Text Summarisation for Low-resourced Languages, A review

Gareth Reeve Edwards[0009−0009−0267−3333] and Tshephisho Joseph Sefara[0000−0002−5197−7802]

Council for Scientific and Industrial Research, Pretoria, South Africa
gedwards@csir.co.za,tsefara@csir.co.za

Abstract. Text summarisation is becoming increasingly important for humans to more quickly understand and analyse documents with large amounts of text. In this paper, we review and discuss approaches and methods used in the development of text summarisation models for low-resourced languages, specifically South African languages. We compare approaches and results to give guidance on what may be the best approach to building a sophisticated text summarisation model for South African languages. The results showed that there is one text summarisation model created for isiXhosa out of 11 South African languages, and only a few studies were done for African languages.
We recommend future work to focus on developing necessary datasets for South African languages, developing language-specific preprocessing tools such as stemmers and stop-word lists, and finally, using the developed data to build or use more sophisticated language models.

Keywords: Text summarisation · Low-resource languages · Natural language processing.

1 Introduction

Text summarisation is the task of condensing a body of text to a shorter and more concise body of text that carries the same meaning and conveys the same message as the original text [20]. Natural language processing (NLP) coupled with various machine learning algorithms has been used to perform this task computationally, with varying degrees of success [24]. There are two main approaches to text summarisation using machine learning, namely, extractive approaches [20] and abstractive approaches [19].

Extractive text summarisation refers to the methodology that summarises text by identifying important words, sentences, or paragraphs in a text using various statistical and linguistic features. These important portions of the text are then extracted from the larger body and concatenated to create the resulting summary [11]. The importance of text in an extractive summary is based on positional or frequency factors such as word/phrase frequency and location in the text. This makes extractive summarisation a simpler method compared to

abstractive summarisation in terms of both computation and conceptual understanding [10]. Extractive text summarisation techniques, unfortunately, suffer from a few drawbacks. The concatenation process of the extractive approach sometimes negatively affects the coherence of the summarised output [11]. Furthermore, important information is usually spread across sentences, and extractive summary techniques are not able to identify these dispersed points of information [10].

Abstractive text summarisation approaches attempt to gain an understanding of concepts in a body of text and then paraphrases those concepts in a clear and concise natural language [10]. Abstractive summarisation follows, more accurately, the way a human would summarise a text. Abstractive approaches are beneficial in that they produce sophisticated summaries and include additional content that enriches the resulting summary [11]. Compared to extractive approaches, abstractive summarisation approaches are more coherent and well structured.

Within each subsection (extractive and abstractive) of text summarisation, there are further, more nuanced methods that can be used within each context. Figure 1 shows a comprehensive plot of the various text summarisation methods and preprocessing methods used in the text summarisation process.

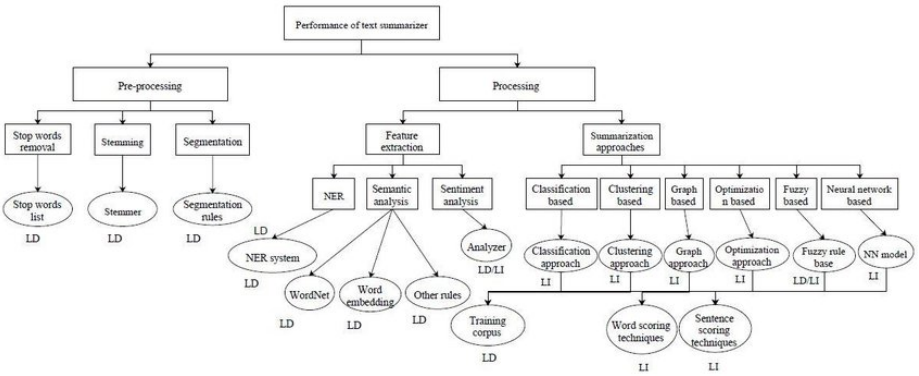

Fig. 1. Spider plot showing the various steps one can take when applying automated text summarisation where LD represents language dependent and LI represents language independent.

