Reaching for upper bound ROUGE score of extractive summarization methods

Iskander Akhmetov [1, 2], **Rustam Mussabayev** [2], **Alexander Gelbukh** [Corresp. 3]

[1] Kazakh-British Technical University, Almaty, Almaty, Kazakhstan
[2] Natural Language Processing Laboratory, Institute of Information and Computational Technologies, Almaty, Almaty, Kazakhstan
[3] Instituto Politecnico Nacional, Mexico, Mexico

Corresponding Author: Alexander Gelbukh
Email address: gelbukh@gelbukh.com

The Extractive Text Summarization (ETS) method for finding the salient information from a text automatically uses the exact sentences from the source text. In this paper, we answer the question of what quality of a summary we can achieve with ETS methods? To maximize the ROUGE-1 score, we used five approaches: (1) adapted Reduced Variable Neighborhood Search (RVNS), (2) Greedy algorithm, (3) VNS initialized by Greedy algorithm results, (4) Genetic algorithm, and (5) Genetic algorithm initialized by the Greedy algorithm results. Furthermore, we ran experiments on articles from the arXive dataset. As a result, we found 0.59 and 0.25 scores for ROUGE-1 and ROUGE-2, respectively achievable by the approach, where the Genetic algorithm initialized by the Greedy algorithm results, which happens to yield the best results out of the tested approaches. Moreover, those scores appear to be higher than scores obtained by the current state-of-the-art text summarization models: the best score in the literature for ROUGE-1 on the same data set is 0.46. Therefore, we have room for the development of ETS methods, which are now undeservedly forgotten.

Reaching for Upper Bound ROUGE Score of Extractive Summarization Methods

Iskander Akhmetov[1,2], Rustam Mussabayev[1], and Alexander Gelbukh[3]

[1]**Institute of Information and Computational Technologies, Almaty, Kazakhstan**
[2]**Kazakh-British Technical University, Almaty, Kazakhstan**
[3]**Instituto Politécnico Nacional, Mexico city, Mexico**

Corresponding author:
Alexander Gelbukh[3]

Email address: gelbukh@gelbukh.com

ABSTRACT

The *Extractive Text Summarization* (ETS) method for finding the salient information from a text automatically uses the exact sentences from the source text. In this paper, we answer the question of what quality of a summary we can achieve with ETS methods. To maximize the ROUGE-1 score, we used five approaches: (1) adapted Reduced Variable Neighborhood Search (RVNS), (2) Greedy algorithm, (3) VNS initialized by Greedy algorithm results, (4) Genetic algorithm, and (5) Genetic algorithm initialized by the Greedy algorithm results. Furthermore, we ran experiments on articles from the arXive dataset. As a result, we found 0.59 and 0.25 scores for ROUGE-1 and ROUGE-2, respectively achievable by the approach, where *the Genetic algorithm initialized by the Greedy algorithm results*, which happens to yield the best results out of the tested approaches. Moreover, those scores appear to be higher than scores obtained by the current state-of-the-art text summarization models: the best score in the literature for ROUGE-1 on the same data set is 0.48. Therefore, we have room for the ETS methods development, which are now undeservedly forgotten.

1 INTRODUCTION

Automatic Text Summarization (ATS) is a process of generating a relatively small-sized text out of a bigger one while preserving all the critical information. The research on the problem started in 1958 (Luhn, 1958) and saw a huge development in terms of methods, approaches, and applications. The most numerous advancements in the ATS happened after 2003 (Parker et al., 2011) when the large data sets and powerful computational resources became available to researchers.

Generally, ATS methods can be classified on the type of Input (Multi-/Single-document), Output (Extractive/Abstractive) and Content (Informative/Indicative); see Figure 1.

The methods shown in Figure 1 described as follows:

1. *Input*
   (a) *Single-document* summarization is when we summarize one single document, using only the textual information within and no additional sources.
   (b) *Multi-document* summarization produces a summary of a set of documents related to a common subject but varying by the time of appearance, size, and source. Applications of the method cover many areas, including literature review in scientific research, business intelligence, government reports, and legal document processing.
2. *Output*
   (a) *Extractive* summary contains only original sentences from the source text, without any change or recombination. Such summaries often lack cohesion between consequent sentences as they are extracted from different parts of the text, taking into account solely the statistical significance of the words they contain.

