Pertanika J. Sci. & Technol. 33 (6): 2855 - 2879 (2025)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Enhancing Extractive Text Summarization Using Semantic-Based Similarity Matching

Junjie Yang, Keng Hoon Gan* and Jun Wang

School of Computer Sciences, Universiti Sains Malaysia, 11800 Gelugor, Pulau Pinang, Malaysia

ABSTRACT

Due to the exponential growth of online information, the ability to efficiently extract key content and target information without requiring extensive reading is becoming increasingly important for readers. This paper investigates the construction of neural extractive summarization systems by framing the task as a semantic text matching problem. The proposed approach, named MatchDocSum, aligns the source document with potential summaries within a semantic space, leveraging pretrained language model contextual representations to enhance the understanding of their interconnectedness. The goal is to address the limitations of conventional methods, which often struggle with capturing intricate semantic relationships and producing coherent summaries. Hence, this study proposes an enhanced document summary matching framework to investigate three main aspects that affect the outcome of a good summary: document pruning, text embedding, and similarity matching measure within the framework. MatchDocSum was evaluated on the Cable News Network (CNN)/DailyMail dataset, showing competitive results against several baselines, including LEAD and bidirectional encoder representations from transformers (BERT) for extractive summarization (BERTSUM). The results demonstrate that our approach performs better than baseline models in some aspects, achieving Recall-oriented Under for Gisting Evaluation (ROUGE)-1 scores of 43.50, ROUGE-2 scores of 20.45, and ROUGE-L scores of 40.75.

Keywords: Extractive summarization, natural language processing, semantic similarity, text summarization

ARTICLE INFO

Article history:
Received: 10 January 2025
Accepted: 29 April 2025
Published: 29 October 2025

DOI: https://doi.org/10.47836/pjst.33.6.19

E-mail addresses:
yangjunjie2023@student.usm.my (Junjie Yang)
khgan@usm.my (Keng Hoon Gan)
wangjun@student.usm.my (Jun Wang)
* Corresponding author

INTRODUCTION

Document summarization condenses essential information while preserving key details, tailored to specific document types. News articles focus on “who, what, when, where, why, and how”, research papers emphasize findings and implications, legal documents distill cases and decisions (Zhong et al., 2020), and book summaries

© Universiti Putra Malaysia Press

highlight plots and themes. Summarization enables readers to efficiently process vast amounts of information.

Extractive summarization identifies core information by selecting relevant sentences. Techniques like Named Entity Recognition (NER), syntactic analysis, and algorithms like PageRank (PR) (Pradhan et al., 2013) effectively handle legal texts. Scientific literature uses citation data to highlight findings (Beltagy et al., 2020), while news articles rely on machine learning models like T5 (Raffel et al., 2020) for key sentence extraction.

Graph-based methods like TextRank (Mihalcea & Tarau, 2004) and LexRank (Erkan & Radev, 2004) rank sentences based on positioning and frequency. Recent advancements in deep learning, including BERT (Devlin et al., 2018) and BERTSUM (Liu & Lapata, 2019), improve contextual understanding and summarization by leveraging segment embeddings. T5 adopts a versatile text-to-text approach, excelling across domains.

As textual data grows exponentially, summarization becomes crucial for generating concise, accurate summaries in legal, scientific, and news domains. Semantic matching ensures coherence and relevance, with models like BERT capturing deep contextual relationships for complex documents. These advancements enhance the effectiveness and applicability of summarization techniques across diverse fields.

The task of extractive summarization (Nallapati et al., 2016) within the MatchSum framework faces several limitations. While BERTSUM (Liu & Lapata, 2019) effectively extracts sentence-level information, it struggles to capture sequential relationships within sentences, leading to potential loss of contextual coherence. Additionally, BERT's transformer-based architecture operates in parallel, relying on positional embeddings to approximate word order, which may compromise its ability to understand sentence structure (Vaswani et al., 2017).

BERT has limited performance on long texts due to the quadratic complexity of its self-attention mechanism (Devlin et al., 2018) and its fixed-length input constraints (Beltagy et al., 2020), leading to information loss and inefficient summarization. In addition, the text embeddings generated by BERT and robustly optimized BERT pretraining approach (RoBERTa) have difficulty in capturing subtle relationships and long-distance dependencies in complex documents. In contrast, decoding-enhanced BERT with disentangled attention (DeBERTa), proposed by He et al. (2020), enhances comprehension and generation via untangled attention and outperforms BERT and RoBERTa in modeling long texts and complex semantic relationships.

The MatchSum framework's reliance on cosine similarity to select candidate summaries introduces additional shortcomings. Cosine similarity inadequately captures word order and contextual importance, often leading to summaries that lack semantic richness and coherence (Zhong et al., 2020). Inspired by Zhong et al. (2020), the MatchDocSum architecture was designed to match documents with candidate summaries. The name combines "match", "doc", and "sum" to select the summary that best matches the content of

a document in the semantic space. Unlike BERTSUM-based MatchSum, which relies only on cosine similarity, MatchDocSum uses DeBERTa to encode documents and candidate summaries.

This study explores an improvised method for extractive summarization to address these challenges. It uses the recurrent neural network (RNN)-based SummaRuNNer (Nallapati et al., 2016) model to prune documents and compares its performance with BERTSUM (Liu & Lapata, 2019) on the CNN/DailyMail dataset. DeBERTa is tested as a text encoder in the MatchDocSum framework alongside BERT and RoBERTa. The goal is to identify which model creates the best semantic alignment between documents and candidate summaries. Additionally, it evaluates dot product similarity as an alternative to cosine similarity, seeking a more precise and contextually relevant metric for summary evaluations.

MATERIALS

Text summarization has progressed from traditional statistical methods, like term frequency–inverse document frequency (TF-IDF) and LexRank, which focused on term frequencies and sentence similarity but struggled with long texts and complex semantics, to deep learning models such as convolutional neural networks, RNNs, and sequence-to-sequence (Seq2Seq) models. These modern approaches use encoder-decoder architectures to capture global context in abstractive summarization, but often suffer from errors and irrelevant content.

The advent of pre-trained language models has transformed text summarization. BERTSUM (Liu & Lapata, 2019) enhances extractive summarization by incorporating sentence-level classification into BERT, which improves contextual understanding. MatchSum (Zhong et al., 2020) redefines the task as a text-matching problem and optimizes sentence selection to ensure coherence and relevance, achieving high ROUGE scores. However, challenges related to accuracy, coherence, and computational efficiency still exist. This study refines BERTSUM and MatchSum to improve summarization quality and broaden their applicability across diverse domains.

Extractive Summarization Methods

X. Zhang et al. (2019) presented HiBERT, a hierarchical transformer model built on BERT specifically designed to handle long documents. This model processes text at both the word and sentence levels, enabling it to handle long-range dependencies and summarize extended texts effectively.

BERT-based models have also demonstrated flexibility in various domains. Dutulescu et al. (2022) proposed an unsupervised BERT model tailored for summarizing clinical reports, demonstrating its ability to adapt to diverse text types. In another development, Yao et al. (2018) introduced a method combining BERT with reinforcement learning. This

approach uses reward signals such as readability and informativeness to enhance summary quality, advancing the evaluation of summarization models.

Among prominent models, Liu and Lapata (2019) introduced BERTSUM, an extension of BERT (Devlin et al., 2018) tailored for extractive summarization. BERTSUM enhances BERT's input schema by adding [CLS] and [SEP] tokens for sentence representation and uses interval segment embeddings to differentiate document sentences. The model incorporates summarization-focused layers, including classifiers, inter-sentence Transformer layers, and RNNs, to improve document-level feature capture. On the CNN/DailyMail and New York Times datasets, BERTSUM achieved state-of-the-art ROUGE scores, demonstrating its practical potential for extractive summarization.

