BantuLM: Enhancing Cross-Lingual Learning in the Bantu Language Family

Naira Abdou Mohamed (✉ nabdoumohamed@insea.ac.ma)
Institut National de Statistique et d'Economie Appliquée

Imade Benelallam
Institut National de Statistique et d'Economie Appliquée

Abdessalam Bahafid
Institut National de Statistique et d'Economie Appliquée

Zakarya Erraji
Institut National de Statistique et d'Economie Appliquée

Research Article

Keywords: Cross-Lingual NLP, Bantu Language Family, BERT

Posted Date: December 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3793749/v1

Additional Declarations: No competing interests reported.

BantuLM: Enhancing Cross-Lingual Learning in the Bantu Language Family

Abdou Mohamed Naira[1,2*], Benelallam Imade[1,2], Bahafid Abdessalam[1,2], Erraji Zakarya[1]

[1*]SI2M Lab, INSEA, Madinat al Irfane, Rabat, Morocco.
[2]ToumAI Analytics, Agdal, Rabat, Morocco.

*Corresponding author(s). E-mail(s): nabdoumohamed@insea.ac.ma;
Contributing authors: i.benelallam@insea.ac.ma; a.bahafid@insea.ac.ma;
zerraji@insea.ac.ma;

Abstract

This paper outlines methods for improving Bantu languages through the application of Natural Language Processing techniques. We trained a Large Language Model known as Bidirectional Encoder Representations from Transformers for the understanding of 18 Bantu languages. More precisely, we pre-trained the model using an unsupervised corpus obtained using pseudo-labeling. This pre-training task aims to comprehend the latent structures of these languages owing to an attention mechanism that enables a deeper understanding of the context. We then conducted various experiments on five downstream tasks: Language Identification, Sentiment Analysis, News Classification, Named Entity Recognition and Text Summarization. Finally, we proposed to test the effectiveness of using multilingualism in a few closely related languages instead of leveraging a vast amount of data and multiple languages that are not necessarily related. In fact, we conducted experiments on unseen languages belonging to the Bantu family and we found that the model demonstrates better ability understanding them due to their similarities to the languages used for pre-training.

Keywords: Cross-Lingual NLP, Bantu Language Family, BERT

1 Introduction

Languages belonging to the same linguistic family consequently share several similarities manifested through common vocabularies, identical word etymologies, or other latent properties. This intuitively means that understanding a language improve the learning process of other languages within the same group. In Natural Language Processing (NLP), a similar observation can be encountered, where a model gains the ability to enhance different languages through cross-lingual learning. In [32], knowledge sharing was observed in Neural Machine Translation (NMT). In fact, the experiments led to a form of generalization by the NMT model to a language different from the one for the model training. This was due to the similarities between them.

Incorporating cross-lingual approaches could significantly enhance the cutting-edge of NLP, particularly when dealing with scenarios of data scarcity. We then want to use such methods to improve NLP for 18 Bantu languages, including low-resource dialects like Shingazidja, Shindzuani and Shimaore spoken in the Comoros Islands, related to Swahili.

In recent years, Large Language Models (LLM) such as Bidirectional Encoder Representations from Transformers (BERT) [8] have significantly played a pivotal role in enhancing language understanding, largely due to their capacity to grasp contextual nuances within documents. Intriguing experiments conducted by [27] demonstrated that the multilingual iteration of BERT, pretrained on 104 languages, possesses the capability, through a transfer learning approach, to better comprehend languages even if they utilize scripts distinct from those languages already enhanced by the model.

In this study, we employ the BERT architecture to construct our pre-trained model. During the training phase, we utilize the NLLB Dataset[1], a parallel dataset inspired by the work of [31]. This dataset is part of a substantial Neural Machine Translation initiative that enhances 200 languages, including 50 African languages. Specifically, we extract the splits relevant to our target languages and merge them for training purposes. To assess the performance of the resulting model, we conduct experiments involving three downstream tasks: Language Identification (LID), Sentiment Analysis (SA) and Named Entity Recognition (NER).

The remainder of this work is structured as follows. We begin by provide an overview of similar works (see Section 2) present in the existing literature. Then we proceed to elaborate on the language choices and offer insights into Bantu languages in Section 3. In Section 4 we describe the main contributions of this work. Subsequently, Section 5 outlines the experimental setups with the results and we discuss these latter in Section 6. Concluding our study, we encapsulate the findings and contributions in Section 7.

2 Related Works

The use of multilingual NLP solutions to enhance African languages has been experimented multiple times before. In [23], interesting results were obtained by only using low-resource data to pretrain a multilingual BERT model. They interestingly found that, in low-resource settings, having a few data provided from closely related languages could ensure the obtainment of effective results compared to mixing a massive amount of data that does not necessarily belong to languages sharing common properties. This consequently led us to limit our corpus to only Bantu languages. This approach was also defended in [3] where they shown that grapheme overlapping between close languages help considerably to enable crosslingual solutions.

One important aspect to consider regarding the scarcity of data in African languages pertains to the scarcity of labeled data, specifically. While obtaining unlabeled data might be a relatively straightforward task, acquiring labeled data presents significant challenges. Given that it's often essential for fine-tuning in downstream tasks, the necessity for a robust data labeling strategy becomes crucial. An initiative introduced in [22] strives to provide an SA dataset for 14 African languages. The effectiveness of the proposed approach is evident, yet its implementation can be complex due to its dependence on manual labeling which can be a time-consuming and resource-intensive endeavor. Thus, proposing automatic labeling approaches could offer tremendous advantages.

3 About Bantu Languages

The Bantu language family consists of 435 living languages spoken by approximately 300 million speakers [1]. As obtaining data for all of these languages poses challenges mainly due to the fact that many of them are spoken by only few speakers who may not generate sufficient data for the development of effective solutions, we have chosen to concentrate solely on 18 languages (refer to the ”LLM” column in Table 1) for the purpose of model pre-training. Given the substantial number of speakers for these selected languages, our solution has the potential to substantially contribute to the advancement of NLP technologies within these languages as well as their closely related languages.

As demonstrated in [6], Bantu languages are notably influenced by code-mixing with colonial languages. For instance, languages like Swahili and Lingala exhibit influences from English and French, respectively, to an extent where foreign words have become an integral part of these

[1] https://huggingface.co/datasets/allenai/nllb

Table 1 Bantu Languages.

