CUET_SSTM at the GEM'24 Summarization Task: Integration of Extractive and Abstractive Method for Long Text Summarization in Swahili Language

Samia Rahman, Momtazul Arefin Labib, Hasan Murad, Udoy Das
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology, Bangladesh
{u1904022, u1904111}@student.cuet.ac.bd, hasanmurad@cuet.ac.bd,
u1804109@student.cuet.ac.bd

Abstract

Swahili, spoken by around 200 million people primarily in Tanzania and Kenya, has been the focus of our research for the GEM Shared Task at INLG'24 for being an underrepresented language. We have utilized the XLSUM dataset and have manually summarized 1000 texts from a Swahili news classification dataset. To understand the baseline, we have tested abstractive summarizers (mT5_multilingual_XLSum, t5, mBART), and an extractive summarizer (based on the PageRank algorithm). However, our adopted system consists of an integrated extractive-abstractive model combining the Bert Extractive Summarizer with an abstractive summarizer (t5 or mBART). The integrated model overcomes the drawbacks of both the extractive and abstractive summarization systems and utilizes the benefits from both of them. Our Integrated extractive-abstractive (t5) system performed better than other systems and outperformed GPT-3.5 in the final evaluation.

1 Introduction

In sub-Saharan Africa, Swahili has been regarded as the most spoken language. It has been serving as the national language of Tanzania and Kenya and is also widely spoken in Uganda, Rwanda, Burundi, the Democratic Republic of Congo, and Comoros. It has been the only African language with an estimated 100 million speakers and has played an important role in East and Central Africa as a lingua franca (at Ohio University Swahili Language). Therefore, summarizing tasks in the Swahili language is crucial.

Summarization in the Swahili language has faced challenges because of its rich morphology, multiple dialects, and regional variations (as mentioned in Jerro, 2018). These variations have been important in understanding the context essential for producing relevant summaries. No study has proposed a Swahili-specific monolingual language model with culturally diverse data, mainly due to Swahili being a low-resource language(LRL) with limited data availability (Martin et al., 2022). Additionally, there has been limited research on Swahili summarization.

The primary goal of this paper has been to summarize the Swahili texts for the Generation, Evaluation, and Metrics (GEM) Workshop at the International Conference on Natural Language Generation (INLG'24). The dataset used in this workshop has been introduced by Davis, 2020 and consists of Swahili news classification texts along with their respective classes.

In recent years, automatic text summarization has gained popularity for its ability to summarize text efficiently, quickly, and accurately while maintaining context. It has been categorized into two classes: extractive summarizers and abstractive summarizers (Hahn and Mani, 2000).

Extractive Text Summarizers (ETS) use mathematical calculations to measure sentence similarity. From this sentence similarity, a similarity matrix is formed which is then converted into a graph. In the graph, sentences are nodes and similarity scores are edges. Finally, the summary is constructed from the sentences with the top scores. This can be problematic in some cases. If a text covers multiple topics, like sports and politics, the similarity score diminishes as the topic changes, and thus in the summary, both topics may not be present. Moreover, the highest-scoring sentences may cause redundancy. An Abstractive Text Summarizer (ATS) focuses on the salient

Proceedings of the 17th International Natural Language Generation Conference: Generation Challenges, pages 112–117

concepts in a text. It not only selects key pieces from the text but also presents these key concepts in a new way, thereby eliminating the redundancy problem often encountered with ETS. Additionally, ATS captures the essence of the text even with multiple topics. However, it is more complex than ETS and is typically implemented using LSTM, seq2seq model. A limitation of ATS is that it can only process up to a limited number of tokens as input and any tokens beyond this limit are truncated. As a result, valuable information may be lost. Thus, ATS is not fully beneficial for summarizing long texts that contain a large number of tokens.

A fusion of extractive and abstractive text summarizers can help by utilizing the strengths of both methods. In many texts of our dataset, the number of tokens has exceeded 512. At first, we have implemented an extractive summarizer that reduced the size of the text beyond 512 tokens, keeping all possibly important information. This slightly summarized text has been summarized again by the abstractive summarizer for further refinement. Slightly summarized texts containing fewer than 512 tokens have undergone direct processing by the abstractive summarizer without using the extractive method. This method has enabled the summarization of longer texts and has provided coherent and comprehensive summaries.

