Advancing Language Diversity and Inclusion: Towards a Neural Network-based Spell Checker and Correction for Wolof

Thierno Ibrahima Cissé, Fatiha Sadat
Université du Québec à Montréal
Montréal, Canada
cisse.thierno_ibrahima@courrier.uqam.ca, sadat.fatiha@uqam.ca

Abstract

This paper introduces a novel approach to spell checking and correction for low-resource and under-represented languages, with a specific focus on an African language, Wolof. By leveraging the capabilities of transformer models and neural networks, we propose an efficient and practical system capable of correcting typos and improving text quality. Our proposed technique involves training a transformer model on a parallel corpus consisting of misspelled sentences and their correctly spelled counterparts, generated using a semi-automatic method. As we fine tune the model to transform misspelled text into accurate sentences, we demonstrate the immense potential of this approach to overcome the challenges faced by resource-scarce and under-represented languages in the realm of spell checking and correction. Our experimental results and evaluations exhibit promising outcomes, offering valuable insights that contribute to the ongoing endeavors aimed at enriching linguistic diversity and inclusion and thus improving digital communication accessibility for languages grappling with scarcity of resources and under-representation in the digital landscape.

Keywords: Spell check and correction, low-ressource language, Wolof, endangererd, Indigenous, parallel corpus, Transformer.

1. Introduction

In recent years, Natural Language Processing (NLP) has made impressive progress in understanding, analyzing and generating human language. Yet, most of this progress is focused on high-resource languages like English, French, and Spanish, leaving low-resource and under-represented languages with limited tools and resources for effective NLP applications. This paper aims to bridge this gap by introducing a novel approach for spell checking and correction in resource-scarce languages. Specifically, we focus on Wolof, an African spoken language that has recently sparked interest in NLP research. We also present a new dataset that can be used for word correction in Wolof. This study contributes to the overarching objective of developing inclusive and effective natural language processing (NLP) tools and resources, in alignment with the ethos of ”no language left behind”.

Wolof, a Senegambian language primarily spoken in Senegal, Gambia and Mauritania (Diouf et al., 2017), serves as an example of a low-resource language that could greatly benefit from NLP advancements. Despite having over 10 million native speakers (Eberhard et al., 2019), there is a significant lack of digital resources and computational tools for most of (if not all) African languages, among them the Wolof language. As the world increasingly connects through digital platforms, it is vital to ensure robust NLP tools are available for low-resource languages like Wolof. Providing speakers of the language with accurate and effective spell checking and correction systems can enhance linguistic accessibility and promote digital communication across diverse linguistic communities.

Developing spell checkers and correction systems for low-resource languages is difficult due to the limited availability of annotated data, morphological complexity, and the absence of well-established computational resources. Traditional methods like rule-based or dictionary-based systems may not adequately address these challenges, requiring alternative approaches. Deep learning techniques, particularly transformer models, have demonstrated immense potential in various NLP tasks lately. These techniques can learn complex language patterns and generate context-sensitive representations, making them ideal for tackling challenges associated with low-resource language spell checking and correction.

This paper presents a transformer-based model for word correction and spelling in Wolof. Our model is trained on a parallel corpus consisting of misspelled sentences and their error-free counterparts, optimizing the model to translate error-prone text into accurate sentences. Furthermore, we contribute to the advancement of NLP for the Wolof language by creating a new corpus of misspelled sentences and their error-free counterparts. This corpus serves as a benchmark and state-of-the-art in word correction and spelling for Wolof, provid-

The Fifth Workshop on Resources for African Indigenous Languages @LREC-COLING-2024 (RAIL), pages 140–151

ing a valuable resource for future research. This resource will facilitate the development of more advanced NLP tools and applications for Wolof.

The remainder of this paper is organized as follows: Section 2 reviews related work on the Wolof language and offers an overview of low-resource language spell checking and correction, as well as neural networks and transformer models in NLP. Section 3 details the methodology employed in developing our transformer-based spell checking and correction system. Section 4 presents our evaluation results, including a discussion of the system's performance. In Section 5, we examine the limitations of our system and discuss potential areas for improvement. Finally, Section 6 concludes the paper, underlines the implications of our findings and suggests future research directions.