Both extractive and abstractive text summarisation approaches using machine learning have been extensively applied to popular world languages such as English, French, Spanish, and others. Almost all text summarisation packages today cater for these more prominent languages. Unfortunately, South African and other low-resourced languages are not well represented in the context of automatic text summarisation, as data for these languages are scarce [13].

This paper investigates current text summarisation models used in low-resourced languages and compares the approaches and results to give guidance to what may be the best approach to building a sophisticated text summarisation model for South African languages.

The remainder of the paper is outlined as follows. Section 2 discusses current available datasets for low-resource languages. Section 2 also discusses the generation and curation of datasets. Section 3 discusses various techniques applied in text summarisation for low-resource African languages. Section 4 explains the current methods of text summarisation. In Section 5, we discuss different evaluations of text summarisation models. Section 6 concludes the paper with future work and recommendations.

2 Datasets Generation and Curation

It is common knowledge that data is essential for any task related to machine learning or artificial intelligence (AI). NLP tasks are no different. Extensive research concerning automated text summarisation shows that the more sophisticated models are those trained with large datasets. Unfortunately, when it comes to low-resourced languages, there are no large datasets available to form such sophisticated models. This has come to the attention of researchers when developing text summarisation models for low-resourced languages. As a result, data creation or data curation is a highlighted task within this research context.

Data curation is essential to overcome the low-resource status of a language. One of the more notable efforts in this regard is the XL-Sum project [12]. XL-Sum is a diverse and comprehensive dataset of article-summary pairs obtained from the BBC news site. The XL-Sum dataset covers 44 different languages, ranging from low-resource to high-resource languages. XL-Sum is one of the largest abstractive datasets available. This means that abstractive text summarisers can be effectively trained on this dataset. Often low-resourced languages can only make use of rudimentary extractive techniques to summarise the text [5, 20]. The creation of large datasets for low-resourced languages allows flexibility in the text summarisation approaches one could use. XL-Sum was curated using a web scraping tool that scraped article-summary pairs from the BBC news website. The effectiveness of the dataset was tested by using an abstractive summarisation model and evaluating its performance. If the model could be trained on the data and generate effective abstractive summaries, then there is proof that the data set is of good quality for abstractive summarisation tasks. The very popular and high-performing transformer model was applied to the low-resourced languages represented in the XL-Sum dataset (e.g., Bengali, Swahili, and others.) [28]. The transformer model produced promising results in generating abstractive summaries from low-resourced languages. The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scores of abstractive summaries for low-resourced languages are shown in Table 1. ROUGE is an objective text summarisation metric to evaluate the output of an automated text summary.

The study notes that a multilingual model (mT5) performed slightly better in low-resourced languages compared to the monolingual transformer model used [29]. The benefit of multilingual models is that, if similar languages are grouped together, there is a positive transfer effect between them during the model training phase [14]. The performance of the multilingual models compared to the monolingual transformer model is shown in Table 1.

Table 1. Performance comparison between a monolingual transformer model and a multilingual transformer model applied to low-resourced languages represented in the XL-Sum dataset. The values represent rouge scores (ROUGE-1, ROUGE-2 and ROUGE-L).

Language	**Monolingual**			**Multilingual**		
	R-1	R-2	R-L	R-1	R-2	R-L
Amharic	15.33	5.12	13.85	17.49	6.34	15.76
Azerbaijani	16.79	6.94	15.36	19.29	8.20	17.62
Bengali	25.33	9.50	22.02	28.32	11.43	24.02
Japanese	44.55	21.35	34.43	47.17	23.34	36.20
Swahili	34.29	15.97	28.21	38.18	18.16	30.98

A very similar low-resourced language dataset creation / curation project called LR-Sum is ongoing [22]. LR-Sum is a multilingual data set focused on low-resourced languages. It consists of article-summary pairs for 40 different low-resourced languages. There are other similar studies completed or ongoing on the creation of multilingual data sets, namely, MILSUM [25], MultiLing [9], MassiveSumm [27], and MultiSumm [4].