Language	Examples		Models			
	Singular	Plural	LLM	LID	SA	NER
Swahili	m·tu	wa·tu	✓	✓	✓	✓
Zulu	umu·ntu	aba·ntu	✓	✓	✓	✓
Xhosa	um·ntu	aba·ntu	✓	✓	✓	✓
Kinyarwanda	umu·ntu	aba·ntu	✓	✓	✓	✓
Northern Sotho	mo·tho	ba·tho	✓	✓	✓	✗
Tswana	mo·tho	ba·tho	✓	✓	✓	✓
Chichewa	mu·nthu	a·nthu	✓	✓	✓	✓
Southern Sotho	mo·tho	ba·tho	✓	✓	✓	✗
Kongo	muu·ntu	baa·ntu	✓	✓	✓	✗
Rundi	umu·ntu	aba·ntu	✓	✓	✓	✗
Umbundu	omu·nu	oma·nu	✓	✓	✓	✗
Luganda	omu·ntu	aba·ntu	✓	✓	✓	✓
Luba-kasaï	mu·ntu	ba·ntu	✓	✓	✓	✗
Tsonga	mu·nhu	va·nhu	✓	✓	✓	✗
Tumbuka	mu·nthu	wa·nthu	✓	✓	✓	✗
Swati	umu·ntfu	ba·ntfu	✓	✓	✓	✗
Lingala	mo·to	ba·to	✓	✓	✓	✗
Shona	mu·nhu	va·nhu	✗	✓	✗	✓
Bemba	umu·ntu	aba·ntu	✗	✓	✗	✗
Shingazidja	m·ndru	wa·ndru	✗	✓	✗	✗
Shimaore	mu·tru	wa·tru	✗	✓	✗	✗
Kalanga	n·thu	ba·thu	✗	✓	✗	✗
Chokwe	mu·tu	a·tu	✗	✓	✗	✗
Kamba	mu·ndu	a·ndu	✗	✓	✗	✗
Kikuyu	mu·ndu	a·ndu	✗	✓	✗	✗
Makua	mu·tthu	a·tthu	✗	✓	✗	✗

languages. This phenomenon contributes to the interconnectedness of Bantu languages due to the shared influence of the same colonial languages. Moreover, even before considering these observations, Bantu languages share fundamental linguistic components, such as the noun class system [21, 35] (refer to Table 2) and even vocabulary. As evident in Table 1, the term ”person” has analogous translations[2] across a significant number of Bantu languages.

4 Main Contributions

4.1 Motivation

The main purpose of this work is to introduce BantuLM, a pre-trained multilingual model contributing on the representation of low-resource languages in the field of NLP especially Bantu lmanguages. Instead of processing each language separately we propose to resort to a multilingual approach that could have the ability to enhance at the same time a certain number of Bantu languages and that could also manage unseen languages belonging to the Bantu family even if these languages are not present in the pre-training phase. After this latter, we evaluate the model on five downstream tasks as shown in Figure 1.

These tasks could be divided into three subcategories : Text Classification, Token Classification and Text Generation. Test Classification

[2] We obtained these translations using our knowledge on certain languages and the Glosbe dictionary that can be reached in this url : https://fr.glosbe.com/.

Table 2 Nominal Prefixes in three Languages.

Class	Prefix	Meaning	Example
1	*mu-	person, human	mo- (lingala), mu- (swahili), umu- (zulu)
2	*ba-	plural of class 1	ba- (lingala), wa- (swahili), aba- (zulu)
3	*mu-	plant, inanimate	mo- / mu- (lingala), m- (swahili), umu- (zulu)
4	*mi-	plural of class 3	mi- (lingala), mi- (swahili), imi- (zulu)

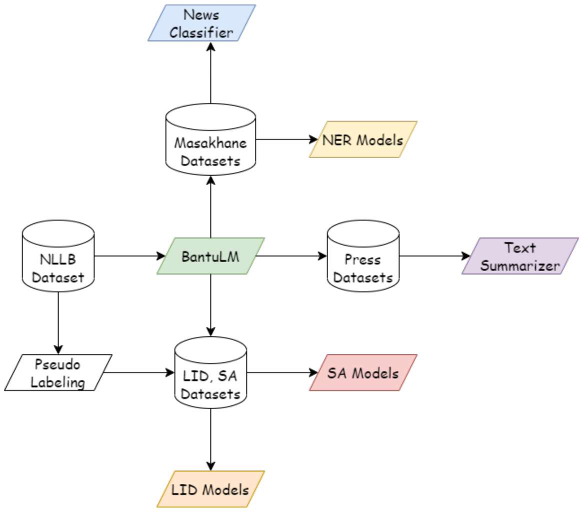

Fig. 1 Global Pipeline.

is represented by Language Identification, Sentiment Analysis and News Classification while for Token Classification we experiment on Named Entitity Recognition and on Text Summarization for Text Generation. One last thing to notive here is that this list of tasks is not exhaustive. Indee, pre-trained language model could be fine-tuned on various other tasks (Part-Of-Speech Tagging, Spelling Correction, etc.). We simply choose here to experiment on a limited number of use cases that we consider as one of the most used in NLP and depending of the ability of data to conduct the experiments.

4.2 Downstream Tasks

In this section, we elucidate the data processing methodologies that we employ following the training particulars, to assess the effectiveness of our pre-trained model. All conducted experiments were carried out on a T4 GPU within the Colab Notebook[3] environment. Due to resource constraints, we impose limitations on the size of the training corpora. For instance, in the case of LID and SA, we limit the maximum size of each training set at 200,000 sentences.

4.2.1 Language Identification

LID holds significant importance in the field of NLP, particularly in multilingual systems. In previous researches such as [24], a Naive Bayes model was employed to identify Sotho and Tswana languages, while [9] focused on 9 South African Bantu languages using Rank Order Statistics methodology. AfroLID [2], on the other hand, addresses 517 African languages, including several Bantu languages. These studies have demonstrated competitive results for their respective target languages. However, there are some limitations to consider. Some of these works exhibit challenges in terms of generalization to other Bantu languages [9, 24], while others consider non-closely related Bantu languages [2], which can dilute the specific Bantu language properties that we aim to enhance.

