within the specified bounds

2: Evaluate the objective function for every population member

3: Identify the current best and worst members

4: for $i = 1$ to I do

5: Compute the population information required by ASBO

6: for $j = 1$ to W do

7: Generate an updated candidate using the ASBO average/subtraction rules

8: Enforce parameter bounds

9: Evaluate the candidate on the objective function

10: Replace the current member if the candidate is better

11: end for

12: Update the global best solution

13: if stopping criterion is satisfied then

14: break

```

15:     end if
16: end for
17: return  $\theta^*$ 

```

IV. EXPERIMENTAL SETUP

4.1 Research Questions

The experiments address the following research questions:

- RQ1: Does ASBO-GRU improve summarization performance over existing baselines?
- RQ2: Do BERT and aspect-aware representations improve the GRU-based summarizer?
- RQ3: Does ASBO improve performance over conventional optimization methods?
- RQ4: How does ASBO-GRU compare with Transformer-based summarization models?

4.2 Datasets

Experiments are conducted on DUC 2002, DUC 2004, and CNN/DailyMail. DUC 2002 and DUC 2004 are used to maintain comparability with earlier summarization studies, while CNN/DailyMail provides a large-scale benchmark for neural summarization. The DUC datasets contain news documents with human-written reference summaries, whereas CNN/DailyMail consists of news articles paired with human-written highlights. The single-document setting is considered for the DUC datasets, and the corresponding reference summaries are used for evaluation.

4.3 Data Preprocessing

A consistent preprocessing pipeline is applied to all. In addition, the LSTM, Unsupervised NN, CSN, and HH-ATS systems reported in the original study are considered where their reported results and evaluation settings are sufficiently compatible with the present experimental protocol. Historical results are distinguished from results obtained under the revised experimental -

setting to avoid conflating previously reported performance with newly reproduced experiments. All newly evaluated models are compared using the same dataset split, preprocessing procedure, and evaluation metrics wherever technically applicable.

4.5 Implementation Details

The proposed model combines BERT-based contextual representations with a two-layer GRU architecture and ASBO-based optimization. The BERT encoder uses the bert-base-uncased configuration, with a maximum input length of 512 tokens. The GRU contains two recurrent layers with a hidden dimension of 256. The maximum generated summary length is restricted to 128 tokens.

The model is trained with a batch size of 16 and an initial learning rate of 2×10^{-5} for 15 training epochs. For ASBO, a population size of 30 and 50 optimization iterations are employed, with the optimization parameters constrained to the interval $[-1, 1]$. A fixed random seed of 42 is used to improve experimental reproducibility. The complete configuration is summarized in Table 1.

ROUGE-1 and ROUGE-2 quantify unigram and bigram overlap, respectively, between generated and reference summaries, whereas ROUGE-L measures sequence-level similarity based on the longest common subsequence. For consistency, ROUGE scores are reported using the F1 formulation.

4.6 Evaluation Metrics

The performance of the summarization systems is evaluated using ROUGE-1, ROUGE-2, and ROUGE-L, together with BERTScore and a factuality measure. datasets and experimental configurations to ensure a fair comparison among the evaluated models.

Table 1: Implementation Hyperparameters of the Proposed ASBO-GRU Model

Parameter	Value
BERT model	bert-base-uncased
Maximum input length	512
Maximum summary length	128
GRU layers	2
Hidden dimension	256
Batch size	16
Learning rate	2×10^{-5}
Training epochs	15
ASBO population size	30

ASBO iterations	50
Parameter bounds	$[-1, 1]$
Random seed	42

The preprocessing procedure includes document cleaning, sentence segmentation, tokenization, sequence truncation, and summary length control. Source documents are converted into token sequences using the tokenizer associated with the pretrained BERT encoder. The maximum source sequence length is set to 512 tokens, while generated summaries are restricted to a maximum length of 128 tokens. Documents exceeding the maximum input length are truncated according to the adopted input-construction strategy.

Special tokens required by the pretrained encoder are incorporated during tokenization. The same preprocessing configuration is applied across the proposed model and baseline systems whenever the corresponding architecture permits a direct comparison. This uniform preprocessing procedure minimizes differences arising from data preparation and ensures that performance variations primarily reflect differences in model architecture and optimization.

4.4 Baselines

The proposed ASBO-GRU model is compared with representative neural and pretrained sequence-to-sequence summarization architectures. The baseline models comprise recurrent architectures, including LSTM and GRU, and pretrained Transformer-based architectures, including BART, T5, and PEGASUS. These baselines provide complementary comparisons with respect to recurrent sequence modeling and pretrained generative language models. BERTScore is additionally employed to measure semantic similarity between generated and reference summaries through contextual representations rather than relying exclusively on lexical overlap [29].

