11th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN), Nis, 3-6 June 2024

Comparing Evolved Extractive Text Summary Scores of Bidirectional Encoder Representations from Transformers

Aleš Zamuda
Faculty of Electrical Engineering and Computer Science, Institute of Computer Science
University of Maribor
Koroška cesta 46, 2000
Maribor, Slovenia
Email: ales.zamuda@um.si

Jani Dugonik
Faculty of Electrical Engineering and Computer Science, Institute of Informatics
University of Maribor
Koroška cesta 46, 2000
Maribor, Slovenia
Email: jani.dugonik@um.si

Elena Lloret
Department of Software and Computing Systems
University of Alicante
Apartado 99, E-03080
Alicante, Spain
Email: elloret@dlsi.ua.es

***Abstract*—This paper presents a comparison of a set of two evaluation metrices on a summarization system. The summarization system applied is based on algorithm CaBiSDETS (Constraint-adjusting Binary Self-adaptive Differential Evolution for Data-Driven Models Extractive Text Summarization Optimization). The summarization algorithm is executed on DUC (Document Understanding Conference) corpus. The two compared metrices set consists of ROUGE (Recall-Oriented Understudy for Gisting Evaluation) that was the original metric from DUC, and of the more recent BERTScore from BERT (Bidirectional Encoder Representations from Transformers). Twelve sample initial documents from the corpus are selected and their evolutionary runs are plotted. The plots report the scores on the two corresponding metrices over obtained evolved summaries through each of the runs. Then, observations are discussed regarding the feedbacks from both metrices.**

I. Introduction

In this paper, an empirical comparison of a set of two evaluation metrices for an extractive text summarization system is presented. The two compared metrices are ROUGE [1] (Recall-Oriented Understudy for Gisting Evaluation) that was the original metric from DUC (Document Understanding Conference) and indeed, is the standard widely used metric for summarization evaluation, and the more recent BERTScore metric [2], [3] based on contextual embeddings in pretrained BERT (Bidirectional Encoder Representations from Transformers) [4], a Google AI Large Language Model (LLM). The summarization system assessed using these metrices is based on algorithm CaBiSDETS [5] (Constraint-adjusting Binary Self-adaptive Differential Evolution for Data-Driven Models Extractive Text Summarization Optimization) and is executed on DUC 2002 corpus at https://duc.nist.gov/, to observe correlations of scores among the two metrices on same texts. Both ROUGE and BERTScore compute the similarity of two texts, but as ROUGE compares the matching parts of text, BERTScore computes a sum of cosine similarities between tokens' embeddings, which is closer to the CaBiSDETS summary fitness evaluation that also uses sum of cosine similarities and hence the fitness values and metrices become an interesting point of observation and comparison.

In the next section, more related work is presented. In the third section, the presented method of selecting both metrices and evaluation on DUC is presented. In the fourth section, the evaluation result are presented and discussion about them regarding both understanding evaluation metrices is provided. The fifth section provides conclusions and some further research directions, then the references are listed.

II. Related Work

In this section, related work on text summarization and algorithms is presented, followed by some more background on summary evaluation using metrices like ROUGE and BERTScore in DUC corpus for extractive text summaries.

A. Text Summarization

The task of text summarization has been researched for more than 50 years. Despite this, it is still a very challenging task in Natural Language Processing (NLP) [6], [7]. It is difficult due partly to the fact that there is a lot of subjectivity in the process that is influenced by many cognitive aspects [8], [9]. Whereas the main objective of text summarization is to produce a summary automatically, i.e., with no human intervention, such a summary could be output in many different forms, also being influenced by a wide range of factors that should be considered during the process of summary generation. In this sense, there are different classical taxonomies proposed in the literature that lead to different types of summaries [10], [11]. Text Summarization, the problem selected as the case study in this paper, is not only an important challenge within NLP due to the real and big data processing, but also, moreover, technically, as a large scale, non-linear, constrained, and non-separable problem in the benchmarking domain [12].