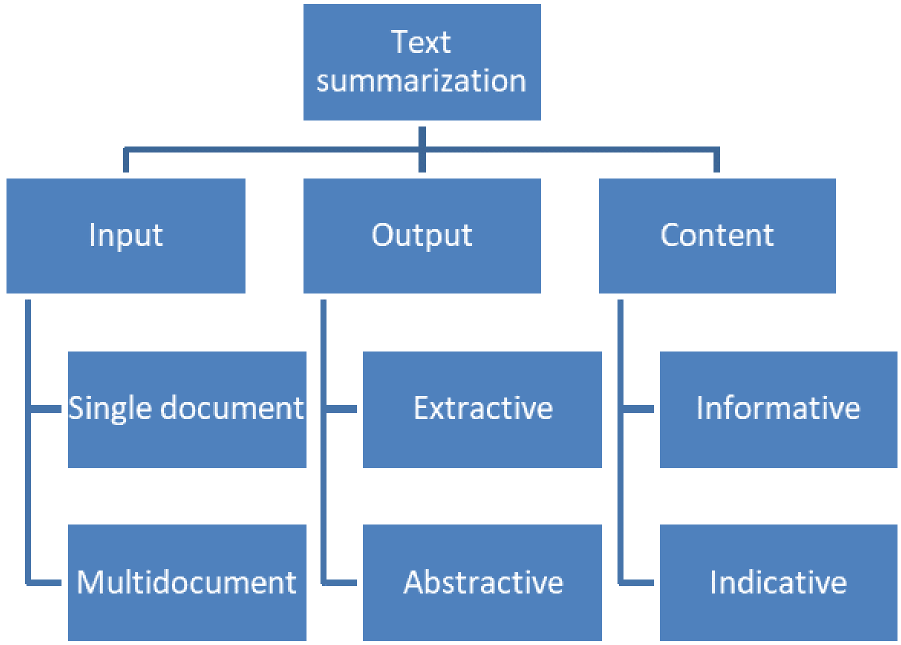

Figure 1. Classification Automatic Text Summarization methods (Radev et al., 2002; Abualigah et al., 2020).

   (b) *Abstractive* summary is a completely new text generated relying on the information in the source text put through the prism of the opinion and understanding of the information consumed by the reporter. The method requires more sophisticated Natural Language Generation (NLG) models and approaches than Extractive methods.

3. *Content*

   (a) *Informative* summaries contain all the critical information from the source text and avoid redundancy. Generally, it is achievable at the 20% compression rate (Kupiec and Pedersen, 1995).

   (b) *Indicative* summaries aim at teasing the reader to consume the whole article to stimulate the article purchase or spend time on a long read.

Thus, Extractive Summarization methods “extract” sentences or other text items, such as words or paragraphs, from the original text to make summaries without making up even a single word. The advantage of these methods is that they are always factually correct according to the processed text. On the other hand, abstractive summarization methods often give related information from sources other than the original text.

The challenging question we want to answer in this paper is whether we have room for developing *Extractive Text Summarization* (ETS) methods. Or are they outdated and have to be replaced by *Abstractive Text Summarization* methods? Additionally, we question what maximum summary quality we can achieve using ETS methods.

In this paper, to assess the quality of generated summaries, we use ROUGE-1 and ROUGE-2 scoring, which are the quantitative evaluations of the number of words shared by a candidate summary with the reference (or “golden”) summary, divided by the number of words in these summaries, and the harmonic mean between these two numbers; see section 3.3.

Therefore, we define the ATS optimization problem as finding the ultimate set of sentences for the summary to yield the maximum ROUGE score possible. However, the problem belongs to NP-full class of problems, and solving it with the Brute Force algorithm would not be feasible, and we need to find a better way by applying a heuristic algorithm.

For this purpose we compare the use of the *Variable Neighborhood Search* (VNS) (Hansen and Mladenović, 2018; Hansen et al., 2010) method; see section 3.2.1, with a *greedy algorithm*, which extracts sentences from the source text containing the maximum number of words from the "golden" summary; see section 4.2, and finally, with the *genetic algorithm*.

We also run experiments with Variable Neighborhood Search (VNS) and Genetic algorithms initialized by the Greedy solution; see section 4.3 and 4.5.

The contribution of our research to the scientific knowledge is in 1) discovery of the ETS methods ROUGE score upper bound, 2) a dataset of scientific texts with high-ROUGE score extractive summaries produced by the algorithms discussed in this paper, and valuable text statistics[1], 3) code to replicate the implemented research[2].

At the same time, we raise a discussion on several important topics for further research in section 6.

In section 2, we gave a short overview of the research and developments made in the area of ATS. Then, in section 3 we describe the data used for our experiments and the methods and the Experiment setup is described in section 4. In section 5, we show the obtained results , followed by discussion of the issues and thoughts we found during our research in section 6, and concluding the work in section 7 with setting out prospects for future work.

2 RELATED WORK

Most Automatic Text Summarization (ATS) research papers are devoted to summarization methods. However, few papers research the upper bound of quality achievable by the summaries generated.

Ceylan et al. (2010), working on the texts in the domains of scientific, legal, and news texts, used an exhaustive search strategy to explore the summary space of each domain and found respective Probability Density Function (PDF) of the ROUGE score distributions. Then using the obtained PDF function, they ranked the summarization systems that existed for the time by percentiles.