Nallapati et al. (2016) introduced SummaRuNNer, an RNN-based framework for extractive summarization framed as sequence classification. Using a bi-directional gated recurrent unit (GRU)-RNN, it processes text hierarchically, capturing fine-grained and document-level dependencies. The model's interpretability stems from its consideration of features like richness, salience, novelty, and position. A novel training mechanism generates approximate extractive labels using a greedy algorithm to optimize ROUGE scores, bypassing reliance on extractive labels. SummaRuNNer performed strongly on CNN/DailyMail and New York Times datasets, achieving high ROUGE metrics, making it a robust tool for extractive summarization.

Hybrid and two-stage summarization approaches have also gained traction. These methods typically involve an extraction phase to select key content, followed by a refinement or compression phase. Early works by Alyguliyev (2009), Galanis et al. (2012), and H. Zhang et al. (2019) extracted important fragments and refined them into coherent summaries. Bae et al. (2019), as well as Chen and Bansal (2018), introduced hybrid frameworks incorporating reinforcement learning to bridge content extraction and rewriting. Liu and Lapata (2019) and Zhong et al. (2020) advanced the "extract-then-compress" methodology, training extractors to identify relevant content and condense it into concise summaries, significantly enhancing summarization quality.

Pre-Trained Models Used in Text Summarization

Extractive summarization selects key sentences from a document to create concise summaries. The introduction of BERT (Devlin et al., 2018) revolutionized this field by capturing semantic relationships and contextual information. As a pre-trained transformer-based model, BERT learns deep bidirectional representations by conditioning on both left and right contexts. When fine-tuned for summarization, it consistently outperforms traditional methods. Figure 1 illustrates the BERT architecture.

RoBERTa enhances BERT's performance by refining its pretraining methods. It removes the Next Sentence Prediction (NSP) task, focusing entirely on Masked Language

Modeling (MLM) to better capture contextual relationships. Trained on a larger dataset (over 160 GB), RoBERTa incorporates extended training times, larger batch sizes, and dynamic masking to prevent overfitting and enhances contextual understanding. These optimizations allow RoBERTa to achieve new benchmarks on tasks like General Language Understanding Evaluation (GLUE) and SquAD, and highlight the significant impact of improved pretraining on generalization. Figure 2 illustrates RoBERTa's architecture.

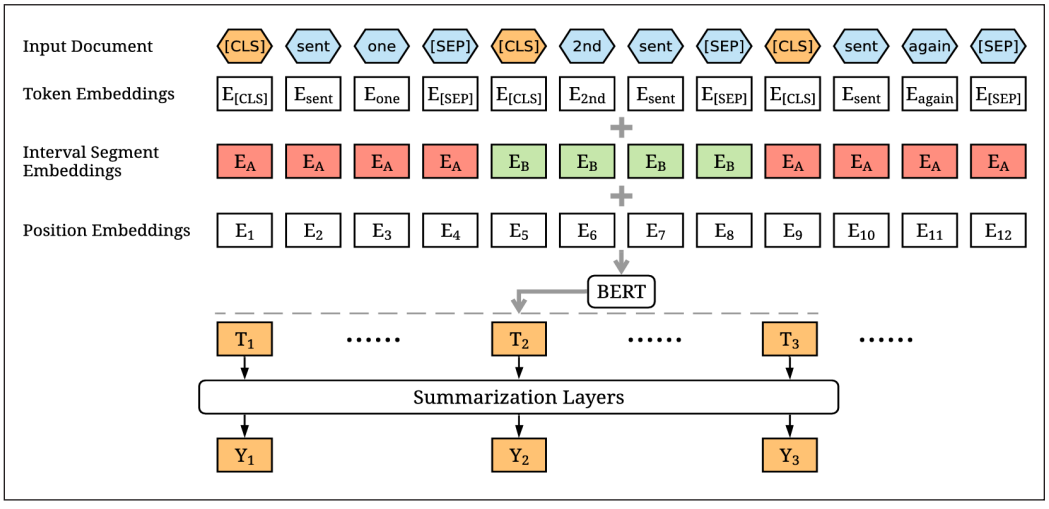

Figure 1. Architecture of bidirectional encoder representations from transformers (BERT) (Liu, 2019)

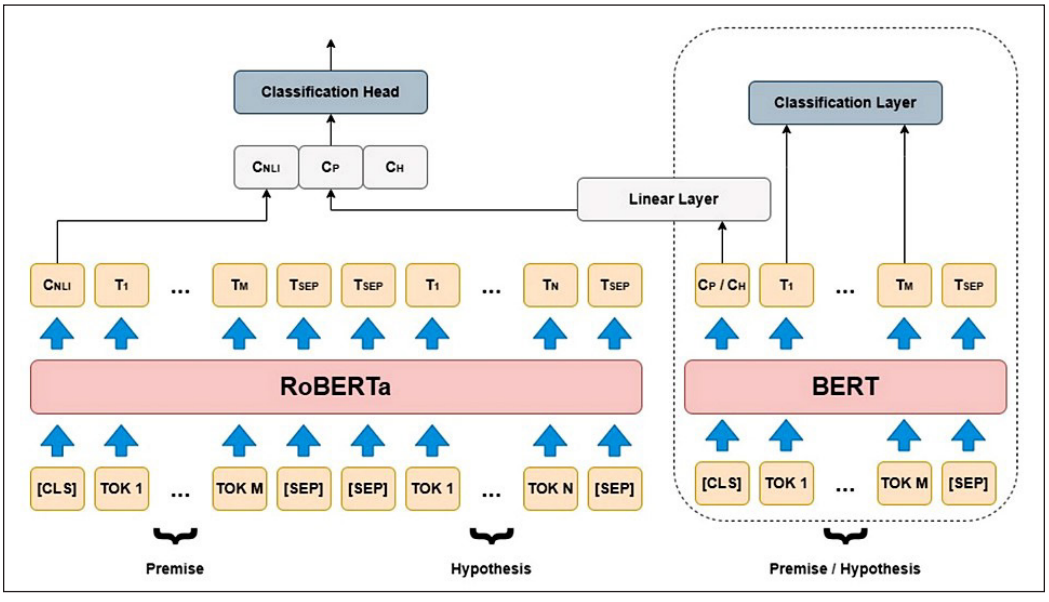

Figure 2. Architecture of robustly optimized BERT pretraining approach (RoBERTa) (Liu et al., 2019)
Note. BERT = Bidirectional encoder representations from transformers

DeBERTa advances BERT by introducing two key innovations: disentangled attention and an enhanced mask decoder. Unlike traditional models, it separates content and position embeddings, which allows it to compute attention scores independently and enhances contextual understanding. The enhanced mask decoder integrates absolute and contextual position embeddings, excelling in tasks requiring precise word order comprehension. These innovations allow DeBERTa to outperform BERT and RoBERTa across benchmarks and demonstrate the effectiveness of disentangled representations and advanced decoding strategies. Figure 3 illustrates DeBERTa's architecture.