To achieve our goal, we have manually summarized 1000 texts from the provided Swahili news classification dataset. Next, We have combined the XLSUM dataset with our manually prepared summaries. After that, we evaluated three abstractive summarizers (mT5_multilingual_XLSum, t5-small, mBART-50), one extractive summarizer (based on the PageRank algorithm), and two integrated extractive-abstractive summarizes. In the integrated system, we have integrated the Bert Extractive Summarizer with some abstractive summarizers(t5-small, mBART-50).

During our comparative analysis, we have trained all the systems on the prepared dataset. The integrated extractive-abstractive summarizer system with the "t5-small" model emerged as the most effective, achieving the highest ROUGE scores. In the final evaluation, This system outpaced GPT-3.5 in the automatic evaluation report in Mille et al., 2024.

Our core contributions in this work include the manual summarization of 1,000 texts from the provided Swahili news classification dataset (Davis, 2020). Also, we have integrated a Bert Extractive Summarizer and an Abstractive Summarizer to ensure context-based summarization of the larger texts. Detailed information on implementation is available in the GitHub repository linked below- `https://github.com/Samia2001/CUET_SSTM_GEM24`.

2 Related Work

The effort of enabling computers to automatically generate summaries has been practiced extensively, due to its vast applications in the processing of natural languages. Previous works on automatic text summarization can be classified into two categories (Hahn and Mani, 2000). They are Extractive summarization and Abstractive summarization.

One of the pristine approaches in Extractive summarization has been found in Luhn, 1958. They have taken into account the frequency of words and their relative positions to rank the sentences. Graph-based algorithms have been used to introduce faster and more scalable extractive summarization approaches. TextRank (Mihalcea and Tarau, 2004) and PageRank (Page, 1998) are two basic and prominent graph-based extractive summarizers. Later on, many other graph-based algorithms have been developed based on these two algorithms. Such as LexRank (Erkan and Radev, 2004) based on PageRank and TopicRank (Bougouin et al., 2013), PositionRank (Florescu and Caragea, 2017) based on TextRank.

Due to the lack of comprehensibility and rationality of Extractive summarization approaches (as mentioned in Saggion and Poibeau, 2013), Abstractive summarization has been introduced. CNN, RNN, LSTM-GRU and GAN-based approaches have been used frequently (Rekabdar et al., 2019, Yang et al., 2020). However, the ultimate improvement in summarization has been done by using Transformers. T5 (Raffel et al., 2020), BART (Lewis et al., 2019) etc. transformer

architectures have been used to summarize texts. They have multilingual versions such as mT5 (Hasan et al., 2021) and mBART (Tang et al., 2020) which enables summarization in the Swahili language. Long text summarization has been a drawback of Abstractive summarization. To overcome this issue, integration of both the extractive and abstraction have been proposed in Wang et al., 2017.

3 Data

The given dataset (Davis, 2020) for this shared task contains a total of 23268 texts that have been collected from BBC News Swahili[1] and several other Tanzanian news websites. We have not used this dataset because it has been prepared for text classification rather than summarization. As a result, they don't contain a summary which is a must for training the system. Manually summarizing and training with such a large dataset would have needed a significant amount of time and resources.

We have utilized a different dataset XLSUM (introduced in Hasan et al., 2021) that contains summaries. This dataset includes 7,898 training, 987 development, and 987 test samples. These samples have been merged, resulting in a total of 9,872 samples. We have also used a custom dataset called SWAS[2] (Swahili Summarization) which was taken from the dataset provided by Davis, 2020. We have taken the first 1,000 texts and generated summaries with GPT-4 ensuring understandability, compactness, grammaticality, coherence, faithfulness, and saliency. SWAS dataset and XLSUM samples together have yielded a total dataset of 10,872 samples which we have used for training and evaluating our system.