Further, Verma and Lee (2017) explored the upper bound limits for Single and Multi-Document summary quality on DUC01/02 datasets. However, they made it only for the recall part of the ROUGE scoring metrics, stating that the upper limit for the recall is achieved by using the whole source text as a summary leading to that metric going up as far as 90-100%. Nevertheless, using the entire text as a summary is not what we are looking for in the ATS task.

Abstractive summaries composed by humans using their own words leave little chance for Extractive Summarization to get a high ROUGE score. W. M. Wang et al. propose nine heuristic methods for generating high-quality sentence-based summaries for long texts from five different corpora. They demonstrated that the results achieved by their heuristics methods are close to those of Exhaustive (or Brute Force) algorithms but work much faster (Wang et al., 2017).

In this work, we used the VNS heuristic algorithm (Hansen and Mladenović, 2001) for finding the set of sentences in the original text to assemble the best ROUGE score summary. VNS iteratively changes the initial random solution and updates the rate of change if no improvement occurs, fixing the best result.

We also applied a Greedy algorithm (Black, 2005), widely applied in different text summarization approaches:

- Maximal Marginal Relevance (MMR) Carbonell and Goldstein (1998) struggles to increase relevance while reducing redundancy of the selected sentences.
- Integer Linear Programming (ILP) Gillick et al. (2009), identifying the key concepts in the summarized text and then greedily selecting the sentences covering those concepts at maximum.
- Submodular selection: optimized semantic graph submodule extraction, built on the text being summarized Lin et al. (2009).

Nevertheless, in this paper, we use the Greedy algorithm to find the upper bound of the ROUGE score achievable by the Extractive Summarization models.

We applied Genetic Algorithm (Mitchell, 1998), a nature-inspired technique used in many optimization problems applying the concepts of mutation and crossover. The algorithm is popular in the summarization models, both Single and Multi-document methods:

[1] https://data.mendeley.com/datasets/nvsxfcbzdk/1

[2] https://github.com/iskander-akhmetov/Reaching-for-Upper-Bound-ROUGE-Score-of-Extractive-Summarization-Methods

- Genetic Algorithm application to maximize the fitness function, which mathematically expresses such summary properties as topic relation, readability, and cohesion (Chatterjee et al., 2012) in documents represented as a weighted Directed Acyclic Graphs (DAG) Li and McCallum (2006) applying the popular Graph Methods in NLP Mihalcea and Radev (2011)

- The strength of Genetic Algorithms was demonstrated in finding optimal sentence feature weights for ETS methods. It was discovered that sentence location, proper noun, and named entity features get relatively higher weights because they are more critical for summary sentence selection (Meena and Gopalani, 2015).

- Vector representations produced by identifying and extracting the relationship between the input text main features and repetitive patterns, optimized by the Genetic Algorithm, used to generate precise, continuous, and consistent summaries (Ebrahim et al., 2021).

In the scope of our research, we are to apply a Genetic Algorithm to find the upper bound for summary quality achievable with the ETS methods. For example, Simón et al. (2018) described a method based on a Genetic Algorithm to find the best sentence combinations of DUC01/DUC02 datasets in Multi-Document Text Summarization (MDS) through a Meta-document representation.

3 METHODS AND DATA

3.1 Data

The arXive[3] dataset, firstly introduced in 2018 (Cohan et al., 2018), contains 215K scientific articles in the English language from the astrophysics, math, and physics domains. The dataset comprises article texts, abstracts (reference or "golden" summary), article section lists, and main texts divided into sections.

Articles with abstracts that were accidentally longer than the main text and those with extremely long or short texts were excluded from the dataset. Thus, we end up with 17,038 articles with abstracts of 10 to 20 sentences; see Table 1.

	Text length	Abstract length
count	17,038	
mean	263.44	11.75
std	102.57	2.13
min	100.00	10.00
25%	179.00	10.00
50%	252.00	11.00
75%	338.00	13.00
max	500.00	20.00

Table 1. Cleaned arXive dataset description.

3.2 Methods

3.2.1 Variable Neighborhood Search (VNS)

VNS is a heuristics method, exploiting the idea of gradual and systematical change in initial random solution space to find the approximate optimum of the objective function (Burke and Graham, 2014).

The VNS bases on the following facts (Burke and Graham, 2014):

1. Local minima of different neighborhood structures are not necessarily the same.

2. The global minimum is the same for all existing neighborhood structures.

3. In many problems, neighborhood structures local minima are close to each other.

The pseudo-code of the Reduced VNS, a variant of VNS that is not using the local search algorithm applied in this paper, is given in Figure 2.