We extract a subset of the data used for model pre-training and append the corresponding language as a label column. To mitigate the impact of data imbalances and facilitate the fine-tuning process, we limit the languages with substantial data to 20,000 samples. However, these samples are not chosen randomly. Indeed, we retain the top samples with the highest *source_sentence_lid* and *target_sentence_lid* scores. Additionally, to enable data reuse for SA, we retain only the bitext instances where a Bantu language is translated into English or French and vice versa. For the datasets involving closely related languages, we

[3] https://colab.research.google.com/

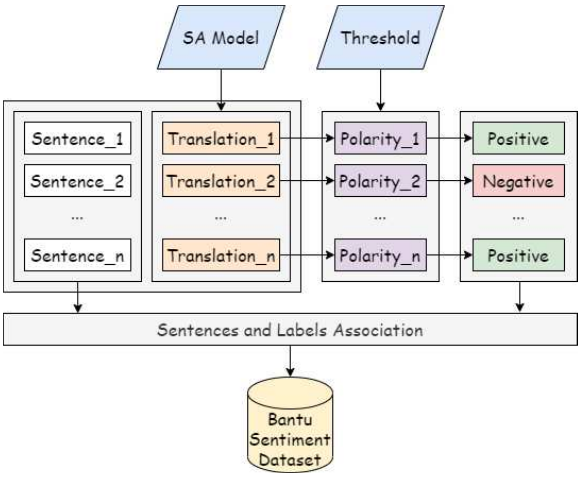

Fig. 2 Pseudo Labeling for Sentiment Analysis.

extract texts from PDF files in the JW Website[4] for the 11 closely related Bantu languages. This process yields a total of 75,000 samples distributed across these languages.

4.2.2 Sentiment Analysis

We adopt a pseudo-labeling approach, outlined in Fig. 2, for SA. Specifically, for samples where the translated sentences are in French, we employ [5], which has been fine-tuned from CamemBERT [17]. For English, we use [11], which has been fine-tuned from RoBERTa [15], an optimized BERT variant. However, not all sentences are retained for analysis. To ensure the reliability of the selected sentences, we apply a polarity threshold: only sentences with polarities less than -0.8 and greater than 0.8 are retained. The first set is labeled as Negative, while the second set is labeled as Positive.

4.2.3 News Classification

For News Classification, we employ the MasakhaNEWS dataset [4], which encompasses 16 languages, including 6 Bantu Languages: Lingala, Luganda, Rundi, Shona, Swahili and Xhosa. The dataset comprises a total of 9062 sentences. Due to the relatively low monolingual splits (refer to Table 3), we opt for multilingual experiments, hypothesizing that the lexical similarities among these languages could enhance the learning process for each language.

We conduct two multilingual scenarios (refer to Figure 3), inspired by the experiments conducted in [19] for Multilingual Speech Recognition:

- **Joint-Multilingual Scenario** : The Joint-Multilingual scenario represents a "naive approach" involving the straightforward concatenation of monolingual datasets without providing any information about the languages involved.
- **Language-Dependent Scenario** : In this approach, we augment the texts with a language tag during the data processing phase. The objective is to aid the model in recognizing the language, thereby facilitating the identification of latent properties specific to each language and ultimately improving sentence classification. During inference, we utilize the LID model described in 4.2.1.

4.2.4 Named Entity Recognition

For NER, we make use of the MasakhaNER Dataset [3], implementing several pre-processing operations. This dataset covers 20 languages, including 8 Bantu languages (see Table 1). Our approach involves the initial training of a multilingual model, followed by 8 monolingual models.

NER essentially involves token-level classification. The dataset consists of sentences, with each word annotated. Our goal is to classify words into four entity types: dates (DATE), persons (PER), organizations (ORG) and locations (LOC). The "O" tag is used for undefined tokens, while "B-" and "I-" denote the start of an entity and subsequent tokens belonging to it, respectively. This distinction is crucial as an entity may span one or multiple tokens. Table 4 presents an overview of the token distribution within the dataset.

4.2.5 Text Summarization

Text Summarization is a crucial application in NLP that focuses on distilling the most essential information from lengthy texts [33, 34]. Two primary methods are commonly used in Text Summarization: Extractive and Abstractive summarizations. The Extractive method involves selecting highly relevant sentences from the document and combining them to form the summary. This approach is widely adopted in practical systems.

[4] https://www.jw.org/fr

Table 3 Nominal Prefixes in three Languages.

Classes	Languages					
	Lingala	Luganda	Rundi	Shona	Swahili	Xhosa
business	82	169	76	500	316	72
entertainment	-	-	158	-	98	500
health	193	228	372	425	500	100
politics	500	500	500	500	493	308
religion	-	91	73	-	233	-
sports	95	116	419	417	400	496
technology	-	-	-	-	132	-

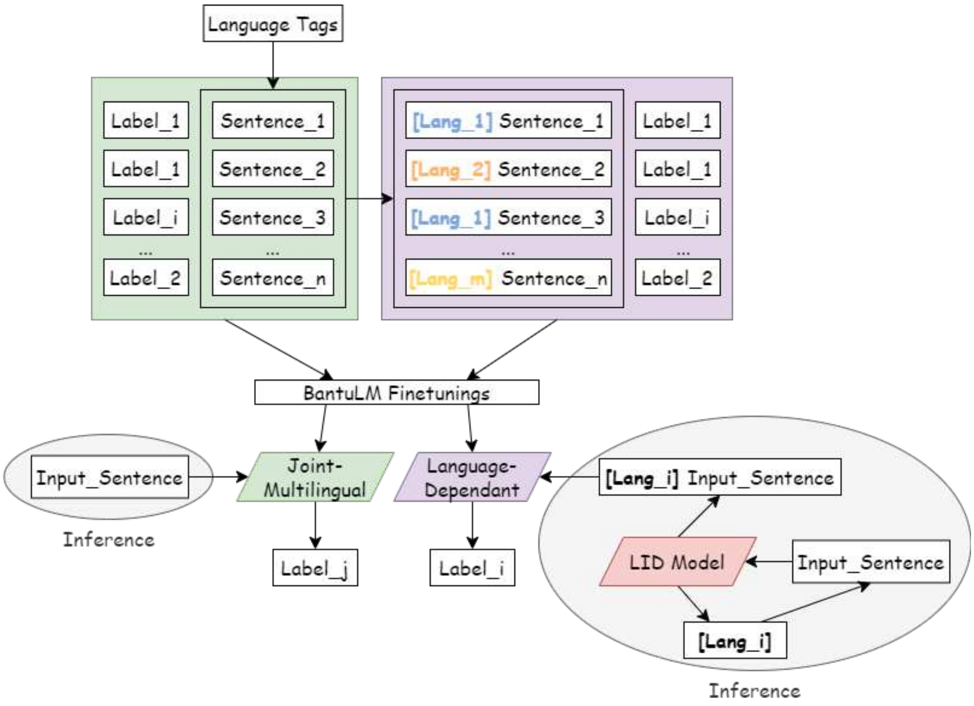

Fig. 3 Multilingual News Classification.