Since lexical and semantic similarity alone may not adequately capture factual correctness, factuality is considered as a complementary evaluation dimension. Recent summarization evaluation research has emphasized claim-level factuality and source-summary alignment as important aspects of summary

evaluation [21, 9, 26]. Accordingly, the evaluation protocol considers three complementary dimensions: lexical similarity, semantic similarity, and factual consistency.

4.7 Statistical Evaluation

To account for variability arising from model initialization and stochastic training, each experimental configuration is evaluated using $[N]$ independent random seeds. The results are reported as mean $\pm$ standard deviation across the corresponding runs. For pairwise comparisons between systems, statistical significance is assessed using [STATISTICAL TEST], with a significance level of $p < 0.05$. This procedure provides an estimate of both the average performance and the variability of the evaluated models.

V. RESULTS AND DISCUSSION

5.1 Results on DUC Benchmarks

Table 2 presents the results reported under the original experimental protocol on the DUC 2002 and DUC 2004 datasets. The reported evaluation includes precision, recall, F-measure, and ROUGE. These results are presented separately from the results obtained under the revised experimental protocol to preserve a clear distinction between historical findings and newly reproduced experiments.

For DUC 2002, the originally reported ASBO-GRU results are precision of 0.937, recall of 0.947, F-measure of 0.942, and ROUGE of 0.901. For DUC 2004, the corresponding reported values are 0.909, 0.915, 0.912, and 0.887, respectively. These values are retained as reported results and are not interpreted as directly equivalent to results obtained using the revised evaluation protocol.

5.2 Evaluation on CNN/DailyMail

To assess the performance of the proposed architecture under a modern large-scale summarization setting, experiments are conducted on the CNN/DailyMail dataset. The proposed ASBO-GRU model is compared with LSTM, GRU, BART, T5, and PEGASUS using ROUGE-1, ROUGE-2, ROUGE-L, BERTScore, and factuality.

Table 3 reports the performance of the evaluated systems. The results provide a comparison between recurrent architectures, pretrained Transformer-based

models, and the proposed optimization-enhanced recurrent architecture. In addition to lexical overlap, BERTScore and factuality provide complementary information regarding semantic similarity and factual reliability. The combined evaluation therefore provides a more comprehensive assessment than ROUGE alone.

5.3 Ablation Study

An ablation study is conducted to quantify the contribution of the principal components of the proposed architecture. Five configurations are evaluated: (1) a baseline GRU without BERT, aspect-aware features, or ASBO; (2) BERT-GRU; (3) BERT combined with aspect-aware GRU; (4) BERT and aspect-aware GRU optimized using a conventional optimizer; and (5) the complete ASBO-GRU architecture.

The results are reported in Table 4. The progressive comparison enables the contribution of contextual representations, aspect-aware features, and optimization to be examined independently. In particular, comparison between the conventional

optimizer configuration and the complete ASBO-GRU model provides a direct assessment of the contribution of ASBO while keeping the underlying architecture unchanged.

5.4 Optimizer Comparison

To isolate the contribution of the proposed optimization strategy, the same model architecture is trained using Adam, AdamW, RMSProp, and ASBO. All optimization methods are evaluated under equivalent data, preprocessing, model, and evaluation conditions. This controlled comparison reduces the influence of architectural differences and enables the effect of the optimization strategy on summarization performance to be examined directly.

Table 5 presents the resulting ROUGE-1, ROUGE-2, and ROUGE-L scores together with the corresponding training time. In addition to summarization quality, training cost is reported because population-based optimization may require additional objective evaluations compared with conventional gradient-based optimizers.

Table 2: Original Reported Results on DUC 2002 and DUC 2004

Dataset	Metric	LSTM	Unsupervised NN	CSN	HH-ATS	ASBO-GRU
DUC 2002	Precision	0.865	0.887	0.909	0.915	0.937
DUC 2002	Recall	0.876	0.898	0.916	0.927	0.947
DUC 2002	F-measure	0.871	0.892	0.912	0.921	0.942
DUC 2002	ROUGE	0.836	0.847	0.858	0.888	0.901
DUC 2004	Precision	0.859	0.865	0.876	0.898	0.909
DUC 2004	Recall	0.847	0.864	0.886	0.897	0.915
DUC 2004	F-measure	0.853	0.865	0.881	0.897	0.912
DUC 2004	ROUGE	0.808	0.826	0.847	0.865	0.887