Table 1 presents a comparative analysis of various large-scale pre-trained language models on the GLUE benchmark, a widely used suite of natural language understanding (NLU) tasks. As shown in Table 1, DeBERTa has the highest average score of 90.00,

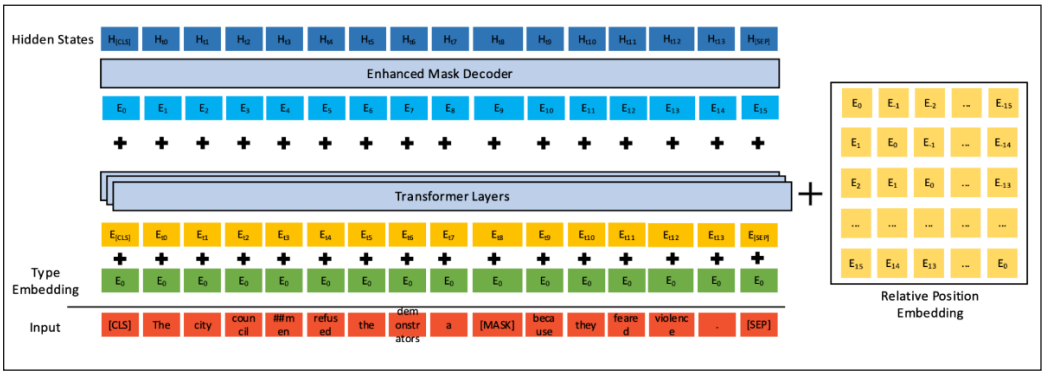

Figure 3. Architecture of decoding-enhanced BERT with disentangled attention (DeBERTa) (He et al., 2021)

Table 1
Comparison results on the General Language Understanding Evaluation (GLUE) development set

Model	CoLA Mcc	OOP Acc	MultiNLI-m/mm Acc	SST/2 Acc	STS-B Corr	QNLI Acc	RTE Acc	MRPC Acc	Avg.
BERT	60.6	91.3	86.6/-	93.2	90.0	92.3	70.4	88.0	84.05
RoBERT	68.0	92.2	90.2/90.2	96.4	92.4	93.9	86.6	90.9	88.82
XLNet	69.0	92.3	90.8/90.8	**97.0**	92.5	94.9	85.9	90.8	89.15
ELECTRA	69.1	**92.4**	90.9/-	96.9	92.6	95.0	88.0	90.8	89.46
DeBERTa	**70.5**	92.3	**91.1/91.1**	96.8	**92.8**	**95.3**	**88.3**	**91.9**	**90.00**

Note. Bolded figures represent the best results; CoLA = Corpus of Linguistic Acceptability; OOP = Out of position; MultiNLI-m/mm = Multi-Genre Natural Language Inference-matched/ mismatched; SST/2 = Stanford Sentiment Treebank (binary version); STS-B = Semantic textual similarity-Benchmark; QNLI = Question Natural Language Inference; RTE = Recognizing textual entailment; MRPC = Microsoft Research Paraphrase Corpus; Mcc = Matthews correlation coefficient; Acc = Accuracy; Corr = Correlation; Avg. = Average; BERT = Bidirectional encoder representations from transformers; RoBERTa = Robustly optimized BERT approach; XLNet = Generalized autoregressive pretraining for language understanding; ELECTRA = Efficiently learning an encoder that classifies token replacements accurately; DeBERTa = Decoding-enhanced BERT with disentangled attention

outperforming the other models. DeBERTa demonstrates its excellent ability to capture linguistic nuances and long-distance dependencies in text.

Background of Document Summary Matching Framework

A Siamese Network, introduced by Bromley et al. (1993), is designed to process and compare two input data points using two identical sub-networks with shared architecture, parameters, and weights. These sub-networks generate feature representations of inputs, mapping them into a latent space where similar points are closer together and dissimilar points are farther apart. After processing, a similarity function (e.g., cosine similarity or dot product) evaluates their relationship. The network uses contrastive loss to minimize distances between similar pairs and maximize distances for dissimilar ones (Koch et al., 2015).

Building upon the Siamese Network framework, Siamese-BERT integrates BERT's contextual embeddings (Devlin et al., 2018) to compute similarity scores between text pairs. Twin BERT models with shared parameters generate embeddings, which are compared using distance metrics to assess semantic similarity. This approach excels in tasks like sentence similarity, paraphrase identification, duplicate question detection, and natural language inference.

The effectiveness of this approach has been extensively validated. Reimers and Gurevych (2019) introduced Sentence-BERT (SBERT), a fine-tuned version of Siamese-BERT, achieving state-of-the-art results on benchmarks such as STS-B and Quora Question Pairs. Applications include identifying duplicate questions on platforms like Quora and Stack Overflow, improving query-document matching in information retrieval, and enhancing reasoning in inference tasks. Siamese-BERT consistently outperforms traditional models like InferSent and Universal Sentence Encoder, benefiting from BERT's rich embeddings and the Siamese network's shared weights, which ensure robust and consistent representations.

A crucial aspect of Siamese-BERT's similarity evaluation is the use of cosine similarity and dot product similarity, which help determine the degree of semantic relatedness between text pairs. Cosine similarity calculates the cosine of the angle between normalized vectors, capturing semantic closeness, while dot product similarity emphasizes vector magnitude, highlighting a sentence's contribution to overall semantic content. These similarity measures are particularly useful in extractive summarization, where they aid in selecting sentences that best preserve a document's core meaning, ensuring coherence and informativeness.

Despite its success, Siamese-BERT faces challenges like high computational costs for fine-tuning and sensitivity to training data quality. Future improvements may involve efficient training techniques like knowledge distillation and incorporating external

knowledge bases or multi-modal data. These advancements could further enhance its performance and applicability.

Research Gap

Despite significant advancements in extractive summarization techniques, critical challenges persist. One key issue is the limited ability of current models to capture nuanced semantic similarities between sentences. While BERT and its variants have improved contextual understanding, they often struggle with subtle semantic distinctions and relationships essential for high-quality summaries.

Most extractive models rely on surface-level features like sentence position or length to rank and select sentences. Although useful, these heuristics fail to capture deeper, more abstract relationships between sentences, resulting in less coherent and contextually accurate summaries. This limitation is especially pronounced in domains like legal or scientific texts, where precise terminology and logical structure are crucial.

Handling long documents remains another challenge. Models like BERTSUM and RoBERTa struggle to maintain coherence in lengthy texts due to fixed input lengths, leading to truncated or incomplete summaries. Although Longformer (Beltagy et al., 2020) provides some improvements for handling longer contexts, it still needs further refinement to preserve the overall semantic flow in complex documents.

Additionally, existing similarity measures, such as cosine similarity, used in frameworks like MatchSum (Zhong et al., 2020), inadequately capture word order and contextual significance. This leads to summaries that fail to fully reflect the intricate relationships in the source text, particularly when word meaning is context-dependent.

In conclusion, current models exhibit limitations in assessing semantic similarity. While methods such as cosine and dot-product similarity are practical, more advanced metrics are needed to preserve semantic integrity. Future efforts should focus on enhancing semantic understanding to produce more coherent, relevant, and semantically rich summaries.

METHODS

This chapter outlines the methodology for developing an extractive text summarization framework using deep learning and the DeBERTa model. By framing extractive summarization as a semantic text alignment problem, the framework improves summary selection through steps including data preprocessing, candidate pruning with SummaRuNNer, and summary generation using DeBERTa. A Siamese-DeBERTa architecture ensures strong semantic alignment between the final summary and the original document, enhancing coherence and relevance.

Formula and Task Definitions

Article (Document) definition: A document D is defined as a sequence of sentences:

$$D = \{s_1, s_2, \ldots, s_n\}$$

where s_i represents the i-th sentence in the document, and n is the total number of sentences.

Pruned document definition: To focus on the most relevant content, the document D undergoes a pruning process, resulting in a pruned document D'. The pruned document D' contains a subset of the original sentences from D:

$$D' = \{s_1, s_2, \ldots, s_m\}$$

where $m \leq n$, and m represents the indices of the selected sentences after pruning.

Candidate summary definition: A candidate summary C is a subset of sentences selected from the pruned document D', and the collection of all possible candidate summaries is denoted as D':

$$C \in D', C = \{s_1, s_2, \ldots, s_k\}$$

where k is the number of sentences in the candidate summary. The sentences $s_1, s_2, \ldots, s_k$ are selected in order they appear in D'.