A similar approach has been taken with the Indonesian language [13]. The IndoSum dataset contains roughly 20 000 Indonesian articles and their corresponding abstractive summary created by an Indonesian native speaker. The data set was tested using extractive approaches and produced promising results. However, abstractive summarisation approaches have not yet been tested on the data set.

An Igbo (a Nigerian dialect) data set was also created for use in text summarisation models [17]. Researchers found that creating a text summarisation model for Igbo is difficult given the limited online text data. They went on to create an Igbo dataset by curating article and summary pairs in English and translating them into Igbo using the Google Translate API. Unfortunately, the translations were not always accurate, so native Igbo speakers were used to rectify the translations as a preprocessing step before applying the extractive summarisation technique used in the study (TexRank and LexRank). The Igbo data set consisted of 1500 Igbo articles and their corresponding summaries.

3 Summarisation Techniques for African Languages

There are very limited resources when it comes to text summarisation for African languages and even less when it comes to native South African languages. In the section below we present research and text summarisation models used on African languages such as Xhosa, Igbo, Hausa and Afan Oromo.

3.1 Xhosa

Based on current research, it seems that research on text summarisation for South African languages is extremely scarce. A study was found that develops an extractive summariser for Xhosa, one of the most spoken languages in South Africa [20]. The extractive summarisation approach in their study was based on fairly rudimentary statistical methods developed by H. P. Edmundson in 1969. The sentence extraction technique used was based on sentence weighting. Sentences that have more document-relevant terms are weighted higher than sentences with comparatively less document-relevant words [7]. The authors curated a data set of 200 Xhosa news articles from online sources. Fifteen of the 200 articles were used to create manual summaries for model evaluation purposes. These manual summaries were created by Xhosa experts. Before any model could be applied to the data, data preprocessing was performed. The preprocessing step focused mainly on tokenization, stop-word removal, and word stemming. A stop-word list had to be created specifically for the Xhosa language, shedding further light on the low-resource nature of Xhosa (and other South African languages). A custom stemmer was also required [21].

The study used a subjective and objective approach for model evaluation. The subjective approach was carried out by handing the fifteen manual summaries and fifteen corresponding automated summaries to native Xhosa speakers and each participant was to choose the "best" summary based on three criteria:

1. Informativeness
2. Linguistic quality
3. Coherence and structure

The objective evaluation approach used the common precision, recall and f1 score metrics. Their study produced promising results, with both subjective and objective evaluations showing that the automated summary was better than the manual summary written by the Xhosa experts.

3.2 Igbo

Igbo is a popular language spoken in Nigeria and is spoken by approximately 30 million people [6]. A study has been conducted to develop an automated text summarisation model for the Igbo language. However, due to the low-resource nature of the Igbo language, the authors quickly realised that more emphasis is needed on the first step, which is creating a data set [17]. The Igbo data set was

created by obtaining English articles and summaries, then translating the texts and their corresponding summaries from English to Igbo using Google translate API. As a result, a data set of 1500 Igbo article-summary pairs was created. These summaries were used as reference summaries for later model evaluation.

The TextRank and LexRank extractive summarisation algorithms were applied to the 1500 Igbo article dataset (IgboSum1500). TextRank is a graph-based model that ranks sentences based on scores. Sentences with higher scores are considered more relevant to the document topic. These high scoring sentences would be more likely to be added to the summary [18]. These scores are based on the amount of topic-relevant words in a sentence and the positioning of the sentences in the document. LexRank is also a graph-based method. LexRank computes the importance of sentences based on the concept of centrality of the eigenvector in a graph representation of sentences [8].

The results of these extractive approaches were fairly promising, as it outperformed the base summary (which was just the title of the text) by a considerable amount. With an increase in ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L and ROUGE-S) between the base summary and the TextRank and LexRank summaries increasing by more than 50%.

3.3 Hausa

The Hausa language is another common African language. Hausa is a Chadic language widely spoken in West Africa with approximately 150 million people using it as a first or second language [3]. Hausa is also a low-resourced language as text data for Hausa is limited.