On the other hand, Abstractive Summarization creates a summary by generating sentences that may not be directly present in the original text. This method may involve rewriting or paraphrasing, making it the more challenging of the two approaches.

In Abstractive Summarization, the primary challenge lies in obtaining labeled data, as it necessitates documents along with their summaries for model training. Many existing works employing this method rely on press articles [10, 12, 28, 30]. This approach typically involves treating the articles and their headlines as the summaries. Despite its effectiveness, we opt not to use this approach due to the scarcity of labeled data for the languages and dialects addressed in this study. Consequently, we adopt an Extractive Summarization approach.

Our proposed architecture draws inspiration from the works in [10, 14, 29]. It follows the pipeline outlined in Figure 4 and is structured as follows:

- **Data Preparation** : In this initial step, we begin with straightforward data processing, which includes tasks such as removing URLs, hashtags, emojis and other similar elements.

Table 4 Named Entity Recognition Tokens.

B-DATE	I-DATE	B-LOC	I-LOC	B-ORG	I-ORG	B-PER	I-PER	O
15793	16569	30926	9416	26337	24726	38832	21088	1290327

Following this preprocessing, we proceed to extract sentence embeddings using BantuLM.

- **Centroid-Based Clustering** : In the second step, we initially employ the Elbow method [18] to identify the most optimal number of clusters that K-means may generate. Subsequently, we incorporate this parameter when conducting the clustering process. At the conclusion of this step, we obtain embeddings partitioned into clusters, each associated with its centroid.
- **Summary Generation** : For each embedding within a given cluster, we compute its Cosine Similarity with the centroid. We retain cases where the similarity exceeds 0.9. Finally, we concatenate the corresponding sentences that we regard as the summary.

5 Experiments

5.1 Model Pre-training

5.1.1 Dataset Preparation

The dataset utilized in this study is constructed based on the guidelines outlined in [31] involving mining bilingual text data from 200 languages. The raw data is organized as follows:

> {"*translation*": {"*eng_Latn*": "Has a nation changed its gods, even when they are not gods?", "*swh_Latn*": "Taifa wamebadili miungu yao, ingawa siyo miungu?"}, "*laser_score*": 1.25, "*source_sentence_lid*": 1, "*target_sentence_lid*": 0.99}

The *translation* value contains information about the language pair and the associated *laser_score*, which signifies the translation quality. A high LASER (Language-Agnostic SEntence Representations) score indicates a strong alignment between the two sentence pairs. Regarding the *source_sentence_lid* and *target_sentence_lid* values, higher values indicate a greater likelihood that the sentences correspond to the identified languages. During our data processing, we consider various criteria:

- **Sentence Length**: We only retain sentences that consist of at least three words. This criterion is employed because a sentence typically comprises at least a subject, a verb and a complement.
- **Metrics**: We prioritize samples with higher LID scores.
- **Cleanliness**: We avoid samples that exhibit significant noise, such as sentences containing extremely long words or words with excessive successive vowels or consonants, among other factors.

5.1.2 Pre-training Setup

The pre-training procedure is executed using the Google Cloud platform, leveraging a TPU resource obtained through the TRC[5] program for researchers. The model is composed of 12 encoder blocks, each equipped with a hidden size of 512 and 12 attention units. For optimization, we employ the Adam optimizer with a learning rate of 1e-4 and a batch size of 128.

Before initiating the processing task, we undertake tokenization using SentencePiece [13]. To avoid out-of-memory issue due to the high resource consumption of SentencePiece, we randomly select 200,000 samples to construct the vocabulary. To prevent bias, we randomize the sentences not only at the corpus level but also at the level of language-specific sentences. This entails taking approximately 11,000 random sentences for each language. During pre-training, 15% of the tokens are masked and replaced by the [MASK] token to facilitate Masked Language Modeling (MLM). The pre-training process ultimately takes around 120 hours for 150,000 steps.

5.2 Evaluation Metrics

5.2.1 Text Classification

We assess the model's classification performance using four metrics: Accuracy, F1-score, Recall

[5] https://sites.research.google/trc/about/

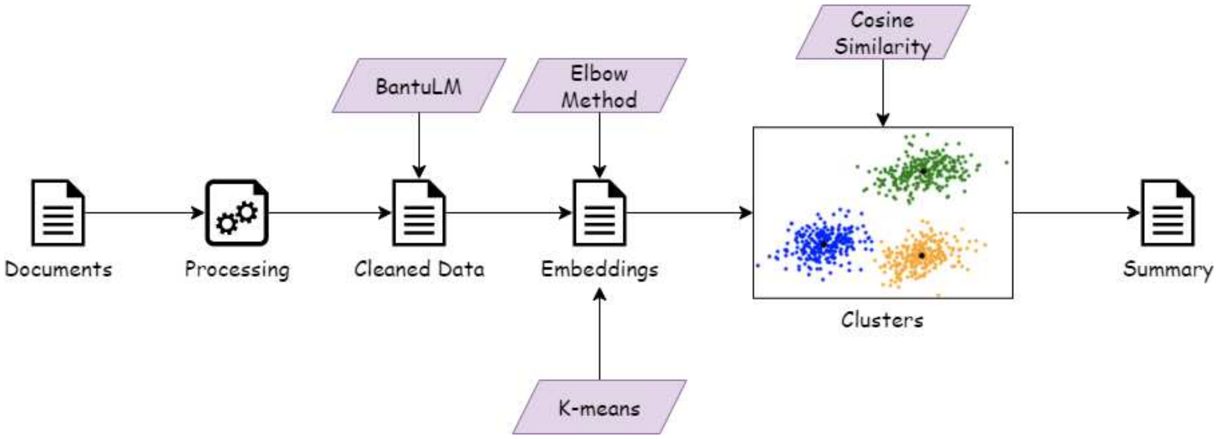

Fig. 4 Extractive Text Summarization.

and Precision. Assuming a classification problem with N classes, overall accuracy gauges the total correct predictions (true positives + true negatives) divided by the total number of observations (Equation 1).