2. Background

2.1. Wolof Language

Wolof is a language belonging to the Senegambian group within the Northern branch of the Atlantic language family, which is part of the broader Niger-Congo language family. It shares strong linguistic connections with Pulaar and Serer languages (Sapir, 1971; Doneux, 1978; Wilson, 1989). The Atlantic language family includes approximately 40 languages, with Pulaar (a dialect of Fula) being the exception, and most are spoken in regions near of the Atlantic coast of Africa. Although Wolof is fundamentally an oral language, its orthography was standardized in 1972 (Robert, 2011).

Descriptive linguistic studies of Wolof can be traced back to the colonial period (Boilat, 1858), while other researches on Wolof morphology and syntax have been conducted by Diagne (1971), Mangold (1977), Church (1981), Dialo (1981), and Ka (1981). In-depth analytical studies of Wolof syntax can be primarily found in the works of Njie (1982) and Dunigan (1994).

Wolof is mainly an aspectual language, focusing on the aspect of an action rather than its tense. This characteristic allows the imperfective marker to combine with various tense markers. The language features a rich verb system, which includes a wide array of basic verbal forms and paradigms. Notably, Mangold (1977) and Church (1981) provide systematic presentations of Wolof verbal paradigms.

In terms of literature and resources, Wolof appears in various forms, such as novels, short story collections, and poetry. However, even in Senegal, it is challenging to find materials written in Wolof. Recent efforts have been made to improve the availability of resources for Wolof speakers. In a study by Gauthier et al. (2016), researchers gathered an Automatic Speech Recognition (ASR) dataset for four African languages, including Wolof. This dataset was then used to create the first ASR system for Wolof. Another initiative was proposed by Nguer et al. (2015), who outlined the creation process for the first collaborative online Wolof dictionary. This project was part of the larger Dictionnaires Langues Africaines - Français (DiLAF) project[1], which has produced dictionaries for seven African languages, including Wolof. However, at the time of writing, all dictionaries are accessible online except for the Wolof one. More recently, Cissé and Sadat (2023) have presented a range of resources for the Wolof language, including a spell checking tool mainly grounded in the language's writing rules.

2.2. Low-Resource Language Spell Checking and Correction

Spell checking and correction for low-resource languages have been of great interest to many researchers. Early approaches often depended on rule-based systems (Armstrong et al., 1995) or statistical methods, such as noisy channel models (Kernighan et al., 1990), n-gram models (Stolcke, 2000), and hidden Markov models (Viterbi, 1967). However, these methods often require substantial linguistic knowledge and annotated data, which may be scarce or non-existent for low-resource languages.

In recent years, researchers have investigated data-driven approaches for low-resource languages, such as unsupervised learning (Soricut and Och, 2015) and bootstrapping techniques (Yarowsky et al., 2001). Some studies have also explored the use of cross-lingual transfer learning (Täckström et al., 2012) or leveraging comparable corpora (Madnani et al., 2012) to enhance spell checking and correction performance in low-resource languages. Nevertheless, these approaches may still be constrained by the availability and quality of parallel and comparable corpora.

2.3. Neural Networks in Spell Checking and Correction

The emergence of deep learning techniques, in particular transformer models (Vaswani et al., 2017) and neural networks, has had a significant impact on the NLP field. These techniques have shown immense potential in a wide range of tasks, including machine translation (Bahdanau et al., 2015), information retrieval, conversational agents, sentiment analysis (Socher et al., 2013), and text summarization (See et al., 2017).

[1] http://pagesperso.ls2n.fr/~enguehard-c/DiLAF/index.php

In the context of spell checking and correction, sequence-to-sequence models (Sutskever et al., 2014) have been employed with promising results, using an encoder-decoder architecture to map misspelled sequences to their correct counterparts (Hládek et al., 2019). Attention mechanisms (Bahdanau et al., 2015) have also been integrated into these models to enhance the alignment between input and output sequences (Garg et al., 2019).

The development of transformer models has further advanced the capabilities of neural networks in spell checking and correction. Transformer models, which rely on self-attention mechanisms, have proven effective in capturing long-range dependencies and providing more accurate context-sensitive representations (Devlin et al., 2019). Recent studies have applied transformer models, such as BERT and GPT (Radford and Narasimhan, 2018), to spelling error detection and correction (Sorokin et al., 2016), or fine-tuned them for specific low-resource languages (Al-Ghamdi et al., 2023).

3. Methodology

Our approach consists of three main steps, namely data preparation, model architecture building, and model training configuration.
Initially, we discuss the process of data acquisition and corpus annotation, which is crucial for training an effective model, especially in the context of low-resource languages. Subsequently, we delve into the architecture of the transformer model, detailing its components and design choices. Finally, we describe the training configurations, including the parameters and settings used to train the model.