A study has been conducted to identify the best model that one can use to summarise the Hausa text [3]. The focus of the study was on a modified PageRank model that they proposed would perform better than other extractive methods. The other methods used for comparison were TextRank, LexRank, a centroid-based method, and the BM25-TextRank method (a modified TextRank method). Centroid-based methods for text summarisation are an unsupervised summarisation approach that uses word embedding techniques that help capture the semantic meaning of words [24]. The BM25-TextRank method is an alteration of TextRank that uses a probabilistic model to rank sentence importance [1]. PageRank is a graph-based model that views each sentence in a document as the vertices of an undirected weighted graph [2]. The edges of the graph were determined using word overlap between sentences. Thus, sentences with higher degree of word overlap contain higher scores and are extracted and used for the summary.

The models were applied to a Hausa dataset containing 113 Hausa news articles. In their study, it is proposed that a modified PageRank algorithm would perform better than the other models used. Their results supported this claim, as the PageRank summaries had significantly high ROUGE scores compared to their counterparts.

3.4 Afan Oromo

Afran Oromo is one of the popular languages in Africa, spoken mainly in Ethiopia, but also in Kenya and Somalia. There are approximately 37.4 million native Afan Oromo speakers in Africa [6]. A study has been conducted to automatically summarise the Afan Oromo text using an extractive text summarisation approach [5]. The extractive method used was based on rudimentary sentence weighting methods developed by Edmundson [7]. Sentences were weighted according to term frequency and sentence positioning. Sentences with more topic relevant words were weighted higher in comparison with other sentences and sentences that were positioned closer to the beginning of the document were weighted higher than sentences toward the end of the document.

Their study used only eight Afan Oromo news articles with at least more than 200 words each. The reference summary was created by four native Afan Oromo speakers. The Afan Oromo speaking participants were to extract sentences from the original document that they thought contained the most salient information. Reference summaries of 10%, 20%, 30%, and 40% of the original text length were created for each of the eight Afan Oromo news articles. The study used three different summarisers, a summariser based only on term frequency (S1), a summariser based on term frequency and sentence positioning (S2), and a summariser based on term frequency and an improved sentence position method (S3).

Each text summarisation method was applied to each of the eight articles after text preprocessing was completed for each article. The preprocessing step consisted of tokenization, stop-word removal, and stemming. The stemmer used was a lightweight custom made stemmer specific for the Afan Oromo language [26].

Each of the three summarisers was evaluated based on an objective and subjective approach. The objective approach used the common evaluation metric of precision, recall, and F-1 score based on the ROUGE-N metric. The subjective approach was done using four native Afan Oromo speakers. Each participant would receive a reference summary and an automated summary, and were asked to choose the best. If two of the four participants selected the automated summary, then the automated summary would have an informativeness score of 50%. The results show that S3 performed the best of the three approaches based on objective and subjective evaluations.

4 Model Training Methods

This section discusses different approaches to text summarisation that were applied to the above-mentioned low-resourced languages. These approaches include extractive and abstractive text summarisation. Extractive summarisation approaches work by choosing the most significant sentences from the original document. On the other hand, abstractive summarisation approaches generate a new summary from scratch, based on the main ideas of the original document.

4.1 Extractive method

(i) **Edmundson Heuristic Summarisation**: automated text summarisation has been a topic of research for a longer time than one might think. Edmundson, in 1969, expanded on an automated text summarisation algorithm created by H. P. Luhn. Luhn proposed that sentence importance in a text is based on the relevant frequency of words and the positioning of words in a sentence. Sentences with more topic-relevant words would be considered more informative than their stop-word heavy counterparts. Furthermore, words found closer to the front of a text would also be weighted higher than words farther away, since Luhn proposed that more topic-relevant words are likely to feature early on in a text [16].
Edmundson built on Luhn's initial idea and proposed that cue words (words like "important" or "significant"), title words, headings and subheadings were also beneficial for text summarisation [7]. Edmundson's method took all these factors into account and developed a scoring method to determine which sentences should be added to a text's summary. Simply put, the score would be calculated as a linear combination of weights and the proposed summarisation factors. The equation would be as follows:

$$Score = (w_1 \times P) + (w_2 \times F) + (w_3 \times C) + (w_4 \times S) \tag{1}$$

where P refers to word position, F refers to topic-relevant word frequency, C refers to cue words and S refers to stop-words (where less stop words will mean a more informative sentence). The highest scoring sentences are then used in the summary.