$$Accuracy = \frac{SumofTruePositivesforallclasses}{TotalObservations} \tag{1}$$

The definitions for the three other each metrics for the $i-th$ class would be as follows:

- $Precision_i$: The Precision is calculated as the number of true positives (correctly classified positive observations) divided by the total number of observations classified as positive (true positives + false positives) (Equation 2).
- $Recall_i$: It measures the number of true positives divided by the total number of actually positive observations (true positives + false negatives) (Equation 3).
- $F1-score_i$: The F1-score is the harmonic mean of precision and recall. It is often used when there is a need to balance precision and recall (Equation 4).

$$Precision_i = \frac{TruePositives_i}{TruePositives_i + FalsePositives_i} \tag{2}$$

$$Recall_i = \frac{TruePositives_i}{TruePositives_i + FalseNegatives_i} \tag{3}$$

$$F1_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \tag{4}$$

If for the class i we have $Support_i$ as the number of samples, the weighted average metric are calculated as follows:

$$Metric = \frac{\sum_{i=1}^{N} Metric_i \times Support_i}{\sum_{i=1}^{N} Support_i} \tag{5}$$

With $Metric_i \in \{Precision_i, Recall_i, F1_i\}$ and $Metric \in \{Precision, Recall, F1\}$.

5.2.2 Text Generation

For Text Generation, we employ the BERTScore [36] metrics to calculate the similarities between the generated text and their references, as illustrated in Figure 5. In contrast to traditional text generation evaluation metrics like BLEU [25], BERTScore computes semantic equivalence instead of merely counting words and n-gram occurrences within the reference texts and the generated texts. This approach enables a better understanding of the meaning conveyed in the texts.

BERTScore also produces metrics similar to those described in Section 5.2.1. The concept is to adapt these metrics to text generation problems. The original BERTScore library[6] has limited supported models. Therefore, to evaluate the performance of BantuLM on Text Generation, we replicate the pipeline outlined in Figure 5 using BantuLM to generate the text embeddings.

[6] https://github.com/Tiiiger/bert_score

Let $x = (x_1, x_2, ..., x_k)$ and $\hat{x} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_k)$ represent a given reference text and its corresponding generated summary, respectively. To compute the BERTScore metrics, we first normalize the vector of each in order to facilitate the similarities computation. Specifically, for a given token vector x_i, the normalized vector will be $x_i = \frac{x_i}{\|x_i\|}$. The cosine similarity between x_i and $\hat{x}_j$ will simply be the scalar product of these vectors. We then follow these steps :

- **Token Representation** : Initially, we utilize the tokenizer trained during model pre-training, employing SentencePiece [13], to transform the text into a sequence of tokens with the contextualized embedding for each.
- **Similarity Measure** : We then calculate the cosine similarity between each token x_i and $\hat{x}_j$.
- **Score Computation** : A greedy matching approach is employed to identify the maximal similar pairs. Each token in the reference is matched to its most similar token in the generated text. Recall, Precision and F1-score are calculated as follows: $R_{BERT} = \frac{\sum_{x_i \in x} \max_j Sim(x_i, \hat{x}_j)}{|x|}$, $P_{BERT} = \frac{\sum_{\hat{x}_j \in \hat{x}} \max_i Sim(x_i, \hat{x}_j)}{|\hat{x}|}$ and $F_{BERT} = 2\frac{P_{BERT} \times R_{BERT}}{P_{BERT} + R_{BERT}}$.
- **Importance Weighting** : More the common words, in Text Summarization, rare words play a major roles as shown in previous works [16, 20]. Here we propose to put a more importance on words depending of their Inverse Document Frequency (idf). Suppose that we have M sentences in a text (reference or generated). We compute the idf of a token w within a given sentence with $idf(w) = \log\left(\frac{M}{df(w)}\right)$, with $df(x)$ the number of sentences in which w appears. The weighted Recall could then be computed like $R_{BERT} = \frac{\sum_{x_i \in x} idf(x_i) \max_j Sim(x_i, \hat{x}_j)}{\sum_{x_i \in x} idf(x_i)}$
- **Baseline Rescaling** : We finally normalize the score between -1 and 1. Re Recall is finally $R_{BERT} = \frac{1}{2}(R_{BERT} + 1)$

5.3 Evaluations

5.3.1 Language Identification

We assess here the performance of BantuLM against AfriBERTa and mBERT, as detailed in Table 5. The outcomes reveal an interesting observation: despite mBERT's ability to handle multiple languages, including Swahili and the fact that it was trained on a substantial amount of data, its performance is notably lower compared to our model, which is trained on a smaller dataset. This discrepancy can be attributed to the fact that the languages present in mBERT's might not be closely related to the Bantu language family. A similar rationale could possibly apply to AfriBERTa, despite its exclusive focus on African languages and its inclusion of 4 languages from the Niger-Congo language family, which encompasses the Bantu languages.

We proceed to another round of experiments involving LID on unseen close languages. The results, as displayed in Table 6, are promising for the unseen languages. However, it's important to note that while achieving good results on LID for these specific languages is a positive outcome, it may not be sufficient to definitively conclude that the model can effectively generalize to all potential downstream tasks and all Bantu languages. Further evaluation is needed to establish the model's overall adaptability and effectiveness.

Detecting close languages is quite difficult [26] due to different reasons like the phonemes and graphemes similarities between them. We focus in this second experience to only two Comorian dialects : Shingazidja and Shimaore. Owing to the high closeness of these dialects, their respective speakers do not encounter difficulties to communicate between them [7]. This means that LID solutions could have difficult to perform better on identifying these languages. But in our case, as shown in the seconds experiments, our model is able to better identify these similar dialects.

5.3.2 Sentiment Analysis

These experiments conducted in [22] utilized manually annotated tweets provided by native speakers in 14 African languages. In contrast, our approach relies on pseudo-labeled data to assess the potential for achieving significant results, particularly in cases where manual labeling is not feasible. The summarized results are shown in Table 7. It's important to note that for most of these languages, these represent some of the first SA models ever developed, to the best of our knowledge.