The merged dataset has been shuffled. After shuffling, 1,000 samples have been split as the validation set, and the remaining 9,872 samples have been used for training.

4 System

In the GEM 2024 shared task, we participated in subtask 1 of the Summarization task, which is an unimodal task. The input text document and the generated summary both are in the Swahili language.

4.1 Data Preprocessing

During summarization, texts have been used as inputs and summaries as outputs. As both have needed preprocessing and summaries are the labels, special care has been required during preprocessing. Thus, we have used different preprocessing functions for texts and summaries. In both cases, all the uppercases have been lowered. However, the removal of punctuation, digits, and stopwords has only been applied to the texts, not to the summaries. Though NLTK[3] has been the renowned method for the resource of stopwords, it has not contained the stopwords of Swahili. So we have used 74 stopwords found in a GitHub repository.[4]

4.2 Initial Experimentation

As the provided dataset does not contain summaries, we have initially approached extractive summarization to establish the baseline for this task. Figure 1 illustrates that this system has first read and tokenized input text into sentences. Next, each sentence has been vectorized based on word frequency, with stopwords eliminated. Cosine Distance has been used to calculate sentence similarity, forming a similarity matrix based on pairwise relationships between sentences. Then the similarity matrix has been converted into a graph, considering sentences as nodes and similarity scores as edges. The PageRank algorithm (as mentioned in Xing and Ghorbani, 2004) has been used to rank the sentences based on their centrality and importance. The top-ranked sentences have been chosen to construct the summary.

Afterward, we have implemented abstractive summarization on our processed dataset. For this, we have used transformer-based models mBART-50, mT5_multilingual_XLSum, and t5. The "mBART-50" (introduced in Tang et al., 2020) has been a multilingual sequence-to-sequence model that supports 50 languages, including Swahili. We have used the "t5-small" checkpoint of the t5 model (introduced in Raffel et al., 2020), which con-

[1] www.bbc.com/swahili
[2] github.com/Samia2001/CUET_SSTM_GEM24
[3] www.nltk.org
[4] github.com/stopwords-iso/stopwords-sw

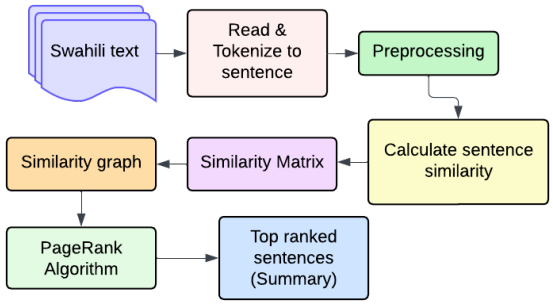

Figure 1: Initial Extractive summarization system.

tains about 60 million parameters. "mT5-multilingual-XLSum" has been an mT5 checkpoint (introduced in Hasan et al., 2021) fine-tuned on 45 languages of the XLSum dataset. We have evaluated this checkpoint to better understand the baseline scores for our dataset. Figure 2 illustrates the initial abstractive summarization system.

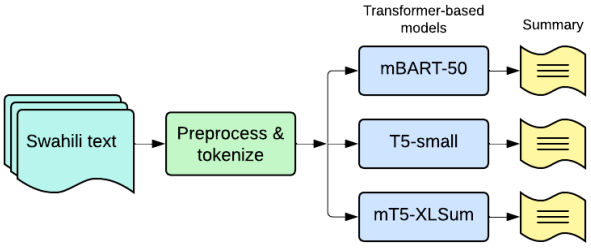

Figure 2: Initial Abstractive summarization system.

4.3 Overview of the Adopted Model

Our final adopted model has been the integration of both extractive and abstractive summarization systems. Our system takes texts of any length and outputs summaries in the Swahili language. Our dataset has contained very large texts, often exceeding 512 tokens, which has been the maximum input token limit of the transformer models. Extractive summarization has been used to shorten these very large texts (more than 512 tokens).