3.1. Data selection and annotation process

The data acquisition and corpus annotation process encompasses two principal phases. Initially, we identified suitable sources for the corpus data, which were available in various formats (e.g., PDF, text, HTML), and subsequently carried out the extraction of content. Following this, we employed a hybrid approach, incorporating both manual and automatic annotation techniques, and conducted thorough proofreading to generate a corpus of accurately corrected sentences.

3.1.1. Data Selection

The data collection process for our Wolof spell correction study involved gathering data from various sources such as news websites[2], social media platforms[3] [4], religious websites[5], religious PDF files (Diagne, 1997), bilingual Wolof-French dictionaries (Diouf and Kenkyūjo, 2001; Cissé, 2004) and bilingual Wolof-French corpora released (Adelani et al., 2022; Costa-jussà et al., 2022).
In total, we collected 78,384 sentences for our corpus. During the collection process, we emphasized the quality and diversity of the content, ensuring that our corpus included sentences from various domains and genres.

3.1.2. Corpus annotation

First, we used Python scripts to scrape data from news websites, social media platforms, and religious websites. This process yielded 25,860 sentences from religious websites, 21,341 sentences from social media platforms, and 13,245 sentences from news websites. Next, we extracted 10,087 sentences from religious PDF files and Wolof-French bilingual dictionaries. Additionally, we used the Wolof data from the bilingual Wolof-French corpora released by Masakhane (Adelani et al., 2022) and Facebook (Costa-jussà et al., 2022; Goyal et al., 2022). The detailed statistics of each corpus, including the number of sentences, are outlined in Table 1.

Splits	Masakhane	Facebook
Train	3360	997
Dev	1506	1012
Test	1500	N/A

Table 1: Corpora statistics

All collected sentences were saved in plain text files using the UTF-8 encoding. We observed that many of the collected sentences contained lexical or grammatical errors. To create a parallel corpus of misspelled sentences and their error-free counterparts, we used a Wolof rule-based spell correction tool (Cissé and Sadat, 2023) to generate a file containing the corrected forms of the sentences. We then manually proofread the generated file to correct any remaining grammatical and lexical errors.
For sentences that were initially error-free, we introduced various typographical errors. Most of the introduced errors involve duplication, omission, transposition, or substitution of characters. Table 2 provides an overview on the typos introduced.

Once our synthetic parallel corpus was completed, we were faced with a crucial decision before embarking on the data preprocessing and model training phase, as we needed to determine the

[2] https://www.wolof-online.com

[3] https://twitter.com/SaabalN

[4] https://www.facebook.com/wolofakxamle

[5] http://biblewolof.com

Initial word	Typo category	Misspelled word
Waxtu	Duplication	Waxxtu
Bunt	Omission	Bnt
Juddu	Transposition	udJdu
Ñëw	Substitution	Gneuw
Xaar	Substitution + Omission	Khare
Jàppale	Substitution + Omission	Diapalé
Caabi	Substitution + Omission	Thiabi
Sàkk	Substitution	Spkk

Table 2: Types of errors

atomic linguistic unit that the model will operate on. A substantial number of NLP models have traditionally used tokens as their smallest unit. However, an emerging trend has been noted towards the use of subword units (Sennrich et al., 2016b) as the fundamental building blocks.

The notion of using words as inputs to our model initially appears to be a logical default strategy, mirroring the approach observed in numerous NLP models. However, when applied to spell correction, the token approach can become overly complicated, owing to potential inaccuracies stemming from punctuation use. Additionally, the necessity for NLP models to function on a fixed vocabulary implies that our spell correction tool's vocabulary would need to be comprehensive enough to include every single possible misspelling of every single word encountered during the training process. The implications of this requirement would result in a costly model, both in terms of training and maintenance.

In consideration of these factors, we have decided to use the character as the fundamental building block for our spell checker. This approach has proven to be very effective in translation tasks by Lee et al. (2017). The adoption of character-level segmentation also allows us to preserve a manageable vocabulary size.

For experimental purposes, the overall dataset is divided into three subsets: a training set, a validation set and a test set. We randomly selected 10% of the generated corpus to form the validation and test sets. This was done to make sure that these sets accurately represent the entire dataset. The leftover 90% of the data was then used to create our training set.