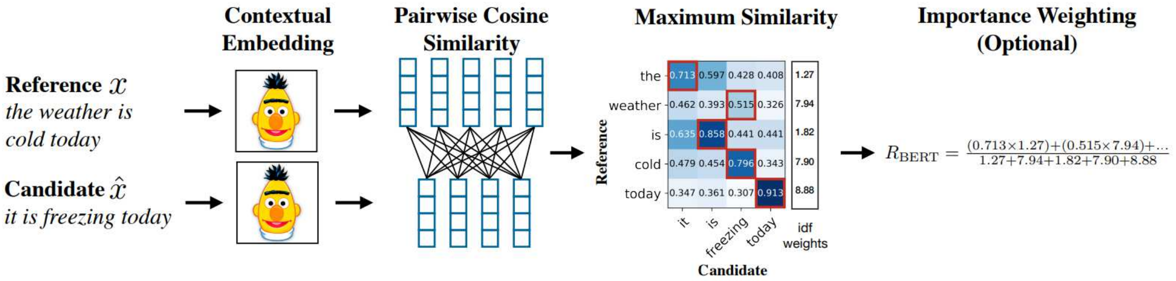

Fig. 5 BERTScore Recall metric calculation as described in [36].

Table 5 Bantu LID.

Model	Metrics			
	Accuracy	F1-score	Recall	Precision
AfriBERTa [23]	0.772012	0.758207	0.770399	0.776148
BantuLM	**0.885092**	**0.882989**	**0.882986**	**0.888356**
mBERT [8]	0.830595	0.828483	0.828369	0.840629

5.3.3 News Classification

Table 8 shows the results obtained with the two multilingual experiments for News Classification. Having high scores for the Language-Dependant approach indicate that the language tokens added to the train dataset play a crucial role in the learning procedure. Despite the efficiency of this approach, its performance relies heavily on the LID model performance. In fact, a false predicted language token could mislead the model and result to incorrect prediction.

One important thing to notice here is that, because of the fact that a model performance is highly influenced by the training data, our experiments could faced severe limitations and bias. For instance, as we can see in the Table 3, the classes and the languages are highly unbalanced. We should then except to observe interesting results for Swahili and Politics compared to the other languages and classes. The low F1-scores for the two experiments confirm the negative impact of data unbalancing to the models.

5.3.4 Named Entity Recognition

We have developed a total of 9 models, with their corresponding results presented in Table 9. The first model constitutes a language-independent NER system, while the subsequent eight models specialize in individual languages. It's noteworthy that while a monolingual model is naturally expected to excel in recognizing entities within the language it specializes in, it's interesting to observe that for some languages, the multingual model's superiority goes beyond efficiency gains (e.g., reduced inference time and storage size). In fact, a multilingual model can often exhibit significant effectiveness, particularly in NER tasks. This phenomenon can be attributed to the fact that entities like names, places and organizations are not always language-dependent, particularly when the languages share the same alphabet.

5.3.5 Text Summarization

We only assess the designed model on Swahili due to the absence of labeled data for other languages. Specifically, we conduct experiments on press articles gathered from RFI[7] and Tuko[8] Swahili news from each media source. This results in approximately 24000 articles and headlines[9], which we utilize as reference summaries. The data collection is performed through web scraping using Selenium[10].

[7] https://www.rfi.fr/sw/

[8] https://kiswahili.tuko.co.ke/

[9] The dataset is available here: https://huggingface.co/datasets/nairaxo/swahili-text-summarization

[10] https://selenium-python.readthedocs.io/

Table 6 LID on Unseen Languages.

Scope	Metrics			
	Accuracy	F1-score	Recall	Precision
All JW Data	0.908478	0.900274	0.900444	0.915489
Comorian dialects	0.963798	0.959927	0.957621	0.962378

Table 7 Sentiment Analysis Results.

Language	Length	Metrics			
		Accuracy	F1-score	Recall	Precision
Kimbundu	5032	0.704071	0.591279	0.600720	0.710907
Rundi	23172	0.678964	0.647289	0.644110	0.657094
Chichewa	62773	0.699243	0.623374	0.619539	0.664367
Kikongo	37590	0.767358	0.739842	0.734871	0.747153
Lingala	59214	0.645698	0.392355	0.500000	0.322849
Luba-Kasai	45875	0.757493	0.717720	0.712436	0.725251
Swati	6619	0.699396	0.653908	0.651856	0.656525
Northern Sotho	69434	0.694102	0.497150	0.542467	0.707626
Tsonga	19007	0.702788	0.646247	0.640406	0.667671
Kinyarwanda	**200000**	0.700100	0.698285	0.700108	0.705047
Tumbuka	17241	0.736735	0.717269	0.718398	0.716267
Umbundu	6847	0.705109	0.672884	0.671893	0.674003
Swahili	**200000**	**0.788550**	**0.787876**	**0.788393**	**0.791910**
Southern Sotho	44002	0.731508	0.683950	0.681501	0.686892
Tswana	123546	0.703642	0.656894	0.651552	0.670611

Table 8 Multilingual News Classification Results.

Approaches	Metrics			
	Accuracy	F1-score	Recall	Precision
Joint-Multilingual	0.8588	0.7377	0.7393	0.7556
Language-Dependant	**0.8709**	**0.8090**	**0.8154**	**0.8107**

Table 9 Named Entity Recognition.

Model	Length	Metrics			
		Accuracy	F1-score	Recall	Precision
Multilingual	**15404**	0.966747	0.827892	0.811618	0.847079
Kinyarwanda	4163	0.973113	0.863082	0.866652	0.863939
Luganda	3066	0.969137	0.817980	0.801221	0.873738
Chichewa	4770	0.977529	0.810543	0.804974	0.877815
Shona	8311	0.965226	0.819159	0.788598	**0.878275**
Swahili	4739	**0.981371**	**0.878474**	**0.885146**	0.873935
Tswana	1744	0.963094	0.744271	0.745633	0.749894
Xhosa	4402	0.951819	0.793964	0.784786	0.815986
Zulu	3065	0.954948	0.754533	0.718613	0.808073

Table 10 Text Results on Swahili.

F1-score	Recall	Precision
0.4450	0.4611	0.4536

Table 10 provides a summary of the BERTScore metrics computed across the test dataset. The metrics calculation is performed without the IDF weighting part described in Section 5.2.2. This is because importance weighting is more effective when working with long sequences. However, for short texts like summaries, which are typically composed of a limited number of sentences, IDF may be less significant. Therefore, we consider the raw scores without importance weighting.

The metrics suggest that the model generates summaries that are semantically close to the highlights, indicating its ability to capture essential information in long texts. However, to validate the relevance of this solution for languages other than Swahili, it would be beneficial to evaluate the model on corpora in those languages.