ISBN 978-86-6200-001-9 ©2024 AM & ETRAN https://doi.org/10.69994/11Ic24043

A special type of summarization is sentence extraction from multiple documents, i.e. multi-document extractive summarization. Different approaches can be found in the literature for addressing multi-document summarization, which range from simple approaches using statistical techniques, such as tf-idf [13], to more recent ones that use neural computing [14], [15], [16]. However, efficiency is not normally taken into account, and although summaries can obtain good results in terms of their content, the associated drawback concerns the impossibility to apply those approaches in realtime scenarios, especially those approaches based on Neural Networks (NNs) that require a lot of training time. One approach in-between is to take into account optimization issues and integrate them within the summarization approach. For instance, Alguliev et al. [17] applied Differential Evolution (DE) to multi-document summarization using sentence extraction, being formalized as a discrete optimization problem. Further on, Alguliev et al. also extended their work, modeling the multi-document summarization tasks as different algorithmic problems, such as a quadratic boolean programing problem; a non-linear programing problem; or as a modified p-median problem [18], [19], [20], [21], [22]. Their approach has been extended in [5]. As an optimizer, DE [23] is a floating-point encoding Evolutionary Algorithm [24], [25] for global optimization over continuous spaces, and, since its introduction, has formed the basis for a set of successful algorithms for optimization domains, such as continuous, discrete, mixed-integer, or other search spaces and features [26]. The whole encompassing research field around DE was surveyed most recently in [27], and even since then, several other domain- and feature-specific surveys, studies, and comparisons have also followed [28], [29], [30], [31], [32], [33], [34], [35]. Theoretical insight and insights to inner workings and behaviors of DE during consecutive generations has been studied in works like [36], [37], [38], [32], [39], [40], [41]. The DE algorithm has a main evolution loop in which a population of vectors is computed for each generation of the evolution loop. During one generation g, for each vector $\mathbf{x}_i$, $\forall i \in \{1, 2, ..., NP\}$ in the current population, DE employs evolutionary operators, namely mutation, crossover, and selection, to produce a trial vector (offspring) and to select one of the vectors with the best fitness value. NP denotes population size and $g \in \{1, 2, ..., G\}$, the current generation number.

B. Summary Evaluation

The existing DUC collection of English newswire documents from the Document Understanding Conferences (DUC) fits very well for experimenting within the text summarization scenarios [5]. On the one hand, it provides pairs of documents-summary or cluster-summary for diverse summarization types (extractive, single-document, multi-document, etc.), and, on the other hand, there is a wide number of previous summarization systems to be compared with, and determine to what extent our approach is effective. In particular, the DUC 2002 dataset, contains 567 documents grouped in 59 clusters (denoted as `d061` to `d120`), where each cluster represents a set of topic-related documents (the average number of documents per cluster is 10). Besides this dataset, a dataset, i.e., CNN/DailyMail has also been made available for the research community [42]. This dataset contains more than 300,000 documents, and it provides summaries of about 50 words.

Concerning the evaluation of summaries, ROUGE [1] is one of the common standard, and most used tools. The idea behind ROUGE is that, if two texts have a similar meaning, they must also share similar words or phrases. As a consequence, it relies on n-gram co-occurrence, and the idea behind it is to compare the content of a peer summary with one or more model summaries, and compute the number of n-gram of words they all have in common. Different types of n-grams can be obtained, such as unigrams (ROUGE-1), bigrams (ROUGE-2), the longest common subsequence (ROUGE-L), or bigrams with a maximum distance of four words in-between (ROUGE-SU4), and, based on them, values for recall, precision and F-measure can be obtained, thus determining the summary accuracy in terms of content (the higher recall, precision and F-measure values, better). Among all the metrices, recall values are then usually reported for peer evaluation of generated model summaries. Other metrices also exist, like AutoSummENG [43] which is an automatic character n-gram based evaluation method with high correlation with human judgments, or SumTriver [44], which does not need to have human summaries for the evaluation; however, they are not often used by the research community, thus it is difficult to find results for comparison purposes. Other ways to evaluate the summaries would be to consider standard similarity metrices, such as Simmetric (https://github.com/Simmetrics/simmetrics).