[3] `arXiv.org`

Initialization. Select the set of neighborhood structures $\mathcal{N}_k$, for $k = 1, \ldots, k_{\max}$, that will be used in the search; find an initial solution x; choose a stopping condition;
Repeat the following sequence until the stopping condition is met:
(1) Set $k \leftarrow 1$;
(2) *Repeat* the following steps until $k = k_{\max}$:
 (a) Shaking. Generate a point x' at random from the kth neighborhood of x ($x' \in \mathcal{N}_k(x)$);
 (b) Move or not. If this point is better than the incumbent, move there ($x \leftarrow x'$), and continue the search with $\mathcal{N}_1$ ($k \leftarrow 1$); otherwise, set $k \leftarrow k+1$;

Figure 2. Pseudo-code for the Reduced VNS

3.2.2 Greedy algorithm
A Greedy algorithm is any algorithm that follows the problem-solving heuristic of taking the best local solution for an optimization task (Black, 2005). For some problems, a greedy heuristic can yield locally optimal solutions approximating a globally optimal solution for a reasonable amount of time.

3.2.3 Genetic algorithm
A *genetic algorithm* is a meta-heuristic method inspired by the natural process of selection belonging to the larger class of evolutionary algorithms. Genetic algorithms are widely used to generate solutions to optimization and search problems by using such operators as a crossover, mutation, and selection, which meet in adaptation and evolutionary processes of living species reproduction (Mitchell, 1998).

3.3 Evaluation
We use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scoring (Lin, 2004) for summary evaluation. The metric basic idea is in calculating the n-grams intersection percentage of reference (*recall*; see Equation 1) and candidate (*precision* summaries; see Equation 2). The harmonic mean integration between *recall* and *precision* is called the $F1$ score (Equation 3).

$$recall = \frac{len(R \cap C)}{len(R)}, \tag{1}$$

where R and C are the set of unique `n_grams` in reference and candidate summaries, and $len()$ is the number of words in a set.

$$precision = \frac{len(R \cap C)}{len(C)}. \tag{2}$$

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall}. \tag{3}$$

4 EXPERIMENTS
In our previous article (Akhmetov et al., 2021b) we searched for the best possible ROUGE-1 score using the VNS heuristic algorithm only. However, in this paper, we added the ROUGE-2 score and applied greedy and genetic algorithms for comparison.

Using the Brute Force algorithm to find the combination of sentences yielding the highest ROUGE score has the *O(n!)* computational complexity and therefore is not feasible; see Equation 4. Therefore, we need to apply a heuristic algorithm to approximate the achievable upper level of summary quality.

We need to apply optimization algorithms because selecting the best possible combination of sentences for a summary from the original text using the Brute Force algorithm has the *O(n!)* computational complexity and therefore is not feasible; see Equation 4.

$$\binom{N_t}{N_a} = \frac{N_t!}{N_a!(N_t - N_a)!} \tag{4}$$

where N_a and N_t - are the respective number of sentences in summary and text.

Optimization algorithms provide a better alternative to Brute Force algorithms by generating not exact but an approximate and satisfactory solution using fewer computational resources and for a reasonable amount of time.

Therefore, we use VNS, Greedy and Genetic algorithms to find the best combinations of sentences from article texts yielding the highest ROUGE-1 score with original article abstracts as a reference.

4.1 VNS

Using the VNS terminology, for every article in our dataset (Table 1), we cyclically applied the following procedures:

1. **Initial solution**: which is a randomly selected set of sentences x in $\mathcal{N}_k = \binom{N_t}{N_a}$ possible neighborhood structure space, for which we get the ROUGE-1 (Lin, 2004) score as the initial best solution to improve on.
2. **Shaking**: we change the initial solution by replacing a randomly selected sentence with a different one from the source text, increasing the rate of changes k up to k_{max} if no improvement in the ROUGE-1 score occurs, limiting the magnitude of the changes to a k_{max} parameter ($k_{max} = 3$, three sentence replacements at a time in our case).
3. **Incumbent solution**: if the obtained summary ROUGE-1 score is better than the previous best solution, we fix the result and reset the k to one sentence.
4. **Stop condition**: we limit the cycle by 60 seconds, 5,000 iterations, or 700 consecutive iterations without improvement of the ROUGE-1 score.

4.2 Greedy algorithm

We used the following Greedy algorithm realization based on the general idea of the optimization algorithm of this class, where we try to find the most feasible immediate solution.