For example, if D' contains five sentences $\{s_1, s_2, s_3, s_4, s_5\}$, a possible candidate summary C could be $\{s_1, s_3, s_5\}$, retaining important information while forming a concise representation of the document.

For the task definition, the input is a document D consisting of n sentences. The document D is pruned into a subset D', which is then used to generate candidate summaries. The function of the task is to select the candidate summary C from D' that maximizes the semantic similarity between the original document D and the candidate summary C. The output is the candidate summary $\hat{C}$ with the highest semantic similarity.

Let r_D be the embedding vector representing the entire document D, capturing its semantic information. Let r_C be the embedding vector representing a candidate summary C generated from D', capturing its semantic information. The goal is to find the candidate summary C that maximizes the dot product similarity between the embedding vectors r_D and r_C. Formally, the task can be expressed as:

$$\hat{C} = \arg\max_{C \in D}(r_D \cdot r_C)$$

where C is the set of all possible candidate summaries generated from the pruned document D', and $\cdot$ denotes the dot product operation, which measures the similarity between the embedding vectors of the original document D and the candidate summary C.

The task aims to generate an accurate and informative summary $\hat{C}$ with the highest semantic similarity to the original document D. The process begins by pruning D to produce D', a subset of sentences that focuses on the most relevant content. The pruned document D' is then embedded into a vector $r_{D'}$ using a pre-trained language model (e.g., BERT, RoBERTa, or DeBERTa) to capture its semantic information. Multiple candidate summaries C are generated from D', with each summary being a subset of sentences from D'. Each candidate summary C is embedded into a vector r_C using the same pre-trained model.

To measure how well a candidate summary represents the original document, the dot product similarity between r_D (the embedding of D) and r_C (the embedding of each candidate summary) is calculated. The candidate summary with the highest similarity score is selected as the final output. This approach ensures that the selected summary $\hat{C}$ maintains strong semantic alignment with D, resulting in a concise, accurate, and informative summary.

For example, suppose that the pretrained model generates embedding vectors for the source document and candidate summary as r_D = [0.5, 0.7, 0.2] and r_C = [0.4, 0.6, 0.3], respectively. According to the dot product calculation, we first compute the product of each corresponding component ($0.5 \times 0.4 = 0.20$, $0.7 \times 0.6 = 0.42$, $0.2 \times 0.3 = 0.06$) and then sum these values to obtain a dot product similarity of $0.20 + 0.42 + 0.06 = 0.68$. This numerical value intuitively reflects the degree of semantic matching between the candidate summary and the source document in the semantic space, thereby supporting the strategy of selecting the final summary based on the highest similarity score.

Proposed Framework

This section presents the methodology for our enhanced extractive summarization framework, which leverages deep learning techniques and the DeBERTa model with semantic similarity matching to improve text segment selection for summary generation.

Document input: The process begins with an input document containing multiple sentences, which are analyzed to extract the most representative content.

Candidate pruning: Sentences are scored using models like SummaRuNNer, which evaluates content richness, salience, novelty, and positional significance. The top-scoring sentences are selected as candidates for summary generation.

Candidate summary generation: Candidate summaries are created by combining 2–3 top-ranked sentences, ensuring coverage of the document's most important content.

Preprocessing: The document undergoes tokenization, sentence splitting, normalization (e.g., lowercasing and punctuation removal), and truncation to limit length and enhance processing efficiency.

Candidate summary embedding: The Siamese-DeBERTa model generates embedding vectors for both the document and candidate summaries, capturing their semantic relationships. Tied weights ensure consistent encoding, maintaining semantic coherence.

Similarity scoring: Dot product similarity between embedding vectors measures alignment between the document and candidate summaries, capturing both vector magnitude and direction for accurate semantic assessment.

Best candidate summary selection: The candidate summary with the highest similarity score is selected, ensuring alignment with the document's main content and semantic meaning.

Final summary selection: The best candidate summary is refined into the final summary.

This framework integrates deep learning techniques and semantic similarity matching to refine traditional extractive summarization. By framing summarization as a semantic text alignment task, it captures intricate relationships between the document and the summary, producing more coherent and semantically accurate results.

Dot product similarity complements this approach by effectively measuring the alignment between the semantic representations of the document and candidate summaries. Its consideration of vector magnitudes highlights the contribution of each sentence to the document's overall content. When combined with DeBERTa's robust contextual embeddings, dot product similarity enhances the precision of evaluations, yielding more accurate and insightful summaries.

In essence, DeBERTa encodes semantic content, while dot product similarity quantifies the closeness of these representations, ensuring that the final summaries maintain semantic integrity and contextual relevance. Figure 4 illustrates the overall model framework.

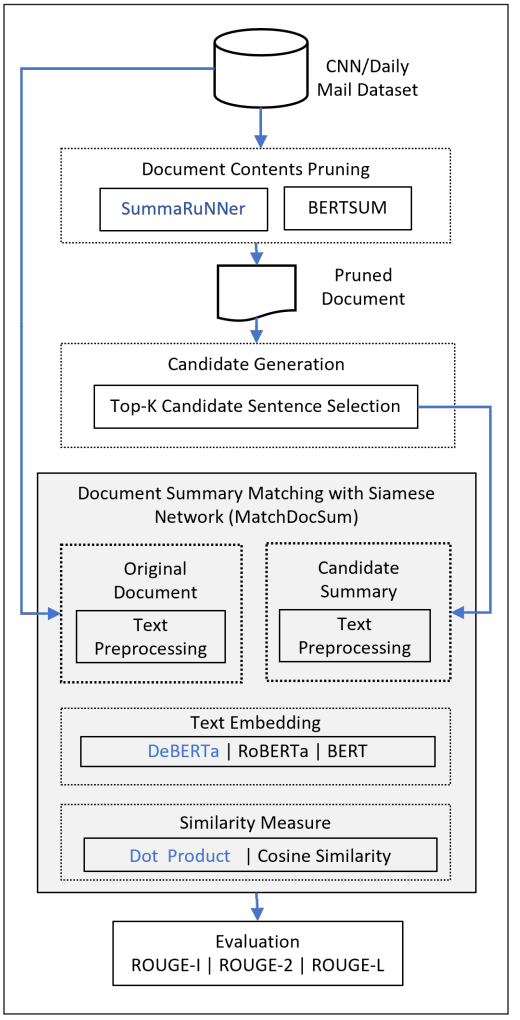

Figure 4. The proposed framework for extractive summarization

Note. CNN = Cable News Network; SummaRuNNer = A recurrent neural network (RNN) based sequence model for extractive summarization of documents; BERTSUM = Fine-tuning BERT for extractive summarization; DeBERTa = Decoding-enhanced BERT with disentangled attention; RoBERTa = Robustly optimized BERT approach; BERT = Bidirectional encoder representations from transformers; ROUGE = Recall-oriented Under for Gisting Evaluation

Datasets

The CNN/DailyMail dataset (Hermann et al., 2015), created for question-and-answer tasks, was later adapted by Nallapati et al. (2016) for text summarization research. Comprising over 300,000 news articles from CNN and Daily Mail, each paired with professionally crafted summaries, the dataset serves as a reliable benchmark for summarization studies. These high-quality summaries are concise yet capture the articles' essential points, enabling rigorous evaluation.

Renowned for its scale, diversity, and quality, the dataset covers topics such as politics, technology, sports, and entertainment. Its extensive training, validation, and testing examples facilitate robust model development and assessment. The CNN/DailyMail dataset remains a cornerstone resource for advancing text summarization and natural language processing research.

Document Contents Pruning

Candidate pruning is the initial step in a two-step summarization process that aims to reduce the document to a smaller subset of sentences, most likely to be included in the final summary. This step optimizes computational efficiency by focusing resources on the most relevant content. The section introduces the candidate-pruning strategy and compares two methods: BERTSUM (Liu & Lapata, 2019), the baseline approach, and SummaRuNNer (Nallapati et al., 2017). By evaluating both approaches, the framework seeks to enhance the effectiveness of sentence selection for summary generation.