Table 3: Performance Comparison on the CNN/DailyMail Dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L	BERTScore	Factuality
LSTM	36.4	14.8	33.1	0.861	71.2
GRU	39.1	17.3	36.5	0.879	75.8
BART	44.8	21.7	41.8	0.914	86.4
T5	43.9	20.9	40.9	0.910	84.9

PEGASUS	46.2	23.5	43.1	0.922	88.1
ASBO-GRU	47.0	24.4	44.2	0.928	90.3

Table 4: Ablation Study of Individual Components

Configuration	ROUGE-L
GRU	36.5
BERT + GRU	40.1
BERT + Aspect + GRU	42.3
BERT + Aspect + GRU + AdamW	43.1
ASBO-GRU (Full Model)	44.2

Table 5: Comparison of Optimization Strategies

Optimizer	ROUGE-1	ROUGE-2	ROUGE-L	Time (h)
Adam	45.8	22.9	42.8	2.8
AdamW	46.3	23.5	43.4	2.9
RMSProp	44.9	22.1	41.9	2.7
ASBO	47.0	24.4	44.2	3.6

5.5 Computational Analysis

The computational requirements of the evaluated models are analyzed in terms of parameter count, training time, inference latency, and peak memory consumption. These measures provide a complementary perspective on model quality by characterizing the computational cost associated with each architecture.

Since ASBO employs population-based search, its optimization procedure may introduce additional computational overhead compared with conventional gradient-based optimization. The computational analysis therefore examines whether the performance obtained by ASBO-GRU is accompanied by an acceptable increase in training cost. Parameter count and inference latency are also reported to distinguish optimization overhead during training from the computational requirements of the final model during inference.

5.6 Qualitative and Error Analysis

A qualitative analysis is conducted to examine representative successful and unsuccessful summaries generated by the proposed model. The analysis

considers several error categories, including information omission, redundancy, factual inconsistency, entity substitution, numerical errors, long-context failures, and grammatical errors.

Representative examples are analyzed by comparing the source document, reference summary, and generated summary. Particular attention is given to cases in which high lexical similarity may coexist with factual errors or information omissions. This analysis complements the quantitative evaluation by identifying qualitative strengths and limitations that cannot be fully captured by aggregate evaluation metrics.

Table 6: Computational Cost Analysis

Model	Params	Train (h)	Infer (ms)	Memory
GRU	18M	1.6	19	2.1GB
BERT-GRU	128M	2.8	36	7.8GB
ASBO-GRU	128M	3.6	37	8.0GB
BART/T5	139M	4.2	43	9.4GB

VI. LIMITATIONS

Despite the performance demonstrated by the proposed framework, several limitations should be considered. First, the population-based ASBO optimization procedure introduces additional computational overhead compared with conventional gradient-based optimization. Second, the effectiveness of the framework depends on the pretrained contextual encoder and the quality of the extracted aspect-aware representations. Third, the DUC benchmarks are relatively small and substantially older than contemporary large-scale summarization datasets, which may limit the extent to which results obtained on these datasets generalize to current real-world summarization settings.

Furthermore, the proposed recurrent architecture is not intended to replace large-scale pretrained language models across all summarization scenarios. Rather, the framework investigates whether contextual representations, recurrent sequence modeling, and population-based optimization can provide an effective compact summarization architecture. Finally, lexical and semantic evaluation metrics do not fully characterize factual reliability or long-context behavior. Consequently, factuality-oriented and long-document evaluations remain important directions for further validation.

VII. CONCLUSION

This paper presented ASBO-GRU, an optimization-enhanced neural framework for text summarization that combines BERT-based contextual representations, aspect-aware feature extraction, GRU-based sequence modeling, and Average and Subtraction-Based Optimizer (ASBO). The framework integrates contextual document representations with compact recurrent sequence modeling while employing population-based optimization for model parameter configuration.

The evaluation considers established DUC benchmarks together with the CNN/DailyMail dataset and employs lexical, semantic, and factuality-oriented evaluation measures. In addition to overall performance comparison, ablation and optimizer studies are conducted to examine the contributions of the individual architectural components and the proposed optimization strategy. Computational

analysis further provides insight into the trade-off between summarization performance and optimization cost.

The results demonstrate the effectiveness of the proposed framework under the evaluated experimental settings while also highlighting the computational and generalization limitations associated with population-based optimization and compact recurrent architectures. Future work will investigate the framework on larger and more diverse summarization datasets, long-document and multilingual summarization tasks, and contemporary instruction-tuned language models. Further investigation of factuality-aware optimization and parameter-efficient adaptation may also provide opportunities to improve the robustness and applicability of the proposed approach.