Given a source text (T) split into Sentences (S), and accompanied by its "golden" summary (A):

1. Compile a vocabulary of words from A as (V).
2. Create a word occurrence matrix (M), where we treat each item in V as columns, sentences in T as rows, and binary values indicating the presence of a word in a sentence.
3. Until matrix M is exhausted:
    - Sum the values in rows of M and get the maximum value sentence index, which is the index of the sentence containing the maximum number of words from the "golden" summary A. Store the obtained index in the Index List (IL).
    - Delete the columns in M for which the current maximum row values sum sentence has non-zero values.
4. To determine the optimal number of summary sentences for maximum ROUGE score:
    - Compute ROUGE score for every `top-n` sentences combination in IL ($1 \leq n \leq len(IL)$).
    - Select the n corresponding to the maximum ROUGE score.
    - Truncate IL to n top sentences.
5. To restore the initial sentence order in T, sort items in IL in the ascending order and assemble a summary by picking sentences from T with the respective indices in sorted IL.
6. Calculate the ROUGE score of the generated summary concerning A.

4.3 VNS initialized by the Greedy

We worked on VNS initialized by the best results achieved by the Greedy algorithm. It is simply the modification of the algorithm described in section 4.1 where we, instead of random initialization, use the sentences from the best summaries attained by the Greedy algorithm. Initialization of the VNS algorithm with a combination of sentences with a relatively high ROUGE score saves the time to achieve this initial result. Moreover, it sets the perspective to improve on top of the result achieved by a different algorithm.

4.4 Genetic algorithm

Inspired by the results which Evolutionary Algorithms show in different applications (Mitchell, 1998), we developed a Genetic algorithm realization for finding the upper bound for the ROUGE score.

Given a text (T) and its abstract (A):

1. Calculate lengths of T and A in number of sentences (len_T and len_A).
2. Shuffle the sentences in T.
3. Generate the initial generation of summary candidates by cutting the sentence list in T to chunks of the size len_A.
4. Set the number of offsprings to half the number of initial candidates ($n\_offsprings$).
5. Proceed for six generations:
   (a) Crossover all candidates between each other by mixing the sentences of two candidates, shuffling them, and randomly selecting len_A number of sentences.
   (b) Calculate the ROUGE-1 score for all the offspring.
   (c) Select top $n\_offsprings$ by ROUGE-1 score and repeat.
6. Select the offspring from the last generation with the highest ROUGE-1 score and return it as the generated summary.

4.5 Genetic algorithm initialized by the Greedy

This algorithm is the same as a randomly initialized Genetic algorithm(section 4.4). Nevertheless, in step 3, we add to the initial candidates the summary generated by the Greedy algorithm (section 4.2). The rationale behind initializing the Genetic algorithm with Greedy algorithm results is to improve on top of already high results, similar to the case in section 4.3.

5 RESULTS

Applying the the algorithms described in section 4 we show that the best results were achieved by the Genetic algorithm initialized by the results of Greedy algorithm 0.59/0.25 for the ROUGE-1/ROUGE-2 scores; see Table 2 and Figure 3. While the best modern neural network models (Zhang et al., 2019; Liu and Lapata, 2019; Lloret et al., 2018) can achieve ROUGE-1 of just $\tilde{0}.48$ and ROUGE-2 of 0.22 on arXive dataset; see Table 3. So there is room for improvement in the ETS methods. Examples of the summaries produced by the algorithms employed in this article can be found at `https://github.com/iskander-akhmetov/Reaching-for-Upper-Bound-ROUGE-Score-of-Extractive-Summarization-Methods/blob/main/arXive_examples.md`

Curiously, the maximum-ROUGE summaries from the five algorithms we used (VNS, Greedy, Genetic, VNS, and Genetic initialized by Greedy) differ in the average number of sentences: 15, 7, 12, 10, and 12, respectively. We attribute the reason that summaries generated by the Greedy Algorithm have seven sentences on average to the fact that the algorithm purposefully chooses the lexically richest sentences, which are longer than average. The issue of selecting long sentences in favor of shorter ones was addressed in MMR paper (Carbonell and Goldstein, 1998), and the proposed solutions sought the balance between the relevance of the sentences and their length by weighing them according to the lexical unit's content. Conversely, VNS tries random sentence combinations not accounting for their properties. Thus, the Greedy algorithm maximizes the ROUGE score with fewer sentences than other algorithms. Moreover, determining the optimal number of sentences to maximize the summary ROUGE score is also challenging.

	VNS		Greedy		VNS_Greedy		Genetic		Genetic_Greedy	
	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
count	17,038									
mean	0.55	0.21	0.55	0.23	0.58	**0.25**	0.58	0.24	**0.59**	**0.25**
std	0.07	0.08	0.08	0.10	0.08	0.10	0.07	0.09	0.08	0.10
min	0.07	0.01	0.04	0.01	**0.09**	**0.02**	**0.09**	0.01	**0.09**	0.01
25%	0.52	0.16	0.51	0.16	0.54	0.18	0.55	0.18	**0.56**	**0.19**
50%	0.56	0.20	0.55	0.21	0.58	0.22	0.59	0.23	**0.60**	**0.24**
75%	0.59	0.25	0.60	0.28	0.62	0.29	0.63	0.29	**0.64**	**0.30**
max	0.84	0.78	**0.97**	0.93	**0.97**	**0.95**	0.86	0.84	0.92	0.88

Table 2. The best ROUGE scores (R-1 and R-2) achievable using ETS methods. Numbers in bold indicate the highest values by row.