3.2. Model architecture

In this study, we employed a customized Transformer model architecture (Vaswani et al., 2017) for the task of Wolof spell correction. The Transformer model has demonstrated remarkable success in various natural language processing tasks by leveraging self-attention mechanisms, which allow it to efficiently process input sequences without the need for recurrent or convolutional layers.

Our model consists of two components: an encoder and a decoder, each comprising five identical layers (Biljon et al., 2020). The encoder's primary task is to manage the input sequences containing misspellings, while the decoder focuses on producing output sequences without misspellings, as illustrated in Figure 1.

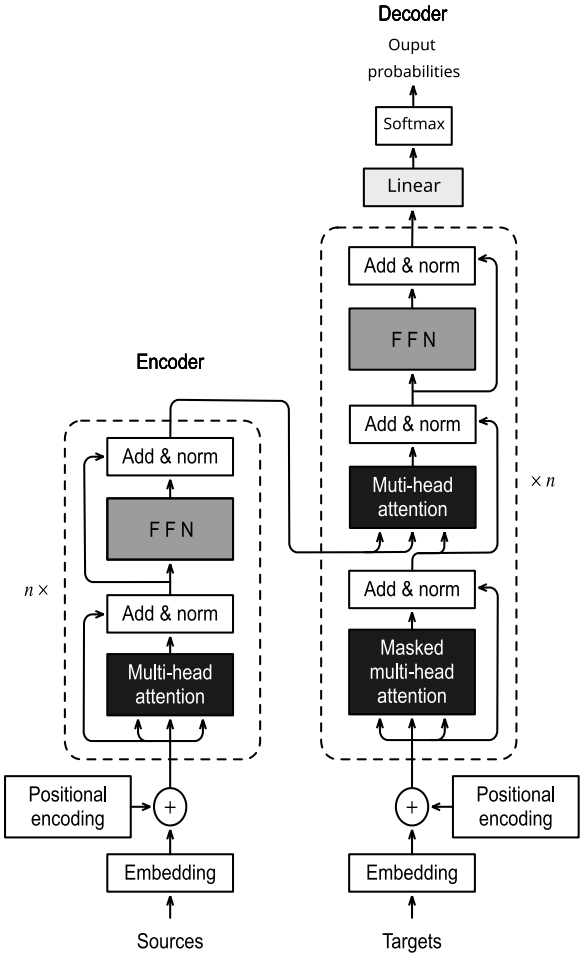

Figure 1: Transformer model(Vaswani et al., 2017)

During the encoding phase, each input word is converted into a vector representation using an embedding layer. To incorporate positional information into the input embeddings, positional encoding is applied. In both the encoder and decoder components of the model, each layer comprises a multi-head self-attention mechanism with two attention heads. This is followed by position-wise feed-forward networks (FFNs) with a hidden size of 256 and a feed-forward size of 1024.

The self-attention process involves generating query (**Q**), key (**K**), and value (**V**) vectors from the input. These vectors are then used to compute a score matrix by performing matrix multiplication between the query and the key vector. The resulting matrix is scaled by the square root of the key vector dimension (d_k). To obtain attention weights, the score matrix is normalized using softmax, representing the importance assigned to different parts

of the input sequence. These attention weights are utilized to derive an output vector, as demonstrated in Eq. 1 (Vaswani et al., 2017). To enable efficient training and stable gradients, residual connections and layer normalization are implemented throughout the network.

$$Attention(Q, K, V) = Softmax\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \times V \quad (1)$$

The decoder includes two multi-headed attention blocks within a single layer: one for the target sequences and another for the encoder's output. The former multi-head attention is masked to prevent computing attention scores for subsequent words. The latter multi-head attention layer employs the encoder's outputs as queries and keys, while the outputs of the first multi-headed attention layer serve as values. This mechanism empowers the decoder to determine the encoder inputs that are most relevant to its generation process, thereby producing an output sequence without any misspellings. The output from the final pointwise feed-forward layer is then forwarded to a linear layer, serving as a classifier, followed by a softmax layer to generate the corrected text.

For initialization, we employ Xavier initialization with a gain of 1.0 (Glorot and Bengio, 2010) for all trainable weights, while the bias terms are initialized with zeros. The embeddings undergo Xavier initialization with a distinct gain of 1.0. To minimize the number of trainable parameters, a common practice is to tie the source and target embeddings, as well as the softmax layer (Press and Wolf, 2017). Since our model operates at the character level, the default vocabulary size is relatively small. We set the embedding dimension to 256 in both the encoder and decoder, which corresponds to the hidden size of the Feed-Forward Network (FFN) for compatibility. Furthermore, we scale the embeddings by the square root of their size.