In NLP, sentence embeddings relate to methodologies that encapsulate the semantic essence of complete sentences, portraying them as condensed numerical vectors. This approach enables a diverse array of subsequent tasks, including but not limited to sentence similarity assessment, paraphrase detection, and text classification. The embeddings derived from BERT capture contextual information, as it computes token similarity using contextual embeddings. Multilingual BERT (mBERT) [45] in text summarization proves to be a transformative approach for enhancing the efficiency and accuracy of the summarization process across diverse languages. As an extension of the original BERT model, mBERT enables the understanding of the meaning and context of words and sentences across multiple languages, having been trained on an extensive corpus with over 100 different languages. Work with mBERT has been applied also in [46], embedding source sentences and translations into a shared vector space for machine translation. BERTScore [2] has been therefore proposed as an evaluation metric derived from BERT models. A comprehensive literature review on summarization evaluation methods and metrices can be found in [47].

III. Methodology

To compare the values from both metrices ROUGE and BERTScore, we use the initial twelve document instances `d061j` to `d072f` from DUC 2002 corpus to execute the runs of summarization. The summarization system employed was an adapted version of CaBiSDETS (Constraint-adjusting Binary Self-adaptive Differential Evolution for Data-Driven Models Extractive) [5], which instead of preprocessing to concepts used just splits of sentences with lematization, but still used sum of cosine similarities to evolve fitness. During the run, the summarization system improves current fitness and for viable summaries, their assessments are reported as (`Fitness`). A `Fitness` is a fitness value of the represented summary which the optimization algorithm extracted and is scoring it using its fitness function during the optimization. For ROUGE and BERTScore metrices, that evolved summary (`peer`) is then compared to one of the two model summaries (`model1` or `model2`) being provided by the DUC corpus of corresponding summaries by humans who wrote the model summaries. The BERTScore metric is configured using the code snippet as seen in Figure 1, where a base model from BERT is used for scoring, while a multilingual BERT model is used for the remaining values. Then, the evaluation through contextual embeddings is seen in Figure 2, where for a generated evolved summary in text file `text0`, two human summaries are used as input text files `text1` and `text2` to print the BERTScore metric values.

```
model_name = 'bert-base-multilingual-uncased'
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertModel.from_pretrained(model_name)
scorer = BERTScorer(model_type='bert-base-uncased')
```

Fig. 1. Configuring the BERT models.

```
inputs0 = tokenizer(text0, return_tensors='pt', padding=True, truncation=True)
outputs0 = model(**inputs0)
embeddings0 = outputs0.last_hidden_state.mean(dim=1).detach().numpy()

inputs1 = tokenizer(text1, return_tensors='pt', padding=True, truncation=True)
inputs2 = tokenizer(text2, return_tensors='pt', padding=True, truncation=True)
outputs1 = model(**inputs1)
outputs2 = model(**inputs2)
embeddings1 = outputs1.last_hidden_state.mean(dim=1).detach().numpy()
embeddings2 = outputs2.last_hidden_state.mean(dim=1).detach().numpy()

# BERT: metric 0 to 1
similarity = np.dot(embeddings0, embeddings1.T) / (np.linalg.norm(embeddings0) \
             * np.linalg.norm(embeddings1))
P, R, F1 = scorer.score([text0], [text1])
sim, P, R, F1 = similarity[0][0], P.mean(), R.mean(), F1.mean()
print(f'BERTScore␣sim/P/R/F1@1:␣{sim:.4f}␣{P:.4f}␣{R:.4f}␣{F1:.4f}␣', end='␣')

# BERT: metric 0 to 2
similarity = np.dot(embeddings0, embeddings2.T) / (np.linalg.norm(embeddings0) \
             * np.linalg.norm(embeddings2))
P, R, F1 = scorer.score([text0], [text2])
sim, P, R, F1 = similarity[0][0], P.mean(), R.mean(), F1.mean()
print(f'sim/P/R/F1@2:␣{sim:.4f}␣{P:.4f}␣{R:.4f}␣{F1:.4f}␣')
```

Fig. 2. Scoring summary files using BERTScore and human input.