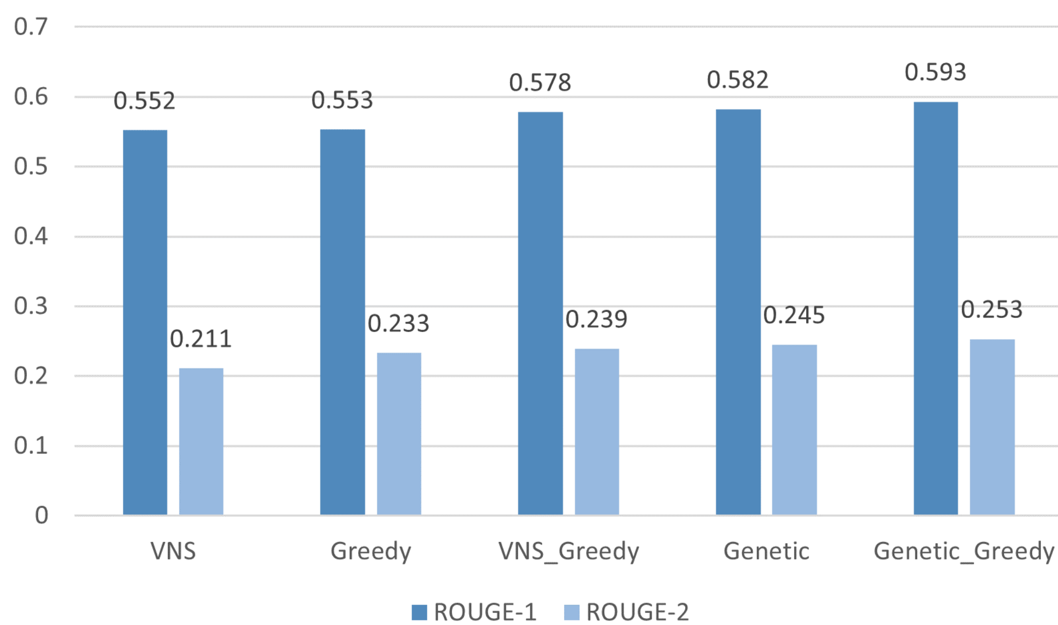

Figure 3. Upper bound ROUGE scores comparison for different methods.

5.1 Error analysis

We have performed an error analysis on the data obtained on ROUGE1/2 calculations for all of the methods employed in this research; see Table 4 and Table 5. For each algorithm we have calculated the Coefficient of Variation (CV) defined in Equation 5, and Confidence Interval (CI) defined in Equation 6, with the confidence level of 95%.

$$CV = \frac{\sigma}{\mu}, \tag{5}$$

where σ is the standard deviation (std) and μ is the mean.

$$CI = \mu \mp Z \times \frac{\sigma}{\sqrt{N}}, \tag{6}$$

where Z is the Z-value associated with the desired confidence level (for 95% confidence level in our case, Z-score = 1.956), and N is the number of observations.

We see in Table 4 that the Genetic algorithm initialized by the Greedy algorithm results demonstrates

Class	Model	ROUGE-1	ROUGE-2
	Genetic_Greedy upper bound	**0.59**	**0.25**
Extractive	SumBasic (Cohan et al., 2018; Lin, 2004; Vanderwende et al., 2007)	0.30	0.07
	LexRank (Cohan et al., 2018; Erkan and Radev, 2004)	0.34	0.11
	LSA (Cohan et al., 2018; Jezek et al., 2004)	0.30	0.07
Abstractive	Attn-Seq2Seq (Cohan et al., 2018; Nallapati et al., 2016)	0.29	0.06
	Pntr-Gen-Seq2Seq (Cohan et al., 2018; See et al., 2017)	0.32	0.09
	Discourse-att (Cohan et al., 2018)	0.36	0.11
	$\text{PEGASUS}_{\text{BASE}}$ (Zhang et al., 2019)	0.35	0.10
	$\text{PEGASUS}_{\text{LARGE}}$ (Zhang et al., 2019)	0.45	0.17
	BigBird-Pegasus (Zaheer et al., 2020)	0.47	0.19
	BertSumExtMulti (Sotudeh et al., 2020)	0.48	0.19
	LongT5 (Guo et al., 2022)	0.48	0.22
	PRIMERA (Xiao et al., 2022)	0.48	0.21

Table 3. Comparison of the upper bound obtained with the leading modern ATS models results on the arXive dataset. Numbers in bold indicate maximum values by column.