(ii) **TextRank**: a text summarisation algorithm that was used to apply automated summarisation for both the Igbo and Hausa language. TextRank is based on the PageRank algorithm. PageRank is a graph-based algorithm created by Google to rank web pages in order of importance. The nodes of the graph would be the web pages and the edges would be links from one page to another. The idea is that nodes with more connected links would be considered more important as they contain higher levels of network traffic [2]. The TextRank algorithm is applied in a similar way. However, instead of web pages for nodes, there are sentences with links between the sentences representing similarity scores. These similarity scores could be cosine similarity, topic-relevant word overlap or any other scoring method that calculates sentence similarity [18]. The idea is that the vertices (or sentences) with higher scores are extracted and placed in the summary. The vertex scoring equation is as follows:

$$S(V_i) = (1 - d) + d \times \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j) \tag{2}$$

Where V_i is the ith vertex of the graph, $In(V_i)$ represents the vertices pointing toward V_i, $Out(V_i)$ represents the vertices pointing out from V_i and d is a dampening factor which integrates into the model the probability of jumping from one vertex to a random vertex.

(iii) **LexRank**: is another extractive summarisation method that was developed by Erkan et al. [8] in 2004. It works by ranking sentences in a text based on their importance, which is measured by their similarity to other sentences in the text. The highest-ranked sentences are then selected to form the summary. LexRank is a more sophisticated method than the Edmundson Heuristic, and it is generally considered to be more effective. It is less susceptible to noise and is able to generate summaries that are more coherent and informative.

(iv) **Centroid-based text summarisation**: is a type of extractive summarisation method that works by selecting the sentences that are most similar to the centroid of the text [23]. The centroid of a text is a vector that represents the average meaning of all sentences in the text. To generate summaries using centroid-based text summarisation, the following steps are taken:
- The centroid of the text is calculated
- Each sentence in the text is represented as a vector
- The similarity between each sentence vector and the centroid vector is calculated
- The sentences with the highest similarity scores are selected to form the summary.

Centroid-based text summarisation is a simple and effective method for text summarisation. It is able to generate summaries that are informative and concise. However, it can be susceptible to noise, such as irrelevant or redundant sentences.

4.2 Abstractive method

(i) **Transformer model**: was developed in 2017, and since, has been the building blocks for many of the advanced large language models in use today (GPT-4, BERT, DistilBERT, BARD, T5, and others). Transformers, introduced by Vashwani et al. [28] have revolutionised how researchers view language-related tasks. This includes text summarisation. The transformer model is an improvement on the previously used sequence-to-sequence models, such as RNNs and LSTMs. The transformer method is an improvement on these models as it contains a self-attention mechanism.This means that transformers are able to capture dependencies and relationships between words in a more comprehensive and parallelised way [28].
The transformer model has shown state-of-the-art results when performing language-related tasks like language translation, text summarisation, question-answering, text generation, and others. Furthermore, due to the self-attention mechanism there is no longer need for convolutions or recurrences in the model thus significantly improving it's training time compared to sequence-to-sequence models [28].

(ii) **mT5 model**: is a large multilingual pre-trained text-to-text transformer created by Google. It is the multilingual version of the T5 model. mT5 extends its capabilities to a wide array of languages, making it a versatile and

invaluable resource for summarisation tasks in linguistically diverse and low-resource contexts. One of the key strengths of mT5 lies in its ability to perform various text-to-text tasks, allowing it to generate effective abstractive summaries. mT5 outperforms previous models in terms of linguistic fluency and content retention, crucial aspects in summarisation tasks for languages with limited data and resources [29]. mT5's remarkable zero-shot and few-shot capabilities, combined with its extensive multilingual coverage, set itself up as a state-of-the-art tool for summarisation in low-resourced languages, where data scarcity and linguistic complexity pose significant challenges.