A fine-tuned BERTSUM model is used to score sentences based on their probability $P(s_i)$ of being part of the summary, ensuring that the model adapts to the CNN/DailyMail dataset and captures its linguistic and structural patterns more effectively. A threshold τ is applied, either empirically determined or set to select a fixed percentage of top-scoring sentences, to filter out less relevant content. This candidate-pruning method reduces the complexity of the summarization task by narrowing the focus to the most important sentences, thereby optimizing subsequent steps in content generation.

A document $D = \{s_1, s_2, \ldots, s_n\}$ consists of n sentences, each assigned a score $P(s_i)$ that reflects its importance for the summary. These scores are generated using a BERTSUM model fine-tuned for extractive summarization, providing a contextual evaluation of each sentence's relevance. The sentence scoring is defined as:

$$P(s_i) = \text{BERTSUM}(s_i)$$

The probability $P(s_i)$ represents the likelihood that sentence s_i should be included in the summary. Using the BERTSUM model's contextual understanding, scores are computed for each sentence. Candidate sentences are selected by applying a threshold τ, where sentences

with $P(s_i) \geq \tau$ are chosen for further processing. The set of candidates can be represented by the following equation:

$$C = \{ s_i \mid P(s_i) \geq \tau \}$$

In this set C, only the sentences that meet or exceed the threshold t are included, reducing the number of sentences for further analysis. Alternatively, a fixed number of top-scoring sentences can be selected by ranking them in descending order of their scores and choosing the top k sentences:

$$C = \{ s_1, s_2, \ldots, s_k \}$$

where,

$$P(s_1) \geq P(s_2) \geq \ldots \geq P(s_k)$$

Here, C contains the top k sentences with the highest scores, ensuring that the most important sentences are selected.

Another candidate pruning strategy is SummaRuNNer, an RNN-based sequence classifier designed to overcome BERTSUM's limitations in capturing hierarchical structures and sequential dependencies. SummaRuNNer frames extractive summarization as a sequence classification task, dividing documents into sentences, which are further split into words and converted into embeddings.

The model consists of a two-layer bi-directional GRU-RNN. The first layer operates at the word level within each sentence, generating hidden state representations, while the second layer processes these representations at the sentence level to create sentence embeddings. A logistic regression layer assigns binary scores to sentences based on content richness, salience, novelty, and positional importance, determining their inclusion in the summary.

Sentences are ranked by their probabilities of inclusion, and top-ranked sentences are selected as candidates. This approach effectively captures both sentence-level and document-level dependencies, enhancing summary quality.

SummaRuNNer employs GRUs to process sentences, leveraging two gates: the update gate u_j, which retains information from the previous hidden state, and the reset gate r_j, which determines the extent of forgetting prior states (Cho et al., 2014). These gates are computed through specific mathematical operations within the GRU architecture as follows:

$$u_j = \sigma(W_{ux} x_j + W_{uh} h_{j-1} + b_u)$$

$$r_j = \sigma(W_{rx} x_j + W_{rh} h_{j-1} + b_r)$$

The input vector x_j represents the current word, h_{j-1} is the hidden state from the previous time step, W and b are weight matrices and bias vectors, and σ is the sigmoid activation function. The hidden state h_j is updated based on these parameters.

$$h'_j = \tanh(W_{hx}x_j + W_{hh}(r_j \odot h_{j-1}) + b_n)$$

$$h' = (1 - u_j) \odot h'_j + u_j \odot h_{j-1}$$

The final hidden state h_j is a combination of the new hidden state and the previous hidden state h_{j-1}, regulated by the update gate u_j (Cho et al., 2014). For each sentence, a representation is created by concatenating the hidden states from the forward and backward GRUs. The document representation d is then obtained through a non-linear transformation of the average of these concatenated hidden states:

$$d = \tanh\left(W_d\left(\frac{1}{N_d}\sum_{j=1}^{N_d}\left[h_j^f, h_j^b\right]\right) + b\right)$$

The number of sentences in the document is denoted by N_d, and the hidden states of the forward h_j^f and backward h_j^b RNNs are concatenated to form the sentence representation (Nallapati et al., 2017). The probability of a sentence being included in the summary is calculated using a logistic regression model:

$$(y_j = 1 | h_j, s_j, d) = \sigma(W_c h_j + h_j^T W_s d - h_j^T W_r tanh(s_j) + W_a p_a + W_r p_r + b)$$

The terms describe various factors influencing the importance of the j-th sentence in a document for summarization (Nallapati et al., 2017). $W_c h_j$ represents the information content of the j-th sentence. $h_j^T W_s d$ measures its salience concerning the entire document. $-h_j^T W_r \tanh(s_j)$ captures redundancy with respect to the current summary state. $W_a p_a + W_r p_r$ encodes the significance of the absolute and relative positions of the j-th sentence. b is a bias term. The dynamic summary representation s_j, up to the j-th sentence, is calculated as a weighted sum of the hidden states of all previous sentences:

$$s_j = \sum_{i=1}^{j-1} h_i P(y_i = 1 \mid h_i, s_i, d)$$

This formulation integrates sentence-level information, salience, redundancy, and positional importance for summary generation.

Two candidate summary pruning methods were introduced: BERTSUM and SummaRuNNer. BERTSUM leverages a fine-tuned BERT model to score sentences based on embeddings, selecting candidates using a threshold or top-ranked scores. SummaRuNNer utilizes a bidirectional GRU-RNN to generate sentence representations

and scores them through logistic regression, considering content, salience, novelty, and position. By selecting the top *k* sentences, SummaRuNNer effectively captures sequential dependencies and minimizes redundancy. These methods will be compared in experiments to assess their pruning performance.

Candidate Generation

After pruning, the top *k* sentences with the highest scores are selected based on their relevance to the document. Candidate summaries are then generated by forming combinations of 2 or 3 sentences from these top *k* sentences. For example, if $k = 5$, a total of $\binom{5}{2} + \binom{5}{3} = 10 + 10 = 20$ candidate summaries can be created.

Text Preprocessing

Text preprocessing is applied to candidate summaries and the source document to prepare them for analysis. The process includes tokenization, splitting text into smaller units like words or sub words; sentence splitting, converting text to lowercase, and truncation. The maximum sequence length is 512 tokens, including special tokens like [CLS] and [SEP].

Document Summary Matching with Siamese Network

Enhanced semantic matching redefines summarization as a semantic text-matching task, leveraging the DeBERTa model to generate richer embeddings that capture the semantic relationships between documents and candidate summaries. This ensures summaries are coherent and accurately reflect the core content of the original text.

Traditional methods, such as frequency-based approaches (Luhn, 1958) and graph-based methods (LexRank, TextRank), often overlook sentence-level semantic relationships, leading to disjointed, less readable summaries. To overcome this, a modified Siamese-DeBERTa framework is proposed that combines DeBERTa with dot-product similarity to enhance semantic alignment.

Siamese-DeBERTa effectively evaluates semantic similarity between the source document and candidate summaries, preserving the original document's essence while generating high-quality, coherent summaries. This approach improves summary accuracy and relevance, making it a valuable tool for diverse NLP tasks.

Inspired by Zhong et al. (2020), the MatchDocSum architecture aligns a document *D* with a candidate summary *C* in semantic space using DeBERTa. Its Siamese design employs tied weights, ensuring consistent embeddings for both inputs. Representations are derived from the [CLS] markers, and similarity between r_D (document embedding) and r_C (summary embedding) is measured using the dot product, preferred over cosine similarity for capturing vector magnitude and semantic richness.