IV. Results

In Figures 3 and 4, the plots using the evolved text summaries are provided. The text summarization metrices shown are Fitness, ROUGE, and BERTScore (BERT). The Fitness metric is the evaluation of the generated evolved summary which the optimization algorithm extracted and is scoring it using its fitness function during the optimization. The values of Fitness are plotted by normalizing the values between 0 and 1 in a certain plot. For ROUGE and BERTScore, that evolved summary (`peer`) is compared to one of the two model summaries (`model1` or `model2`) being provided by the DUC corpus of corresponding summaries by a human who wrote a corresponding model summary. The ROUGE metric values reported are the ROUGE 1, ROUGE 2, ROUGE 4, and ROUGE SU. The BERTScore values reported are the similarity score (`sim`) or BERTScore precision (`P`), recall (`R`), and F1 value (`F1`), respectively.

As seen from the plots in Figures 3 and 4, as the Fitness improves by increasing in its value, with each jump the assessment metric values from ROUGE and BERTScore also change, hence we can deduce from these observations that the metrices do relate a change in response to a different text summary. Also, while the curves of ROUGE values are seen as all lower than numbers from BERTScore, the curves from values of BERTScore similarity score (`sim`) are highest, hence from the experiment, we can expect to obtain values of BERTScore being higher as those from ROUGE. Also, the matching values from BERTScore in evaluation on the two corresponding human summaries are closely correlated, i.e., the generated extractive summary always scored closely related on both corresponding human models: merely in the case of `d068f`, the points in lines at generations $g = 1000$ and $g = 3000$ interweave slightly. From these observations on our summarization system, the main takeaways are that the metric BERTScore provided higher values than ROUGE and that both metrices evaluated a summary closely responsive to a corresponding human model summary.