These are just a few of the many text summarisation methods that have been developed. Each method has its own strengths and weaknesses, and the best method to use will depend on the specific task at hand.

5 Model Evaluation Methods

The text summarisation models applied to the African languages, discussed in the previous section, resulted in varying degrees of success. Some studies evaluated their models using a subjective and objective approach [5, 17, 20], while others used an objective evaluation only [3].

All the African language text summarisation models discussed generated summaries that were subjectively better, on average, compared to the human created reference summary. However, the more common metric to use to evaluate text summarisation models is the objective ROUGE-N metric [15]. ROUGE evaluates summarisation by comparing machine-generated summarisation with other (ideal) summaries created by humans [15]. The ROUGE convention consists of four different evaluation types, namely ROUGE-N, ROUGE-L, ROUGE-W, and ROUGE-S. ROUGE-N compares the number of common n-grams between the generated summary and the ideal summary. The summary with a higher number of overlapping n-grams with the ideal summary is considered better than the generated summaries with fewer overlapping n-grams. ROUGE-L measures the longest common sub-sequence (LCS). Generated summaries with LCSs (when compared to the ideal summary) will be judged as better than summaries with shorter common sub-sequences. ROUGE-W uses an LCS metric also; the only difference is that it weights consecutive word sequences higher than nonconsecutive word sequences. ROUGE-S is a skip-bigram measurement that is similar to the ROUGE-N score; however, ROUGE-S allows for nonconsecutive bigrams. For example, in the sentence, "Hi my name is Fred", an example of a bigram would be "is Fred". An example of a skip bigram would be "name Fred" or "Hi Fred".

Table 2 shows the objective results for each text summarisation model applied to African languages (Xhosa, Igbo, Hausa, and Afan Oromo). All studies used the precision, recall, and F1 score based on overlapping n-grams to objectively evaluate model performance.

The one pitfall is that not all articles explicitly defined the size of n-grams in their evaluation calculations. Table 2 shows the evaluation of each model where

ROUGE-1 and ROUGE-2 were used, an average was taken and presented in the table.

Table 2. Objective evaluation outcomes of the text summarisation models applied to the African languages; Xhosa, Igbo, Afan Oromo and Hausa.

Language	Algorithm	Avg Recall	Avg Precision	Avg F1 Score
isiXhosa	Edmundson sentence weighting	39	35	40
igboSum	TextRank	17.5	8	10
	LexRank	17.5	8	5
Afan Oromo	S1	34	34	34
	S2	47	47	47
	S3	81	81	81
Hausa	Modified PageRank	53.4	53.3	53.3

6 Conclusion

A common theme has been found in automated text summarisation for low-resourced languages, all facing the problem of lack of article-summary type data. As a result, the text summarisation models currently applied to African languages are mostly extractive approaches that use sentence extraction techniques like document-relevant word frequencies or sentences positioning. These extractive approaches do not require a training process and follow a more algorithmic way of summarising a text as opposed to a machine learning type algorithm.

Despite using only extractive summarisation techniques, the articles discussed still produce promising results. However, more sophisticated machine learning approaches are currently not available for use in most African languages, as there are not enough summary articles available. South African languages are no exception.

Abstractive text summarisation techniques using transformers [28] are currently the best performing text summarisation models around today; however, fine-tuning these models to work for South African languages becomes difficult as there are currently no datasets available for such a task.

Research needs to be implemented to form large article-summary data sets for the various low-resourced South African languages. This is the first and most important step in overcoming the "low-resource" status for most South African languages. Various approaches can be taken to develop data sets such as web scraping of news articles [12], translating current article-summary datasets from English to the required target language with the help of native speakers and Google Translate API [17], or extracting target language articles and formulating human summaries for each article [3]. This is not an easy step as data set creation could take a lot of hours if manual summaries are being formed. However, the result is large datasets that can be used by sophisticated abstractive

text summarisation models that could help further develop the native South African languages.