In this system, the input text has first been tokenized. Then, the token count has been checked. If the token count has exceeded 512, we have applied the "Bert Extractive Summarizer" tool available in Python[5] to reduce its token size to less than 512 tokens. This BERT Extractive Summarizer has ensured that the output is large enough to retain valuable information. Subsequently, we have applied a transformer-based model to produce a more precise and accurate summary. We have used the "t5-small" and "mBART-50" checkpoints with the Seq2SeqTrainer API. We haven't used the "mT5-XLSum" in the final adopted model, because this model is pre-trained on the "XL-SUM" dataset we used to train our system. We only used this model in section 4.2 to get a baseline for our task.

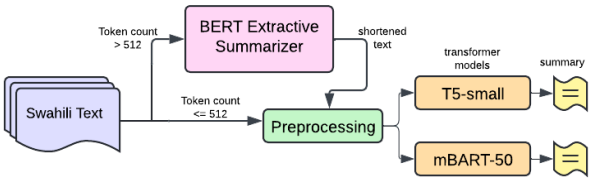

Figure 3: Integrated extractive and abstraction summarization system

5 Results and Analysis

5.1 Parameter Setting

To train the transformer model "t5-small", the parameters have been set as follows: dropout rate and attention rate have been set to 0.1, learning rate to 0.00005, training and evaluation batch size both set to 16, no weight decay and ran for 100 epoch with conditions to save the best model enabling early stopping. The patience for early stopping has been set to 3. It has run for 86 epochs before stopping. "mBART-50" benchmark has also been trained and its parameter setting was as follows: learning rate has been set to 0.00001, training and evaluation batch size both to 8, weight decay 0.1 and ran for 37 epoch.

5.2 Evaluation Metrics

There have been two types of evaluation for this task as mentioned in Mille et al., 2024. They are human evaluation and automatic evaluation. Understandability, faithfulness, saliency, grammaticality, coherence, and compactness of each generated summary have been checked in human evaluation. In the automatic evaluation, ROUGE scores (ROUGE-1 and ROUGE-2), BARTScore, and BERTScore have been evaluated.

[5] pypi.org/project/bert-extractive-summarizer

5.3 Comparative Analysis

We have evaluated our systems with the validation set of our dataset. To ensure the significance of our evaluation result, we have split our dataset into 3 subsets after shuffling named SW-A, SW-B, and SW-C. The evaluation metrics we used for this evaluation have been only ROUGE scores (ROUGE-1 as R1, ROUGE-2 as R2, and ROUGE-L as RL). Table 2 presents the ROUGE scores of the systems used in our study. It clearly shows the Integrated Extractive-Abstractive system where "t5" has been used as the abstractive model, outperformed all other systems. Though the scores are very close, but 3 validation set's score proves the significance of the statistical difference. Thus we submitted this system (named as "CUET_SSTM") for final evaluation. Compared to GPT-3.5, "CUET_SSTM" performed better in ROUGE scores, equal scores in BERTScore, and lower scores in BARTScore.

System	Val Set	ROUGE Score		
		R1	R2	RL
Extractive	SW-A	0.06	0.01	0.05
	SW-B	0.07	0.02	0.06
	SW-C	0.04	0.01	0.03
Abstractive (mBART)	SW-A	0.14	0.03	0.1
	SW-B	0.14	0.03	0.1
	SW-C	0.1	0.02	0.08
Abstractive (t5)	SW-A	0.14	0.05	0.12
	SW-B	0.13	0.03	0.13
	SW-C	0.11	0.03	0.1
Abstractive (mT5-XLSUM)	SW-A	0.11	0.04	0.1
	SW-B	0.1	0.03	0.1
	SW-C	0.09	0.03	0.1
Integrated (t5)	SW-A	**0.16**	**0.06**	**0.15**
	SW-B	0.16	0.05	0.15
	SW-C	0.15	0.05	0.14
Integrated (mBART)	SW-A	0.14	0.04	0.12
	SW-B	0.14	0.05	0.13
	SW-C	0.13	0.04	0.11

Table 1: Performance of different systems on the validation subset

System	R1	R2	BART Score	BERT Score
GPT-3.5	27.12	10.42	-6.305	71.15
CUET_SSTM	**29.33**	**15.87**	-6.791	71.15

Table 2: Performance of our Integrated extractive-abstractive (t5) system in final evaluation in GEM'24.