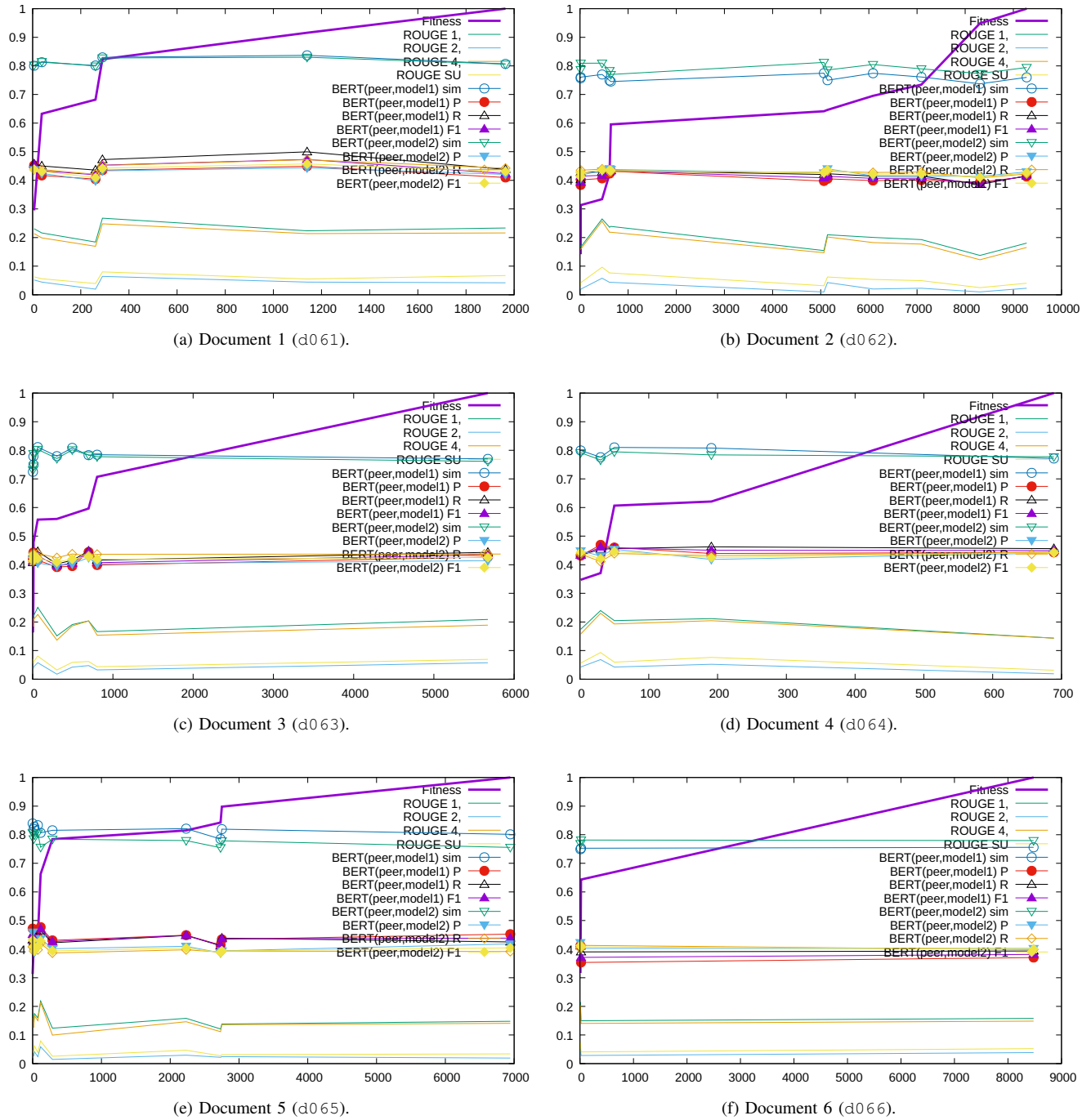
V. Conclusion

This paper presented a comparison of a set of two evaluation metrices for a summarization system, where a summarization system based on algorithm CaBiSDETS was applied. A summarization algorithm was executed on DUC documents and then evaluated on two metrices, ROUGE that was the original metric from DUC, and of the more recent BERTScore using BERT from Google AI. The observed results have shown close correlation on same matching summarization texts.

In future work, research on further use of BERTScore and similar LLMs metrics on DUC corpora summarizers could be conducted, but also improving and analysing summarization algorithms, as well as developing new metrices for text summarization and understanding, or deployment of trained ML systems to different modalities like video and animation.

Acknowledgment

This work is supported by project DAPHNE (Integrated Data Analysis Pipelines for Large-Scale Data Management, HPC and Machine Learning) funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No 957407 and ARIS (Slovenian Research