The MatchDocSum framework utilizes DeBERTa's advanced attention mechanism to capture nuanced semantic relationships, refining text embedding and similarity matching. Pre-trained on a large-scale corpus, DeBERTa produces accurate embeddings, making it highly effective for extractive summarization. Comparative studies confirm that DeBERTa surpasses BERT and RoBERTa in preserving semantic integrity while selecting coherent and relevant sentences.

The framework integrates document pruning, embedding, and matching to deliver high-quality summaries aligned with the source document's content, ensuring both precision and reliability in summary evaluation.

Similarity analysis plays a pivotal role in extractive summarization by ranking candidate summaries based on their semantic similarity to the source document, maintaining coherence and relevance. Common metrics such as cosine and dot product similarity measure semantic closeness, each offering distinct advantages across different applications.

In MatchDocSum, dot-product similarity is selected as the primary matching method due to its ability to capture vector magnitude, enabling more precise semantic evaluation. Research has shown that cosine similarity does not fully capture the relationships in the source text and is particularly limited in representing semantic hierarchy and information weight. By incorporating dot-product similarity, summarization accuracy and quality are enhanced, ensuring that the generated summaries better reflect the semantic richness of the source document. Additionally, cosine similarity is introduced for comparative experiments to further assess the impact of different similarity metrics on summarization effectiveness. Figure 5 illustrates the overall matching process.

A good summary should achieve a higher similarity score than a poor one. To optimize Siamese-DeBERTa, a two-part loss function is employed. The first component, a margin-

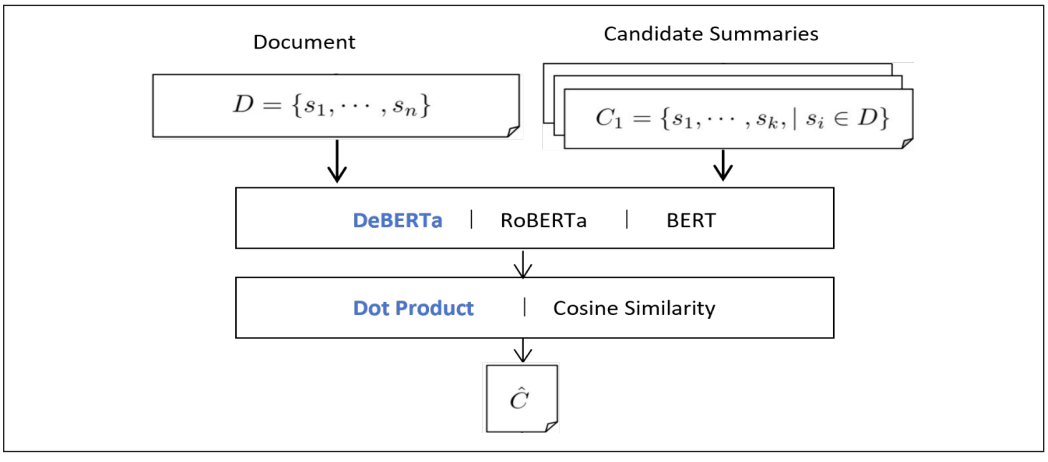

Figure 5. MatchDocSum architecture
Note. DeBERTa = Decoding-enhanced BERT with disentangled attention; RoBERTa = Robustly optimized BERT approach; BERT = Bidirectional encoder representations from transformers

based triplet loss, minimizes the semantic gap between generated and authentic summaries, ensuring fidelity to the source document's core meaning. The second, a ranking gap loss, reinforces score differentiation by minimizing similarity gaps between high- and low-ranking candidates, ensuring high-quality summaries consistently score higher. The loss function is defined as follows:

$$L = \max\left(0, S_C - S_{gt} + \gamma_1\right) + \max\left(0, S_{C_j} - S_{C_i} + \gamma_2 \; (i < j)\right)$$

S_C and S_{gt} represent the similarity values between the candidate summaries, gold summaries, and the source document D, respectively. The i and j denote the order of the candidate summaries, while γ_1 and γ_2 are hyperparameters, with γ_2 being orthogonal to $(i-j)$. By employing this dual loss function, the model considers both semantic proximity to the source document and the ranking gap between candidate summaries, enhancing its ability to produce high-quality, semantically precise summaries.

Evaluation Methods

To evaluate summary quality, ROUGE metrics were employed. Although ROUGE may be limited by abstract diversity constraints, it is widely used in abstracting tasks as a standard assessment metric, is easy to compare, is computationally efficient, and higher scores usually reflect good content coverage. ROUGE-1, ROUGE-2, and ROUGE-L metrics are used to evaluate the quality of generated summaries by comparing them to reference summaries.

ROUGE-1 focuses on the overlap of individual words (unigrams) between the generated summary and the reference summary. It provides a basic measure of how well the hypothesis captures the content of the reference by evaluating word-level similarity.

ROUGE-2 extends this analysis to sequences of words (bigrams). By capturing pairs of consecutive words, it evaluates how well the generated summary preserves the word order and flow of the reference summary. ROUGE-1 and ROUGE-2 calculate the recall of word sequences (unigrams and bigrams), enabling a granular evaluation of text summarization or translation quality.

ROUGE-L measures the longest common subsequence (LCS) between the hypothesis and the reference, allowing for non-contiguous matches. It evaluates similarity at a structural level, combining recall and precision of the LCS to emphasize recall, in alignment with ROUGE's primary objective of measuring content overlap.

RESULTS AND DISCUSSION

Baseline Comparison

This section outlines the baseline models used for comparison with the proposed MatchDocSum model on the CNN/DailyMail dataset:

- LEAD (Nallapati et al., 2016): Selects the first three sentences of each document as the summary, leveraging the inverted pyramid structure of news articles.
- ORACLE (Hirao et al., 2017): Chooses sentences that maximize ROUGE scores relative to reference summaries, representing the upper bound of extractive summarization.
- BERTSUM (Liu & Lapata, 2019): Extends BERT for extractive summarization by introducing interval segment embeddings to capture sentence relationships better.
- MatchSum (Zhong et al., 2020): Frames summarization as a semantic text matching problem, using BERTSUM for sentence encoding and cosine similarity for matching.
- SummaRuNNer (Nallapati et al., 2017): An RNN-based model that selects sentences based on relevance, novelty, and position.

The performance of these models is evaluated using ROUGE metrics for comparison with MatchDocSum.

Experiments on Candidate Pruning

In this experiment, SummaRuNNer was used for candidate pruning in the MatchDocSum model. The two-layer bi-directional GRU-RNN evaluated sentences based on content richness, relevance, novelty, and positional significance. Cosine similarity was employed to select the best sentences, effectively reducing redundancy and improving summary diversity compared to the BERTSUM baseline. The results are summarized in Table 2.

SummaRuNNer combined with RoBERTa achieved the highest ROUGE scores in Table 2: ROUGE-1 of 43.10, ROUGE-2 of 20.10, and ROUGE-L of 40.00, outperforming BERTSUM + BERT, which had the lowest scores (ROUGE-1 of 41.85, ROUGE-2 of 19.34, and ROUGE-L of 39.90). The improvement is attributed to SummaRuNNer's ability to capture sequential dependencies and evaluate multiple features, resulting in summaries that better preserve the document's structure and essential information.