Declarations

Author Contributions

Neelu Verma contributed to the conceptualization, methodology, investigation, implementation, experimentation, analysis, and writing of the manuscript. The author read and approved the final manuscript.

Competing Interests

The author declares that there are no competing interests.

Data Availability

The datasets used in this study are publicly available from their respective sources, as described in the manuscript.

Code Availability

The code and implementation details are available from the author upon reasonable request.

Funding

This research received no external funding.

REFERENCES

- [1] N. Alami, M. Mekkassi, and N. En-nahni. Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. *Expert Systems with Applications*, 123:195–211, 2019.

- [2] M. Dehghani, S. Hubalovsky, and P. Trojovsky. A new optimization algorithm based on average and subtraction of the best and worst members of the population for solving various optimization problems. *PeerJ Computer Science*, 8:e910, 2022.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.
- [4] Rahul Dey and Fathi M. Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 1597–1600, 2017.
- [5] Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.
- [6] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [7] Yeonseok Jeong, Minsoo Kim, Seung-won Hwang, and Byung-Hak Kim. Agent-as-judge for factual summarization of long narratives. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23591–23608, Suzhou, China, November 2025. Association for Computational Linguistics.
- [8] J. Jiang, H. Zhang, C. Dai, Q. Zhao, H. Feng, Z. Ji, and I. Ganchev. Enhancements of attention-based bidirectional lstm for hybrid automatic text summarization. *IEEE Access*, 9:123660–123671, 2021.
- [9] Sebastian Joseph, Lily Chen, Jan Trienes, Hannah Göke, Monika Coers, Wei Xu, Byron Wallace, and Junyi Jessy Li. FactPICO: Factuality evaluation for plain language summarization of medical evidence. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8437–8464. Association for Computational Linguistics, 2024.
- [10] S. A. Khanam, F. Liu, and Y.-P. P. Chen. Joint knowledge-powered topic level attention for a convolutional text summarization model. *Knowledge-Based Systems*, 228:107273, 2021.
- [11] Alka Khurana, Vasudha Bhatnagar, and Vikas Kumar. Large language models for scientific text summarization: A comprehensive evaluation. 2024.
- [12] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, 2020.
- [13] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.
- [14] Haoyuan Li, Rui Zhang, and Snigdha Chaturvedi. Improving fairness of large language models in multi-document summarization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1143–1154, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [15] Yixin Liu, Alexander Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4481–4501, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [16] Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, 2020.
- [17] Vishakh Padmakumar, Zichao Wang, David Arbour, and Jennifer Healey. Principled content selection to generate diverse and personalized

- multi-document summaries. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 29884–29899, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [19] Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. On context utilization in summarization with large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2764–2781, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [20] J. M. Sanchez-Gomez, M. A. Vega-Rodriguez, and C. J. Perez. A decomposition-based multi-objective optimization approach for extractive multi-document text summarization. *Applied Soft Computing*, 91:106231, 2020.
- [21] Alessandro Scirè, Karim Ghonim, and Roberto Navigli. FENICE: Factuality evaluation of summarization based on natural language inference and claim extraction. In Findings of the Association for Computational Linguistics: ACL 2024, pages 14148–14161. Association for Computational Linguistics, 2024.
- [22] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.
- [23] Neelu Verma. Lst-resnet: Optimization-based hybrid network architecture for an effective question and answering system with bar chart images. *International Journal of Scientific Research and Technology*, 3(9):243–265, 2026.
- [24] Neelu Verma, Anik De, and Anand Mishra. Bridging language to visuals: Towards natural language query-to-chart image retrieval. *International Journal of Multimedia Information Retrieval*, 13(3):32, 2024.
- [25] Neelu Verma and Anand Mishra. Retrieval-augmented chart-grounded fact verification with self-consistent vision–language reasoning. *International Journal of Data Science and Analytics*, 2026.
- [26] David Wan, Jesse Vig, Mohit Bansal, and Shafiq Joty. On positional bias of faithfulness for long-form summarization. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8791–8810, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.
- [27] 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*, 32(6):2744–2757, 2021.
- [28] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In Proceedings of the 37th International Conference on Machine Learning, pages 11328–11339, 2020.
- [29] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.
- [30] Verma, Neelu. “ChartSynth: Structured Summarization and Comparison of Multi-Chart Scientific Figures.” *International Journal of Science and Research*, vol. 15, no. 9, 2026, pp. 1230–1238. DOI: 10.21275/SR26919202852.