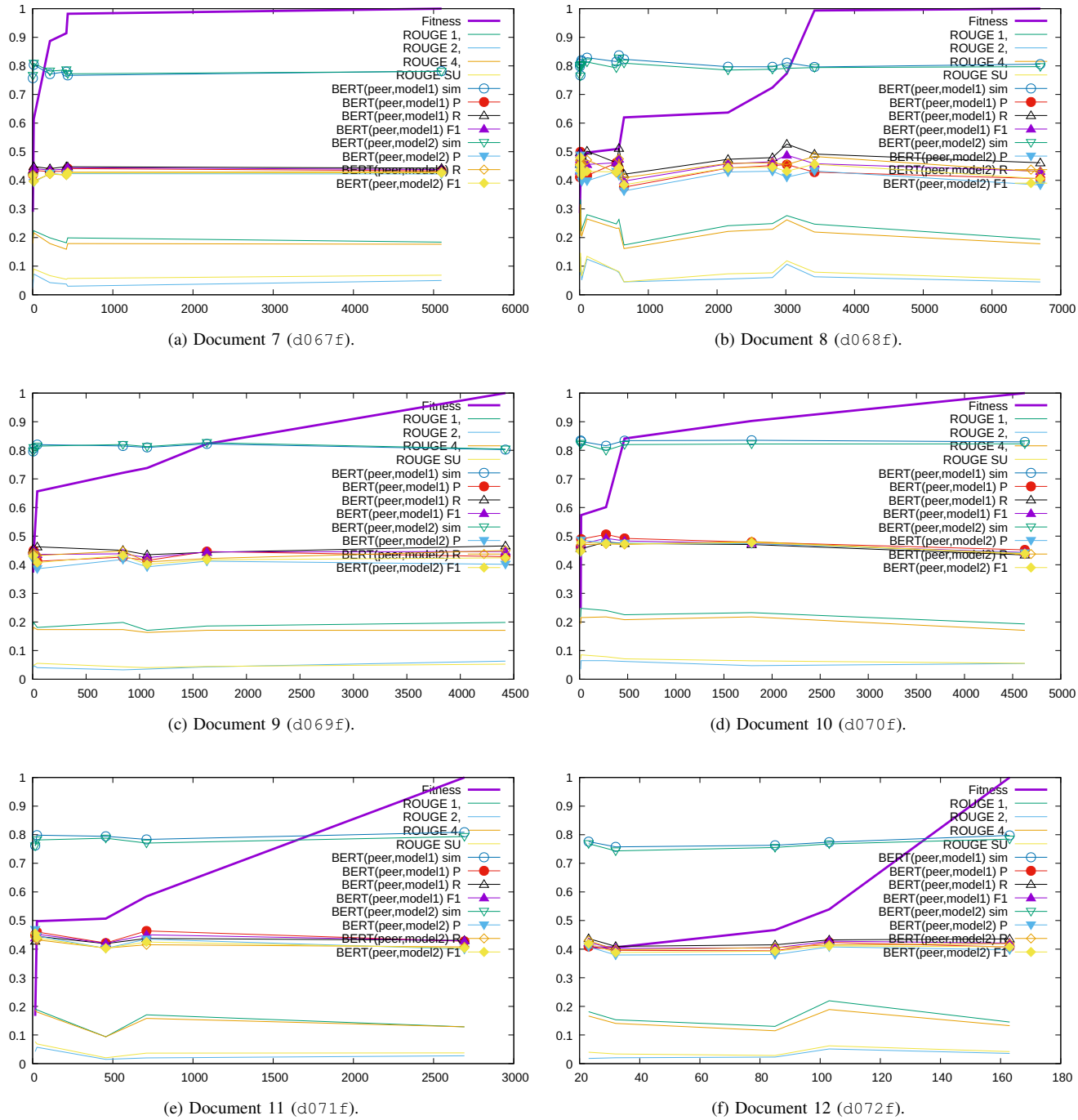
(a) Document 1 (`d061`). (b) Document 2 (`d062`).

(c) Document 3 (`d063`). (d) Document 4 (`d064`).

(e) Document 5 (`d065`). (f) Document 6 (`d066`).

Fig. 3. Evaluations for documents from DUC corpus, for documents 1–6. The subfigures (a) to (f) plot the text summarization metrices (normalized value of Fitness, then values of ROUGE 1/2/4/SU, respectively, and finally the individual comparisons of values from BERTScore) on documents `d061j` to `d066j`. Their value scales are represented on the vertical axis, while on the horizontal axis the current sequential generation number (g) through evolution are shown. Only the feasible values and where there is an improvement in the fitness are plotted. The values from BERTScore are in eight combinations, where each time, the currently generated evolved summary text (`peer`) is compared to one of the two model summaries (`model1` or `model2`), reported though the similarity score (`sim`) or BERTScore precision (`P`), recall (`R`), and F1 value (`F1`), respectively.

And Innovation Agency) programme P2-0041 (Computer Systems, Methodologies, and Intelligent Services). Part of travel expense to present the paper is also supported through IEEE GRSS Inter Society Networking (ISN) Activities grant involving IEEE Slovenia GRSS chapter and IEEE Slovenia CIS chapter. Moreover, this work is also conducted as part of the R&D project "CORTEX: Conscious Text Generation" (PID2021-123956OB-I00), funded by MCIN/ AEI/10.13039/501100011033/ and by "ERDF A way of making Europe".



(a) Document 7 (`d067f`). (b) Document 8 (`d068f`).

(c) Document 9 (`d069f`). (d) Document 10 (`d070f`).

(e) Document 11 (`d071f`). (f) Document 12 (`d072f`).

Fig. 4. Evaluations for documents from DUC corpus, for documents 7–12. The subfigures (a) to (f) plot the text summarization metrices (normalized value of Fitness, then values of ROUGE 1/2/4/SU, respectively, and finally the individual comparisons of values from BERTScore) on documents `d067f` to `d072f`. Their value scales are represented on the vertical axis, while on the horizontal axis the current sequential generation number (g) through evolution are shown. Only the feasible values and where there is an improvement in the fitness are plotted. The values from BERTScore are in eight combinations, where each time, the currently generated evolved summary text (`peer`) is compared to one of the two model summaries (`model1` or `model2`), reported though the similarity score (`sim`) or BERTScore precision (`P`), recall (`R`), and F1 value (`F1`), respectively.