Table 4. ROUGE1 error analysis.

Algorithm	mean	std	CV	CI +/- mean	CI lower	CI upper
VNS	0.5500	0.0700	0.1273	0.0011	0.5489	0.5511
Greedy	0.5500	0.0800	0.1455	0.0012	0.5488	0.5512
VNS_Greedy	0.5800	0.0800	0.1379	0.0012	0.5788	0.5812
Genetic	0.5800	0.0700	**0.1207**	0.0011	0.5789	0.5811
Genetic_Greedy	**0.5900**	0.0800	0.1356	0.0012	**0.5888**	**0.5912**

the highest ROUGE1 score mean (0.5900) and highest values of upper(0.5912) and lower (0.5888) bounds of the CI. However, the Genetic algorithm alone has the lowest CV (0.1207).

Table 5 shows that for ROUGE2 score means and CI upper and lower bounds are highest for both the VNS and Genetic algorithms initialized by the results of the Greedy algorithm. Moreover, we see that CV values for the ROUGE2 score almost tripled, which means that these values are more dispersed and volatile than the ROUGE1 score average values. Moreover, again Genetic algorithm has the lowest CV value.

6 DISCUSSION

As we saw in our experiments, for ETS methods, selecting the optimal number of sentences to extract from the source text is detrimental to maximizing the ROUGE score of summaries. However, we detected no strong correlation between the optimal number of sentences for any of the algorithms and other factors such as the number of characters, words, and sentences in a source text and their derivative features (number of words per sentence or characters per word).

The summary length importance has been studied previously by Ježek and Steinberger (Jezek et al., 2004). However, they inferred by the Latent Semantic Analysis (LSA) evaluation only that the more extended summaries are, the better. Their article was published the same year the ROUGE score was

Table 5. ROUGE2 error analysis.

Algorithm	mean	std	CV	CI +/- mean	CI lower	CI upper
VNS	0.2100	0.0800	0.3810	0.0012	0.2088	0.2112
Greedy	0.2300	0.1000	0.4348	0.0015	0.2285	0.2315
VNS_Greedy	**0.2500**	0.1000	0.4000	0.0015	**0.2485**	**0.2515**
Genetic	0.2400	0.0900	**0.3750**	0.0014	0.2386	0.2414
Genetic_Greedy	**0.2500**	0.1000	0.4000	0.0015	**0.2485**	**0.2515**

introduced by Lin (2004) to assess the summary quality automatically, which is now the summary evaluation "industry" standard. However, using the ROUGE score implies that more extended summaries increase the recall at the expense of precision. So further research is needed to determine the optimal number of summary sentences to maximize the ROUGE score value.

Another issue is that using the ROUGE scoring methodology presumes that the reference summaries are ground truth. However, we still have to check the "golden" summaries relative to their source text as they might be a teaser-style indicative summary. Alternatively, the reference summary we use in ROUGE scoring might be very abstractive, containing different wording than the source text, which leads ETS methods to failure.

A different question of whether the ROUGE metric suits the goal of measuring the information overlap of the generated summary with the golden summary was researched by Deutsch and Roth (2021), and it was found that the metric instead measures the extent to which both summaries have the same topic. So there is a need to develop evaluation metrics to account for the informativeness of the generated summary relative to the source text and golden summary.

7 CONCLUSION

We showed five algorithms to approximate the highest possible ROUGE score for ETS methods tested on the extract from the arXive dataset Cohan et al. (2018). We used the VNS technique in our prior publication (Akhmetov et al., 2021b), and in this paper, we explored Genetic and Greedy algorithms. The latter inspired us to develop a novel type of summarization algorithms (Akhmetov et al., 2021a). We showed that there is still a way to improve the ETS methods to reach the 0.59 ROUGE-1 score, while the latest contemporary summarization models do not surpass 0.48.

Our future work plan is to research:

1. Determine the optimal number of sentences in summary to maximize the ROUGE score in each case.
2. Narrowing the sentence search space for heuristic algorithms by excluding presumably unfit sentences (ex., too short sentences, and others).
3. Test the heuristic algorithms described here on different text summarization datasets.

8 ACKNOWLEDGEMENT

This research was conducted with the support from Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan in the course of "Development of language-independent unsupervised semantic analysis methods large amounts of text data" project.

The work was done with the support from the Mexican Government as well. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico.

REFERENCES

Abualigah, L., Bashabsheh, M. Q., Alabool, H., and Shehab, M. (2020). Text summarization: A brief review. *Studies in Computational Intelligence*, 874(January):1–15.

Akhmetov, I., Gelbukh, A., and Mussabayev, R. (2021a). Greedy optimization method for extractive summarization of scientific articles. *IEEE Access*, pages 1–1.