After the datasets have been created for a low-resourced language, the next important step is to create language-specific text preprocessing software such as stemmers for lemmatisation and stop-word lists for stop-word removal. This is a considerably easier task than dataset creation, as one would typically only need one native speaker to implement the stemming rules or give a list of the language-specific stop-words. All the articles that we discussed applied automated text summarisation to African languages, making use of a custom stemmer and stop-word list [3, 5, 17, 21].

South African languages are especially low-resourced as there is only one known study in which text summarisation is applied to a South African language [20]. Due to the limited data (only 200 Xhosa articles), the summarisation approach used was an extractive one. The further development of the Xhosa article summary data sets, together with other South African language article-summary datasets, will help to build sophisticated abstractive text summarisation models, fine-tune existing advanced language models (T5, GPT-3, BertSum, and others) or build multilingual models like the one presented by the XL-Sum project [12].

We recommend future work to focus on developing these datasets for South African languages, developing language-specific preprocessing tools such as stemmers and stop-word lists, and finally, using the developed data to build or use more sophisticated language models. Once the datasets and preprocessing tools are created, further work can go into using more sophisticated language models for text summarisation and testing the various models to identify which works best for the given language.

References

1. Barrios, F., López, F., Argerich, L., Wachenchauzer, R.: Variations of the Similarity Function of TextRank for Automated Summarization (Feb 2016). https://doi.org/10.48550/arXiv.1602.03606, http://arxiv.org/abs/1602.03606, arXiv:1602.03606 [cs]
2. Bianchini, M., Gori, M., Scarselli, F.: Inside PageRank. ACM Transactions on Internet Technology **5**(1), 92–128 (Feb 2005). https://doi.org/10.1145/1052934.1052938, https://dl.acm.org/doi/10.1145/1052934.1052938
3. Bichi, A.A., Samsudin, R., Hassan, R., Hasan, L.R.A., Rogo, A.A.: Graph-based extractive text summarization method for Hausa text. PLOS ONE **18**(5), e0285376 (May 2023). https://doi.org/10.1371/journal.pone.0285376, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0285376, publisher: Public Library of Science
4. Cao, Y., Wan, X., Yao, J., Yu, D.: MultiSumm: Towards a Unified Model for Multi-Lingual Abstractive Summarization. Proceedings of the AAAI Conference on Artificial Intelligence **34**(01), 11–18 (Apr 2020). https://doi.org/10.1609/aaai.v34i01.5328, https://ojs.aaai.org/index.php/AAAI/article/view/5328, number: 01
5. DebeleDinegde, G., Yifiru Tachbelie, M.: Afan Oromo News Text Summarizer. International Journal of Computer Applications **103**(4), 1–6 (Oct 2014). https://