References

[1] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of the ACL-04 Workshop on Text Summarization Branches Out*, Barcelona, Spain, 2004, pp. 74–81.

[2] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "BERTScore: Evaluating Text Generation with BERT," in *Proceedings of the ICLR 2020*, 2020.

[3] R. Patil, S. Boit, V. Gudivada, and J. Nandigam, "A Survey of Text Representation and Embedding Techniques in NLP," *IEEE Access*, vol. 11, pp. 36 120–36 146, 2023.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training

of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171—-4186.

[5] A. Zamuda and E. Lloret, "Optimizing Data-Driven Models for Summarization as Parallel Tasks," *Journal of Computational Science*, vol. 42, p. 101101, 2020.

[6] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text Summarization Techniques: A Brief Survey," *International Journal of Advanced Computer Science and Applications*, vol. 8, 2017.

[7] M. H. H. Wahab, N. H. Ali, N. A. W. A. Hamid, S. K. Subramaniam, R. Latip, and M. Othman, "A review on optimization-based automatic text summarization approach," *IEEE Access*, 2023.

[8] W. U. Dressler, *Current Trends in Textlinguistics*, ser. Research in text theory. W. de Gruyter, 1978.

[9] C. Leopold, A. Brückner, and S. Dutke, "Summarizing as a Strategy for Science Text Comprehension: Text-Based Versus Content-Based Processing," *Discourse Processes*, vol. 56, no. 8, pp. 728–747, 2019.

[10] K. Sparck Jones, "Automatic summarising: factors and directions," in *Advances in Automatic Text Summarization*. MIT Press, 1999, pp. 1–12.

[11] I. Mani and M. T. Maybury, Eds., *Advances in Automatic Text Summarization*. MIT Press, 1999.

[12] A. Zamuda, G. Hrovat, E. Lloret, M. Nicolau, and C. Zarges, "Examples Implementing Black-Box Discrete Optimization Benchmarking Survey for BB-DOB@GECCO and BB-DOB@PPSN," in *Black Box Discrete Optimization Benchmarking (BB-DOB) Workshop at 15th International Conference on Parallel Problem Solving from Nature (PPSN 2018), September 8-12, 2018, Coimbra, Portugal*, 2018, p. 1.

[13] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de França Silva, S. J. Simske, and L. Favaro, "A multi-document summarization system based on statistics and linguistic treatment," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5780–5787, 2014.

[14] W. Yin and Y. Pei, "Optimizing sentence modeling and selection for document summarization," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15. AAAI Press, 2015, pp. 1383–1389.

[15] X. Zhang, M. Lapata, F. Wei, and M. Zhou, "Neural latent extractive document summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 779–784.

[16] M. Zhong, P. Liu, D. Wang, X. Qiu, and X. Huang, "Searching for effective neural extractive summarization: What works and what's next," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 1049–1058.

[17] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 213–222, 2011.

[18] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "CDDS: Constraint-driven document summarization models," *Expert Systems with Applications*, vol. 40, no. 2, pp. 458–465, 2013.

[19] R. M. Alguliev, R. M. Aliguliyev, and M. S. Hajirahimova, "GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12 460–12 473, 2012.

[20] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," *Knowledge-Based Systems*, vol. 36, pp. 21–38, 2012.

[21] ——, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Systems with Applications*, vol. 40, pp. 1675–1689, 2013.

[22] ——, "Formulation of document summarization as a 0-1 nonlinear programming problem," *Computers & Industrial Engineering*, vol. 64, pp. 94–102, 2013.

[23] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[24] J. Holland, *Adaptation In Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.

[25] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer, 2003.

[26] A. Zamuda, M. Nicolau, and C. Zarges, "A black-box discrete optimization benchmarking (BB-DOB) pipeline survey: taxonomy, evaluation, and ranking," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO 2018)*, 2018, pp. 1777–1782.