6 Discussion

6.1 Pre-trained Model

The solutions devised in this study may encounter certain limitations due to the nature of the data. Specifically, for the model pre-training, the data was sourced using a bitext mining algorithm. Consequently, challenges could arise from the potential presence of non-Bantu languages within the trained sentences, despite our focus on sentences with high LID scores. However, it's worth noting that during MLM tasks (as indicated in Table 11), we observe minimal generation of foreign words and a high relevance of the predicted words. This provides reassurance about the pertinence of the pre-trained model.

6.2 Downstream Tasks

When it comes to the downstream tasks for which we have trained the models, stringent measures in data processing become essential. Despite our rigorous filtering of pseudo-labeled data for SA, complete reliability is still a challenge to achieve.

Table 11 Masked Language Modeling Examples.

Language	Masked Sentence	Sentence Translation	Predicted Words	Words Translation
Luganda	[MASK] yange yamenyese	My [MASK] broke	emmeeme, emmotoka, famire, laavu, essimu	soul, car, family, love, phone
Swahili	Rais wa [MASK] ya tanzania	President of [MASK] of Tanzania	jamhuri, jamuhuri, serikali, benki, nchi	republic, republic, government, bank, country
Zulu	Ubusuku [MASK] kakhulu namuhla	Night [MASK] very much today	obuhle, obude, obukhulu, obumnyama, bonke	beautiful, tall, big, dark, all of them
Chichewa	Ndikudwala ndiyenera kupita [MASK]	I'm sick I have to go [MASK]	kuchipatala, patsogolo, kunyumba, kuntchito, kumwamba	at the hospital, ahead, at home, at work, in heaven

To gauge the effectiveness of the models on manually labeled data, we conducted experiments on the three languages for which existing datasets were available (as shown in Table 12). Our aim there was to assess whether our models perform less effectively when working with real-world data. To achieve this, we utilized the AfriSenti [22] dataset for Kinyarwanda and Tsonga, while for Swahili, we combined splits from the Swahili sentiment dataset by Neurotech[11]. Notably, all of these datasets were manually annotated by native speakers, which implies that these labels are inherently closer to the ground truth compared to our pseudo-labeled data.

What's noteworthy here is that even though there is a reduction in performance when operating on ground truth data, the models still exhibit improved performance overall. This is evident as the decrease in performance is only around 5% to 10%. To further mitigate this diminishment, various strategies can be explored and these will constitute the focus of our future endeavors to enhance performance.

Regarding the pseudo-labeling procedure, one avenue to explore involves refining the annotation process by employing more advanced and accurate models. For instance, the plan is to initially train Sentiment Analysis models in English and French using extensive amounts of data to ensure the models' robust generalization. This strategy aims to enhance the overall quality of the annotations used in the pseudo-labeling process.

7 Conclusion

In this study, we introduced BantuLM, a BERT-based Language Model tailored to address the intricacies of Bantu languages. Our contributions were centered around data construction and models training. We proposed diverse methodologies for data creation, leveraging pseudo-labeling and web scraping, which offered viable alternatives for processing low-resource languages like the ones considered in this project.

Through our conducted experiments, we observed compelling outcomes in multilingual scenarios when concentrating solely on closely related languages during pre-training. However, our intent extended beyond constructing solutions for a limited subset of languages. Our objective was to demonstrate that enhancing a group of languages within the same family can potentially facilitate the adaptation to unseen languages within it. Indeed, we achieved promising results when fine-tuning the models for Language Identification on 11 previously unseen languages. Furthermore, we extended our experimentation to four additional downstream tasks: Sentiment Analysis, News Classification, Named Entity Recognition and Text Summarization.

References

[1] Classement des langues bantoues (et autres langues bantoïdes) par nombre de locuteurs. (Accessed on 10/28/2023).

[2] I. Adebara, A. Elmadany, M. Abdul-Mageed, and A. A. Inciarte. Afrolid: A neural language identification tool for african languages, 2022.

[3] D. Adelani, G. Neubig, S. Ruder, S. Rijhwani, M. Beukman, C. Palen-Michel, C. Lignos, J. Alabi, S. Muhammad, P. Nabende, C. M. B. Dione, A. Bukula, R. Mabuya, B. F. P. Dossou, B. Sibanda, H. Buzaaba, J. Mukiibi, G. Kalipe, D. Mbaye, A. Taylor, F. Kabore, C. C. Emezue, A. Aremu, P. Ogayo, C. Gitau, E. Munkoh-Buabeng, V. Memdjokam Koagne, A. A. Tapo, T. Macucwa, V. Marivate, M. T. Elvis, T. Gwadabe, T. Adewumi, O. Ahia, J. Nakatumba-Nabende, N. L. Mokono, I. Ezeani, C. Chukwuneke, M. Oluwaseun Adeyemi, G. Q. Hacheme, I. Abdulmumin, O. Ogundepo, O. Yousuf, T. Moteu, and D. Klakow. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508, Abu Dhabi, United Arab Emirates, Dec. 2022. Association for Computational Linguistics.

[4] D. I. Adelani, M. Masiak, I. A. Azime, J. O. Alabi, A. L. Tonja, C. Mwase, O. Ogundepo, B. F. P. Dossou, A. Oladipo, D. Nixdorf, C. C. Emezue, S. Al-Azzawi, B. K.

[11] https://github.com/Neurotech-HQ/swahili-sentiment-analysis-dataset

Table 12 Sentiment Analysis Tests on Manually Labeled Data.

Language	Length	Metrics			
		Accuracy	F1-score	Recall	Precision
Kinyarwanda	3190	0.656113	0.624095	0.630023	0.663019
Swahili	**5152**	**0.754270**	**0.753709**	**0.760870**	**0.756784**
Tsonga	1047	0.516714	0.505597	0.550989	0.562391

Sibanda, D. David, L. Ndolela, J. Mukiibi, T. O. Ajayi, T. M. Ngoli, B. Odhiambo, A. T. Owodunni, N. C. Obiefuna, S. H. Muhammad, S. S. Abdullahi, M. G. Yigezu, T. R. Gwadabe, I. Abdulmumin, M. T. Bame, O. O. Awoyomi, I. Shode, T. A. Adelani, H. A. Kailani, A.-H. Omotayo, A. Adeeko, A. Abeeb, A. Aremu, O. Samuel, C. Siro, W. Kimotho, O. R. Ogbu, C. E. Mbonu, C. I. Chukwuneke, S. Fanijo, J. Ojo, O. F. Awosan, T. K. Guge, S. T. Sari, P. Nyatsine, F. Sidume, O. Yousuf, M. Oduwole, U. A. Kimanuka, K. P. Tshinu, T. Diko, S. Nxakama, A. T. Johar, S. Gebre, M. Mohamed, S. A. Mohamed, F. M. Hassan, M. A. Mehamed, E. Ngabire, and P. Stenetorp. Masakhanews: News topic classification for african languages. 2023.