Akhmetov, I., Mladenovic, N., and Mussabayev, R. (2021b). Using k-means and variable neighborhood search for automatic summarization of scientific articles. In Mladenovic, N., Sleptchenko, A., Sifaleras, A., and Omar, M., editors, *Variable Neighborhood Search*, pages 166–175, Cham. Springer International Publishing.

Black, P. E. (2005). Dictionary of algorithms and data structures.

Burke, E. K. and Graham, K. (2014). *Search methodologies: Introductory tutorials in optimization and decision support techniques, second edition*. Springer, Switzerland.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on*

Research and Development in Information Retrieval, SIGIR ’98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Ceylan, H., Mihalcea, R., Özertem, U., Lloret, E., and Palomar, M. (2010). Quantifying the limits and success of extractive summarization systems across domains. In *NAACL HLT 2010 - Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, pages 903–911.

Chatterjee, N., Mittal, A., and Goyal, S. (2012). Single document extractive text summarization using genetic algorithms. In *2012 Third International Conference on Emerging Applications of Information Technology*, pages 19–23.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2:615–621.

Deutsch, D. and Roth, D. (2021). Understanding the extent to which content quality metrics measure the information quality of summaries. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 300–309, Online. Association for Computational Linguistics.

Ebrahim, H., Hamïd, P., Samad, N., Karamollah, Bagherifard Vahideh, R., Zulkefli, M., and Kim-Hung, P. (2021). Automatic text summarization using genetic algorithm and repetitive patterns. *Computers, Materials & Continua*, 67(1):1085–1101.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*.

Gillick, D., Riedhammer, K., Favre, B., and Hakkani-Tur, D. (2009). A global optimization framework for meeting summarization. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4769–4772.

Guo, M., Ainslie, J., Uthus, D., Ontanon, S., Ni, J., Sung, Y.-H., and Yang, Y. (2022). LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736, Seattle, United States. Association for Computational Linguistics.

Hansen, P. and Mladenović, N. (2001). J-means: a new local search heuristic for minimum sum of squares clustering. *Pattern Recognition*, 34(2):405–413.

Hansen, P. and Mladenović, N. (2018). Variable neighborhood search.

Hansen, P., Mladenović, N., Moreno Pérez, J. A., and Moreno Pérez, J. A. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*.

Jezek, K., Steinberger, J., and Ježek, K. (2004). Using latent semantic analysis in text summarization and summary evaluation. In *Proceedings of the 7th International Conference ISIM*.

Kupiec, J. and Pedersen, J. (1995). A trainable document summarizer. *of the 18th annual international ACM*.

Li, W. and McCallum, A. (2006). Pachinko allocation: Dag-structured mixture models of topic correlations. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML ’06, page 577–584, New York, NY, USA. Association for Computing Machinery.

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, H., Bilmes, J., and Xie, S. (2009). Graph-based submodular selection for extractive summarization. In *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, pages 381–386.

Liu, Y. and Lapata, M. (2019). Text summarization with pretrained encoders. *arXiv Computer Science*.

Lloret, E., Plaza, L., and Aker, A. (2018). The challenging task of summary evaluation: an overview. *Language Resources and Evaluation*, 52(1):101–148.

Luhn, H. P. (1958). The automatic creation of literature. *IBM Journal of Research and Development*, 2(2):159–165.

Meena, Y. K. and Gopalani, D. (2015). Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science*, 48:244–249. International Conference on Computer, Communication and Convergence (ICCC 2015).

Mihalcea, R. and Radev, D. (2011). *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.

Nallapati, R., Zhou, B., dos Santos, C., Gulçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, pages 280–290, Stroudsburg PA, USA. Association for Computational Linguistics (ACL).

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2011). English gigaword fifth edition, linguistic data consortium.

Radev, D. R., Hovy, E., and McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics*.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, volume 1, pages 1073–1083.

Simón, J. R., Ledeneva, Y., and García-Hernández, R. A. (2018). Calculating the upper bounds for multi-document summarization using genetic algorithms. *Computación y Sistemas*, 22:11–26.

Sotudeh, S., Cohan, A., and Goharian, N. (2020). On generating extended summaries of long documents.

Vanderwende, L., Suzuki, H., Brockett, C., and Nenkova, A. (2007). Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*.

Verma, R. and Lee, D. (2017). Extractive summarization: Limits, compression, generalized model and heuristics. *Computación y Sistemas*, 21:787–798.

Wang, W., Li, Z., Wang, J., and Zheng, Z. (2017). How far we can go with extractive text summarization? heuristic methods to obtain near upper bounds. *Expert Systems with Applications*, 90:439–463.

Xiao, W., Beltagy, I., Carenini, G., and Cohan, A. (2022). PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. (2020). Big bird: Transformers for longer sequences. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2019). Pegasus: Pre-training with extracted gap-sentences for abstractive subbarization. *arXiv Computer Science*.