Table 2
Performance comparison of different pruning methods and models

Pruning method	Model	ROUGE-1	ROUGE-2	ROUGE-L
SummaRuNNer	BERT	42.50	19.80	39.50
	RoBERTa	43.10	20.10	40.00
BERTSUM	BERT	41.85	19.34	39.90
	RoBERTa	42.00	19.50	40.10

Note. ROUGE = Recall-oriented Under for Gisting Evaluation; SummaRuNNer = A recurrent neural network (RNN) based sequence model for extractive summarization of documents; BERTSUM = Fine-tuning BERT for extractive summarization; RoBERTa = Robustly optimized BERT approach; BERT = Bidirectional encoder representations from transformers

Experiments on Text Embedding Using DeBERTa

This experiment evaluates the effectiveness of text embeddings generated by DeBERTa, BERT, and RoBERTa within a Siamese network, applied to documents pruned using SummaRuNNer and BERTSUM methods. The focus is on assessing the impact of these embeddings on cosine similarity and their ability to preserve the semantic content of the original text. ROUGE-1, ROUGE-2, and ROUGE-L scores are calculated for each combination of pruning method and embedding model to evaluate summary quality.

Results in Table 3 show that DeBERTa consistently outperformed BERT and RoBERTa across all pruning methods, achieving the highest ROUGE scores (43.20) with SummaRuNNer-pruned documents. This indicates DeBERTa's superior ability to maintain contextual integrity and semantic richness in summaries. While BERT and RoBERTa performed well, their scores were slightly lower, emphasizing DeBERTa's advantage for embedding in text summarization tasks, especially when combined with effective pruning methods like SummaRuNNer.

Table 3
Comparison of pruning methods and embedding models on ROUGE metrics

Pruning method	Embedding model	ROUGE-1	ROUGE-2	ROUGE-L
SummaRuNNer	DeBERTa	43.20	20.30	40.50
	BERT	42.50	19.80	39.90
	RoBERTa	43.10	20.10	40.00
BERTSUM	DeBERTa	42.00	19.50	39.70
	BERT	41.85	19.34	39.50
	RoBERTa	42.10	19.60	39.80

Note. ROUGE = Recall-oriented Under for Gisting Evaluation; SummaRuNNer = A recurrent neural network (RNN) based sequence model for extractive summarization; BERTSUM = Fine-tuning BERT for extractive summarization; DeBERTa = Decoding-enhanced BERT with disentangled attention; RoBERTa = Robustly optimized BERT approach; BERT = Bidirectional encoder representations from transformers

Experiment on Embedding Similarity (Dot Product)

This experiment compares dot product and cosine similarity as semantic similarity measures within a Siamese-DeBERTa architecture. Documents pruned by SummaRuNNer and BERTSUM were embedded using Siamese-DeBERTa, and the similarity between embeddings was calculated using both metrics.

Table 4 shows that the dot product consistently outperformed cosine similarity across all ROUGE metrics. For SummaRuNNer, dot product achieved ROUGE-1, ROUGE-2, and ROUGE-L scores of 43.50, 20.45, and 40.75, surpassing cosine similarity's 43.20, 20.30, and 40.50. Similarly, for BERTSUM, dot product scored 42.60, 19.90, and 39.85, exceeding cosine similarity's 42.30, 19.70, and 39.60.

Table 4
Comparison of dot product and cosine similarity for DeBERTa embeddings under different pruning methods

Pruning method	Similarity measure	ROUGE-1	ROUGE-2	ROUGE-L
SummaRuNNer	Dot product	43.50	20.45	40.75
	Cosine similarity	43.20	20.30	40.50
BERTSUM	Dot product	42.60	19.90	39.85
	Cosine similarity	42.30	19.70	39.60

Note. DeBERTa = Decoding-enhanced BERT with disentangled attention; ROUGE = Recall-oriented Under for Gisting Evaluation; SummaRuNNer = A recurrent neural network (RNN) based sequence model for extractive summarization; BERTSUM = Fine-tuning BERT for extractive summarization

The dot product's ability to capture both vector magnitude and direction enhances semantic alignment, yielding more coherent and accurate summaries. These results underscore its superiority over cosine similarity in extractive summarization tasks using DeBERTa embeddings.

Experimental Results Compared to Baseline

Table 5 compares the proposed MatchDocSum framework with baseline methods. LEAD and ORACLE are widely used baselines, with LEAD selecting the first three sentences and ORACLE maximizing ROUGE scores using abstractive summarization principles. ORACLE generally performs better due to its abstraction-based approach. BERTSUM, the primary baseline in this experiment, demonstrated solid performance but suffered from redundancy issues, consistent with previous studies on the CNN/DailyMail dataset.

Re-implementations of MatchSum with BERTSUM using BERT-base, RoBERTa-base, and DeBERTa-base showed incremental gains, with DeBERTa-base achieving the best results by effectively capturing semantic information and reducing redundancy. In comparison, SummaRuNNer with DeBERTa-base outperformed BERTSUM in content richness, diversity, and redundancy reduction, resulting in higher ROUGE scores.

The MatchDocSum framework, which integrates Siamese-DeBERTa with SummaRuNNer for pruning, further improved semantic preservation and reduced redundancy. Although it did not surpass all baselines in every metric, it demonstrated competitive performance. For instance, MatchDocSum achieved a higher ROUGE-L score (40.75) than BERTSUM (DeBERTa-base) at 39.70, though its ROUGE-1 and ROUGE-2 scores (43.50 and 20.45) were slightly lower than BERTSUM's (42.00 and 19.50). These results reflect MatchDocSum's emphasis on semantic coherence over direct sentence extraction.

While MatchDocSum does not fully outperform ORACLE and MatchSum, its strength lies in semantic alignment optimization. ORACLE sets an upper bound on the theoretical optimum by directly selecting sentences that maximize the ROUGE score. MatchSum

Table 5
Comparison of our framework with various baselines on ROUGE metrics

Model	ROUGE-1	ROUGE-2	ROUGE-L
LEAD	40.43	17.62	36.67
ORACLE	52.59	31.23	48.87
BertSumExt(large) (Liu & Lapata, 2019)	43.85	20.34	39.90
MatchSum (BERT-base) (Zhong et al., 2020)	44.22	20.62	40.38
MatchSum (RoBERTa-base) (Zhong et al., 2020)	44.41	20.86	40.55
SummaRuNNer (DeBERTa-base)	43.20	20.30	40.50
BERTSUM (DeBERTa-base)	42.00	19.50	39.70
MatchDocSum (DeBERTa-base+summarunner_prun)	43.50	20.45	40.75
MatchDocSum (Deberta-base+bertsum_prun)	42.60	19.90	39.85

Note. ROUGE = Recall-oriented Under for Gisting Evaluation; LEAD = Lead baseline; ORACLE = Oracle extractive upper bound; BertSumExt(large) = BERT-based extractive summarization (large version); MatchSum (BERT-base) = Extractive summarization as sentence ranking with BERT-base encoder; MatchSum (RoBERTa-base) = Extractive summarization as sentence ranking with RoBERTa-base encoder; SummaRuNNer (DeBERTa-base) = A recurrent neural network based sequence model for extractive summarization (with DeBERTa-base as encoder); BERTSUM (DeBERTa-base) = Fine-tuned DeBERTa-base for extractive summarization; MatchDocSum (DeBERTa-base+summarunner_prun) = Document matching for summarization (with DeBERTa-base encoder + SummaRuNNer-based pruning); MatchDocSum (DeBERTa-base+bertsum_prun) = Document matching for summarization (with DeBERTa-base encoder + BERTSUM-based pruning)

relies on BERTSUM for sentence-level matching to ensure optimal alignment of candidate summaries to the original document. In contrast, MatchDocSum employs DeBERTa for document encoding, which is better able to capture long-distance dependencies and improves the contextual understanding of summaries by matching the semantic representations of candidate summaries with the source document. Although the pruning step may result in a slightly lower ROUGE score than ORACLE, MatchDocSum performs better in terms of semantic consistency and contextual coherence, provides an extractive summarization method that better meets the actual semantic matching requirements, and brings new optimization ideas to the domain.