[5] T. Blard. Theophileblard/french-sentiment-analysis-with-bert: How good is bert ? comparing bert to other state-of-the-art approaches on a french sentiment analysis dataset. https://github.com/TheophileBlard/french-sentiment-analysis-with-bert. (Accessed on 10/28/2023).

[6] E. G. Bokamba. Code-mixing, language variation, and linguistic theory:. *Lingua*, 76(1):21–62, Sept. 1988.

[7] M. A. Chamanga. Chapter 4 shikomori, the bantu language of the comoros: Status and perspectives in: Handbook of language policy and education in countries of the southern african development community (sadc). https://brill.com/display/book/9789004516724/BP000005.xml?language=en. (Accessed on 10/28/2023).

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] M. Dube and H. Suleman. Language identification for south african bantu languages using rank order statistics. In *Digital Libraries at the Crossroads of Digital Information for the Future*, pages 283–289. Springer International Publishing, 2019.

[10] K. Gaanoun, A. M. Naira, A. Allak, and I. Benelallam. *Automatic Text Summarization for Moroccan Arabic Dialect Using an Artificial Intelligence Approach*, page 158–177. Springer International Publishing, 2022.

[11] J. Hartmann, M. Heitmann, C. Siebert, and C. Schamp. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40(1):75–87, Mar. 2023.

[12] A. Issam and K. Mrini. Goud.ma: a news article dataset for summarization in moroccan darija. In *3rd Workshop on African Natural Language Processing*, 2021.

[13] T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics.

[14] S. Lamsiyah, A. El Mahdaouy, B. Espinasse, and S. El Alaoui Ouatik. An unsupervised

method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Systems with Applications*, 167:114152, Apr. 2021.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[16] K. Madatov, S. Bekchanov, and J. Vičič. Uzbek text summarization based on tf-idf, 2023.

[17] L. Martin, B. Muller, P. J. O. Suá rez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot. CamemBERT: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

[18] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono. The determination of cluster number at k-mean using elbow method and purity evaluation on headline news. In *2018 International Seminar on Application for Technology of Information and Communication*. IEEE, Sept. 2018.

[19] N. A. Mohamed, I. Benelallam, A. Allak, and K. Gaanoun. Multilingual speech recognition initiative for african languages. Mar. 2023.

[20] D. Morozovskii and S. Ramanna. Rare words in text summarization. *Natural Language Processing Journal*, 3:100014, June 2023.

[21] M. E. Morrison. Beyond derivation: Creative use of noun class prefixation for both semantic and reference tracking purposes. *Journal of Pragmatics*, 123:38–56, Jan. 2018.

[22] S. H. Muhammad, I. Abdulmumin, A. A. Ayele, N. Ousidhoum, D. I. Adelani, S. M. Yimam, I. S. Ahmad, M. Beloucif, S. M. Mohammad, S. Ruder, O. Hourrane, P. Brazdil, F. D. M. A. Ali, D. David, S. Osei, B. S. Bello, F. Ibrahim, T. Gwadabe, S. Rutunda, T. Belay, W. B. Messelle, H. B. Balcha, S. A. Chala, H. T. Gebremichael, B. Opoku, and S. Arthur. Afrisenti: A twitter sentiment analysis benchmark for african languages, 2023.

[23] K. Ogueji, Y. Zhu, and J. Lin. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.

[24] B. Okgetheng and E. A. Budu. Word-based bantu language identification using naïve bayes. In *2022 IST-Africa Conference (IST-Africa)*. IEEE, May 2022.

[25] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, ACL '02. Association for Computational Linguistics, 2001.

[26] S. Parida, E. Villatoro-Tello, S. Kumar, M. Fabien, and P. Motlicek. Detection of similar languages and dialects using deep supervised autoencoder. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 362–367, Indian Institute of Technology Patna, Patna, India, Dec. 2020. NLP Association of India (NLPAI).

[27] T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert?, 2019.

[28] M. Ravaut, S. Joty, and N. Chen. SummaReranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4504–4524, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[29] G. Rossiello, P. Basile, and G. Semeraro. Centroid-based text summarization through

compositionality of word embeddings. In G. Giannakopoulos, E. Lloret, J. M. Conroy, J. Steinberger, M. Litvak, P. Rankel, and B. Favre, editors, *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, pages 12–21, Valencia, Spain, Apr. 2017. Association for Computational Linguistics.

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

[31] N. Team, M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, A. Sun, S. Wang, G. Wenzek, A. Youngblood, B. Akula, L. Barrault, G. M. Gonzalez, P. Hansanti, J. Hoffman, S. Jarrett, K. R. Sadagopan, D. Rowe, S. Spruit, C. Tran, P. Andrews, N. F. Ayan, S. Bhosale, S. Edunov, A. Fan, C. Gao, V. Goswami, F. Guzmán, P. Koehn, A. Mourachko, C. Ropers, S. Saleem, H. Schwenk, and J. Wang. No language left behind: Scaling human-centered machine translation, 2022.

[32] A. Tebbifakhr, M. Negri, and M. Turchi. Machine-oriented NMT adaptation for zero-shot NLP tasks: Comparing the usefulness of close and distant languages. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 36–46, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics (ICCL).

[33] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, A. Affandy, and D. R. I. M. Setiadi. Review of automatic text summarization techniques & methods. *Journal of King Saud University - Computer and Information Sciences*, 34(4):1029–1046, Apr. 2022.

[34] D. Yadav, J. Desai, and A. K. Yadav. Automatic text summarization methods: A comprehensive review, 2022.

[35] B. Zawada and M. N. Ngcobo. A cognitive and corpus-linguistic re-analysis of the acquisition of the zulu noun class system. *Language Matters*, 39(2):316–331, Nov. 2008.

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert, 2020.