[27] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[28] A. P. Piotrowski, "Review of differential evolution population size," *Swarm and Evolutionary Computation*, vol. 32, pp. 1–24, 2017.

[29] A. P. Piotrowski and J. J. Napiorkowski, "Some metaheuristics should be simplified," *Information Sciences*, vol. 427, pp. 32–62, 2018.

[30] R. D. Al-Dabbagh, F. Neri, N. Idris, and M. S. Baba, "Algorithmic design issues in adaptive differential evolution schemes: Review and taxonomy," *Swarm and Evolutionary Computation*, vol. 43, pp. 284–311, 2018.

[31] A. P. Piotrowski and J. J. Napiorkowski, "Step-by-step improvement of JADE and SHADE-based algorithms: Success or failure?" *Swarm and evolutionary computation*, vol. 43, pp. 88–108, 2018.

[32] A. Zamuda and J. Brest, "Self-adaptive control parameters' randomization frequency and propagations in differential evolution," *Swarm and Evolutionary Computation*, vol. 25, pp. 72–99, 2015.

[33] A. Zamuda, J. D. H. Sosa, and L. Adler, "Constrained Differential Evolution Optimization for Underwater Glider Path Planning in Sub-mesoscale Eddy Sampling," *Applied Soft Computing*, vol. 42, pp. 93–118, 2016.

[34] A. Zamuda and J. D. H. Sosa, "Success history applied to expert system for underwater glider path planning using differential evolution," *Expert Systems with Applications*, vol. 119, no. 1 April 2019, pp. 155–170, 2019.

[35] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, and A. Zamuda, "Distance Based Parameter Adaptation for Success-History based Differential Evolution," *Swarm and Evolutionary Computation*, vol. 50, p. 100462, 2019.

[36] M. Weber, F. Neri, and V. Tirronen, "A Study on Scale Factor in Distributed Differential Evolution," *Information Sciences*, vol. 181, no. 12, 2011.

[37] F. Neri, G. Iacca, and E. Mininno, "Disturbed exploitation compact differential evolution for limited memory optimization problems," *Information Sciences*, vol. 181, no. 12, pp. 2469–2487, 2011.

[38] M. Weber, F. Neri, and V. Tirronen, "A study on scale factor/crossover interaction in distributed differential evolution," *Artificial Intelligence Review*, vol. 39, no. 3, pp. 195–224, 2013.

[39] A. Viktorin, R. Senkerik, M. Pluhacek, and A. Zamuda, "Steady success clusters in Differential Evolution," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016, pp. 1–8.

[40] R. Tanabe and A. S. Fukunaga, "How Far Are We From an Optimal, Adaptive DE?" in *14th International Conference on Parallel Problem Solving from Nature (PPSN XIV)*. IEEE, 2016, pp. 145–155.

[41] K. R. Opara and J. Arabas, "Differential Evolution: A survey of theoretical analyses," *Swarm and evolutionary computation*, vol. 44, pp. 546–558, 2019.

[42] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, pp. 1693–1701.

[43] G. Giannakopoulos, V. Karkaletsis, G. Vouros, and P. Stamatopoulos, "Summarization system evaluation revisited: N-gram graphs," *ACM Trans. Speech Lang. Process.*, vol. 5, no. 3, pp. 5:1–5:39, Oct. 2008.

[44] L. A. Cabrera-Diego and J. Torres-Moreno, "SummTriver: A new trivergent model to evaluate summaries automatically without human references," *Data Knowl. Eng.*, vol. 113, pp. 184–197, 2018.

[45] T. Pires, E. Schlinger, and D. Garrette, "How Multilingual is Multilingual BERT?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 4996–5001.

[46] J. Dugonik, M. Sepesy Maučec, D. Verber, and J. Brest, "Reduction of Neural Machine Translation Failures by Incorporating Statistical Machine Translation," *Mathematics*, vol. 11, no. 11, 2023.

[47] E. Lloret, L. Plaza, and A. Aker, "The challenging task of summary evaluation: an overview," *Language Resources and Evaluation*, vol. 52, no. 1, pp. 101–148, 2018.