Lastly, a noticeable result from the performance between different ROUGE metrics used shows that our proposed models are consistent with the baselines. The scores show that, on average, the models can capture a reasonable portion of the words (when order or context is not considered), but struggle to capture the sequential relationships between words. The overall drop from ROUGE-1 to ROUGE-2 shows the limitation of the models in capturing more complex linguistic structures. Compared to ROUGE-1 and ROUGE-2, ROUGE-L's scores are between the formers, which still show it falls short in fully preserving the structure and the flow of the summary. Despite being able to capture individual words (ROUGE-1), it struggles with capturing more complex structures like bigrams (ROUGE-2)

and maintaining the exact sequence of words (ROUGE-L). Since our approach belongs to the category of extractive summarization, ROUGE measures remain a robust performance metric to capture the basic structure of the summary.

CONCLUSION

The proposed research presents an improvised extractive summarization framework, redefining the task as a semantic text matching problem. It incorporates SummaRuNNer for document pruning to reduce redundancy and improve summary diversity, DeBERTa for generating rich semantic embeddings, and dot product similarity for enhanced semantic alignment. Evaluated on the CNN/DailyMail dataset, the framework demonstrates effectiveness, achieving better semantic preservation and contextual accuracy in summaries, though with slightly lower ROUGE scores than ORACLE due to pruning.

This study holds both theoretical and practical significance. Theoretically, it deepens understanding of how pre-trained models like DeBERTa capture semantic relationships between documents and summaries. The integration of dot product similarity further explores how similarity metrics influence summary quality. Practically, the framework has applications in fields such as news aggregation, information retrieval, and automated report generation, where concise, semantically rich summaries are crucial.

Despite its strengths, the research faces limitations. DeBERTa struggles with long-range dependencies in lengthy texts, potentially missing dispersed key relationships. Its high computational cost limits accessibility for resource-constrained researchers. Evaluation on the CNN/DailyMail dataset raises concerns about generalizability to domains like legal or scientific texts, which have distinct structures. Candidate pruning methods like SummaRuNNer rely on surface-level features such as sentence position and length, which may not accurately reflect sentence relevance. Additionally, ROUGE, as the primary evaluation metric, emphasizes lexical overlap but neglects readability, coherence, and informativeness.

Future improvements could address these limitations by enhancing the framework's capacity to handle long-range dependencies using architectures like memory-augmented networks. Techniques such as model compression or knowledge distillation could lower computational costs, enabling real-time applications. Expanding the framework to diverse domains through domain adaptation and fine-tuning can improve generalizability. Advanced candidate pruning methods, such as reinforcement learning, could reduce biases from surface-level features. Finally, combining ROUGE with semantic metrics like BERTScore could offer a more comprehensive evaluation of summary readability, coherence, and informativeness.

ACKNOWLEDGEMENTS

The authors would like to thank Universiti Sains Malaysia, School of Computer Sciences, for their support in this research work.

REFERENCES

Alyguliyev, R. M. (2009). The two-stage unsupervised approach to multidocument summarization. *Automatic Control and Computer Sciences*, *43*, 276–284. https://doi.org/10.3103/S0146411609050083

Bae, S., Kim, T., Kim, J., & Lee, S.-G. (2019). Summary level training of sentence rewriting for abstractive summarization. In L. Wang, J. C. K. Cheung, G. Carenini, & F. Liu (Eds.), *Proceedings of the 2nd Workshop on New Frontiers in Summarization* (pp. 10–20). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5402

Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The long-document transformer*. arXiv. https://doi.org/10.48550/arXiv.2004.05150

Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., Lecun, Y., Moore, C., Säckinger, E., & Shah, R. (1993). Signature verification using a "Siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, *7*(4), 669–688. https://doi.org/10.1142/S0218001493000339

Chen, Y.-C., & Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In I. Gurevych & Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics* (pp. 675–686). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1063

Cho, K., van Merriënboer, B., Gulcehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1179

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of deep bidirectional transformers for language understanding*. arXiv. https://doi.org/10.48550/arXiv.1810.04805

Dutulescu, A.-N., Dascalu, M., & Ruseti, S. (2022). Unsupervised extractive summarization with BERT. In *24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (pp. 158–164). IEEE. https://doi.org/10.1109/SYNASC57785.2022.00032

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479. https://doi.org/10.1613/jair.1523

Galanis, D., Lampouras, G., & Androutsopoulos, I. (2012). Extractive multi-document summarization with integer linear programming and support vector regression. In M. Kay & C. Boitet (Eds.), *Proceedings of COLING 2012* (pp. 911–926). The COLING 2012 Organizing Committee.

He, P., Liu, X., Gao, J., & Chen, W. (2020). *DeBERTa: Decoding-enhanced BERT with disentangled attention*. arXiv. https://doi.org/10.48550/arXiv.2006.03654

He, P., Liu, X., Gao, J., & Chen, W. (2021). *Microsoft DeBERTa surpasses human performance on the SuperGLUE benchmark*. Microsoft. https://www.microsoft.com/en-us/research/blog/microsoft-deberta-surpasses-human-performance-on-the-superglue-benchmark/

Hermann, K. M., Kočiský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., & Blunsom, P. (2015). *Teaching machines to read and comprehend*. arXiv. https://doi.org/10.48550/arXiv.1506.03340

Hirao, T., Nishino, M., Suzuki, J., & Nagata, M. (2017). Enumeration of extractive ORACLE summaries. In M. Lapata, P. Blunsom, & A. Koller (Eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (Vol. 1, pp. 386–396). Association for Computational Linguistics. https://aclanthology.org/E17-1037.pdf

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). *Siamese neural networks for one-shot image recognition*. https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

Liu, Y. (2019). *Fine-tune BERT for extractive summarization*. arXiv. https://arxiv.org/abs/1903.10318

Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 3730–3740). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1387

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT pretraining approach*. arXiv. https://doi.org/10.48550/arXiv.1907.11692

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165. https://doi.org/10.1147/rd.22.0159

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing* (pp. 404–411). Association for Computational Linguistics.

Nallapati, R., Zhai, F., & Zhou, B. (2017). SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1), 3075–3081. https://doi.org/10.1609/aaai.v31i1.10958

Nallapati, R., Zhou, B., dos Santos, C., Gülçehre, Ç., & Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In S. Riezler & Y. Goldberg (Eds.), *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 280–290). Association for Computational Linguistics. https://doi.org/10.18653/v1/K16-1028

Pradhan, S., Moschitti, A., Xue, N., Ng, H. T., Björkelund, A., Uryupina, O., Zhang, Y., & Zhong, Z. (2013). Towards robust linguistic analysis using OntoNotes. In J. Hockenmaier & S. Riedel (Eds.), *Proceedings of the Seventeenth Conference on Computational Natural Language Learning* (pp. 143–152). Association for Computational Linguistics.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(1), 5485–5551.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 3982–3992). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need*. arXiv. https://doi.org/10.48550/arXiv.1706.03762

Yao, K., Zhang, L., Luo, T., & Wu, Y. (2018). Deep reinforcement learning for extractive document summarization. *Neurocomputing*, *284*, 52–62. https://doi.org/10.1016/j.neucom.2018.01.020

Zhang, H., Cai, J., Xu, J., & Wang, J. (2019). Pretraining-based natural language generation for text summarization. In M. Bansal & A. Villavicencio (Eds.), *Proceedings of the 23rd Conference on Computational Natural Language Learning* (pp. 789–797). Association for Computational Linguistics. https://doi.org/10.18653/v1/K19-1074

Zhang, X., Wei, F., & Zhou, M. (2019). HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In A. Korhonen, D. Traum, & L. Márquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 5059–5069). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1499

Zhong, M., Liu, P., Chen, Y., Wang, D., Qiu, X., & Huang, X. (2020). Extractive summarization as text matching. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 6197–6208). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.552