5.4 Discussion

Our model produced very condensed summaries, typically 1-2 lines, due to the small labeled summaries in the XLSUM dataset used for training. To maintain consistency, the dataset we have created also contains brief summaries. XLSUM's labeled summaries don't ensure six key criteria required by the shared task—understandability, compactness, grammaticality, coherence, faithfulness, and salience. However, we've ensured these qualities in our manually created dataset, but it consists of only 1,000 entries. Additionally, the dataset we have used is relatively small (10,872 training samples), and the extractive summarizer we have used relies on cosine similarity, and does not always capture the full essence of longer texts. Our future work aims to produce summaries that accurately reflect and capture the essence of the original text. We also plan to expand our manually created dataset while ensuring it meets the six key criteria mentioned above.

6 Conclusion

Swahili summarization is challenging due to limited resources and no dedicated models. We manually summarized 1,000 texts from a Swahili news classification dataset and combined them with XLSUM's Swahili data. Using an extractive-abstract method, we applied a BERT-based summarizer for length reduction, followed by an abstractive T5-small model. Our system outperformed GPT-3.5 in R1 and R2 scores and matched its BERTScore, but GPT-3.5 outperformed our model in BART-score, particularly with highly condensed summaries.

Ethics Statement

While analyzing, preprocessing, and implementing the systems, we have ensured to keep the highest ethical standards. Our contribution will impact positively the development of a more sophisticated summarization system in the Swahili language by helping mass people.

References

OHIO University OHIO Center International Studies African Studies African Languages at Ohio University Swahili Language. Swahili. https://www.ohio.edu/cis/african/languages/swahili.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. Topicrank: Graph-based topic ranking for keyphrase extraction. In *IJCNLP*.

David Davis. 2020. Swahili: News classification dataset (0.1).

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*.

Corina Florescu and Cornelia Caragea. 2017. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th annual meeting of the ACL (volume 1: long papers)*.

Udo Hahn and Inderjeet Mani. 2000. The challenges of automatic summarization. *Computer*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the ACL: ACL-IJCNLP 2021*. ACL.

Kyle Jerro. 2018. Linguistic complexity: A case study from swahili. *African linguistics on the prairie*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. 2022. Swahbert: Language model of swahili. In *Proceedings of the 2022 Conference of the North American Chapter of the ACL: Human Language Technologies*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Simon Mille, João Sedoc, Yixin Liu, Elizabeth Clark, Agnes Axelsson, Miruna-Adriana Clinciu, Yufang Hou, Saad Mahamood, Ishmael Obonyo, and Lining Zhang. 2024. The 2024 GEM shared task on multilingual data-to-text generation and summarization: Overview and preliminary results. In *Proceedings of the 17th International Conference on Natural Language Generation: Generation Challenges*, Tokyo, Japan. Association for Computational Linguistics.

Lawrence Page. 1998. The pagerank citation ranking: Bringing order to the web. technical report. *Stanford Digital Library Technologies Project, 1998*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*.

Banafsheh Rekabdar, Christos Mousas, and Bidyut Gupta. 2019. Generative adversarial network with policy gradient for text summarization. In *2019 IEEE 13th international conference on semantic computing (ICSC)*. IEEE.

Horacio Saggion and Thierry Poibeau. 2013. Automatic text summarization: Past, present and future. *Multi-source, multilingual information extraction and summarization*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Shuai Wang, Xiang Zhao, Bo Li, Bin Ge, and Daquan Tang. 2017. Integrating extractive and abstractive models for long text summarization. In *2017 IEEE international congress on big data (BigData congress)*. IEEE.

Wenpu Xing and Ali Ghorbani. 2004. Weighted pagerank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004.* IEEE.

Min Yang, Chengming Li, Ying Shen, Qingyao Wu, Zhou Zhao, and Xiaojun Chen. 2020. Hierarchical human-like deep neural networks for abstractive text summarization. *IEEE Transactions on Neural Networks and Learning Systems*.