doi.org/10.5120/18059-8990, http://research.ijcaonline.org/volume103/number4/pxc3898990.pdf
6. Eberhard, D., Simons, G., Fennig, C.: Ethnologue: Languages of the World, 22nd Edition. SIL International (Feb 2019)
7. Edmundson, H.P.: New Methods in Automatic Extracting. Journal of the ACM **16**(2), 264–285 (Apr 1969). https://doi.org/10.1145/321510.321519, https://dl.acm.org/doi/10.1145/321510.321519
8. Erkan, G., Radev, D.R.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research **22**, 457–479 (Dec 2004). https://doi.org/10.1613/jair.1523, https://www.jair.org/index.php/jair/article/view/10396
9. Giannakopoulos, G., Kubina, J., Conroy, J., Steinberger, J., Favre, B., Kabadjov, M., Kruschwitz, U., Poesio, M.: MultiLing 2015: Multilingual Summarization of Single and Multi-Documents, On-line Fora, and Call-center Conversations. In: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. pp. 270–274. Association for Computational Linguistics, Prague, Czech Republic (Sep 2015). https://doi.org/10.18653/v1/W15-4638, https://aclanthology.org/W15-4638
10. Gupta, V., Lehal, G.: A Survey of Text Summarization Extractive Techniques. Journal of Emerging Technologies in Web Intelligence **2** (Aug 2010). https://doi.org/10.4304/jetwi.2.3.258-268
11. Hahn, U., Mani, I.: The challenges of automatic summarization. Computer **33**(11), 29–36 (Nov 2000). https://doi.org/10.1109/2.881692, conference Name: Computer
12. Hasan, T., Bhattacharjee, A., Islam, M.S., Samin, K., Li, Y.F., Kang, Y.B., Rahman, M.S., Shahriyar, R.: XL-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages (Jun 2021), http://arxiv.org/abs/2106.13822, arXiv:2106.13822 [cs]
13. Kurniawan, K., Louvan, S.: Indosum: A New Benchmark Dataset for Indonesian Text Summarization. In: 2018 International Conference on Asian Language Processing (IALP). pp. 215–220 (Nov 2018). https://doi.org/10.1109/IALP.2018.8629109
14. Lample, G., Conneau, A.: Cross-lingual Language Model Pretraining (Jan 2019). https://doi.org/10.48550/arXiv.1901.07291, http://arxiv.org/abs/1901.07291, arXiv:1901.07291 [cs]
15. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Text Summarization Branches Out. pp. 74–81. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-1013
16. Luhn, H.P.: The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development **2**(2), 159–165 (Apr 1958). https://doi.org/10.1147/rd.22.0159, https://ieeexplore.ieee.org/abstract/document/5392672, conference Name: IBM Journal of Research and Development
17. MBONU, C.E., Chukwuneke, C.I., Paul, R.U., Ezeani, I., Onyenwe, I.: Igbosum1500-introducing the igbo text summarization dataset. In: 3rd Workshop on African Natural Language Processing (2022)
18. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-3252
19. Moratanch, N., Chitrakala, S.: A survey on abstractive text summarization. In: 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). pp. 1–7 (Mar 2016). https://doi.org/10.1109/ICCPCT.2016.7530193

20. Ndyalivana, Z., Shibeshi, Z., Botha, C.: IsiXhoSum: An Extractive Based Automatic Text Summarizer for Xhosa News Items. In: Southern Africa Telecommunication Networks and Applications (Sep 2016)
21. Nogwina, M., Shibeshi, Z., Mali, Z.: Towards Developing a Stemmer for the IsiXhosa Language. In: Southern Africa Telecommunication Networks and Applications. Boardwalk, Port Elizabeth (Aug 2014)
22. Palen-Michel, C., Lignos, C.: LR-Sum: Summarization for Less-Resourced Languages (Dec 2022). https://doi.org/10.48550/arXiv.2212.09674, http://arxiv.org/abs/2212.09674, arXiv:2212.09674 [cs]
23. Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based summarization of multiple documents. Information Processing & Management **40**(6), 919–938 (2004)
24. Rossiello, G., Basile, P., Semeraro, G.: Centroid-based Text Summarization through Compositionality of Word Embeddings. In: Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres. pp. 12–21. Association for Computational Linguistics, Valencia, Spain (Apr 2017). https://doi.org/10.18653/v1/W17-1003, https://aclanthology.org/W17-1003
25. Scialom, T., Dray, P.A., Lamprier, S., Piwowarski, B., Staiano, J.: MLSUM: The Multilingual Summarization Corpus (Apr 2020). https://doi.org/10.48550/arXiv.2004.14900, http://arxiv.org/abs/2004.14900, arXiv:2004.14900 [cs]
26. Tesfaye, D.: Designing a Stemmer for Afaan Oromo Text: A Hybrid Approach. Master's thesis, Addis Ababa University (Jun 2010), http://thesisbank.jhia.ac.ke/5785/
27. Varab, D., Schluter, N.: MassiveSumm: a very large-scale, very multilingual, news summarisation dataset. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 10150–10161. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.797, https://aclanthology.org/2021.emnlp-main.797
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
29. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer (Mar 2021). https://doi.org/10.48550/arXiv.2010.11934, http://arxiv.org/abs/2010.11934, arXiv:2010.11934 [cs]