To address the issue of overfitting, we employ dropout techniques on various Transformer components. Initially, we apply an embedding dropout rate of 10^{-1} to the encoder and decoder, which helps in dropping words from the embedding matrix (Gal and Ghahramani, 2016). Furthermore, we apply dropout only within the decoder layers at a rate of 3×10^{-1} (Srivastava et al., 2014).

3.3. Model Training

The model training procedure was carefully designed, considering various parameters to ensure rigorous and repeatable results.

We employed deterministic training by using a fixed random seed of 42. To optimize the model, we chose the widely used Adam optimizer (Kingma and Ba, 2015), which features adaptive learning rates and momentum-based parameter updates. For the first and second-order moments, we assigned beta values of $[9 \times 10^{-1}, 999 \times 10^{-3}]$, respectively. The learning rate was initialized at 10^{-4}, and a minimum threshold of 10^{-8} was set to terminate training upon convergence or near-convergence.

To optimize the learning process, we adopted a plateau-based scheduling strategy (Smith, 2017). With a patience value of 5, the learning rate was reduced by a factor of 7×10^{-1} if the validation score did not improve over five consecutive validation rounds. This dynamic adaptation of the learning rate, based on performance feedback, led to enhanced convergence and optimization.

We facilitated efficient parallel computation during training by using a batch size of 4096 tokens (Ott et al., 2018). The token-based batching approach optimized computational resources by forming batches based on the total number of tokens instead of the number of sentences.

Throughout the entire training process, we placed significant emphasis on the model's ability to generalize and perform well by conducting regular evaluations. To ensure a thorough assessment, we established validation intervals of 2000 updates, covering 50 epochs. We carefully selected this interval, considering that setting a validation frequency that is too high might not provide ample opportunities for the model to learn and improve during validation. Furthermore, excessively frequent validation could lead to extended training times and potentially prematurely terminate the training if the validation patience value is not set high enough. Thus, we decided on the mentioned interval to strike a balance.

Moreover, to enhance our ability to closely track the training process and gain comprehensive insights into the model's development, we implemented a logging frequency of 200 updates.

We implemented early stopping by minimizing our cross-entropy loss function, which is a common approach in model training. Continuously monitoring the loss function allowed us to terminate the training when a new low score was reached, effectively preventing overfitting.

To promote diverse predictions and mitigate overfitting, we incorporated two regularization techniques: label smoothing and weight decay. Specifically, we employed label smoothing with a coefficient of 10^{-1} (Szegedy et al., 2016), and weight decay at a rate of 10^{-4} (Srivastava et al., 2014). Label smoothing is a regularization method that redistributes the probability weight from reference tokens to other vocabulary tokens. By reducing the overemphasis on specific reference tokens, label smoothing fosters diversity in the model's output and helps prevent overconfidence in predictions. Weight decay, also known as L2 regularization, is

a technique used to control the complexity of the model. During training, it reduces the magnitude of the model's weights by adding a penalty term proportional to the weight values to the loss function. This regularization term encourages smaller weight values, preventing the model from overfitting the training data and improving generalization performance.

During training, our primary evaluation metric was the well-established BLEU score (Papineni et al., 2002), which measures the similarity between predicted and reference sequences. For efficient evaluation, we used a token-based batching strategy with a batch size of 1024 tokens.

To manage the length of generated sequences during decoding, we set a maximum output length of 175 tokens. Furthermore, we maintained progress monitoring and validation integrity by consistently printing three validation sentences during each validation run.

4. Evaluations

Evaluating spell-checking and correction systems is a crucial task that will help understand their effectiveness and general applicability. While there is no universally accepted standard for evaluating spellchecking and correction systems, three main methodologies have emerged. These methodologies involve classification metrics, machine translation metrics, and information retrieval metrics.

Classification metrics, such as precision, recall, and F-score, are used to assess the performance of automatic spelling correction systems (Starlander and Popescu-Belis, 2002). Machine Translation metrics, including BLEU score (Papineni et al., 2002), CER or WER (Popović and Ney, 2007), and ChrF++ (Popović, 2015, 2017), are also employed in the evaluation. Additionally, information retrieval metrics like MRR (Mangu et al., 2000) can be used.

Considering that our spell checker operates by translating a source text with errors into its most likely correct form, machine translation metrics are the most suitable for measuring our system's performance. For example, the BLEU metric has been widely used to evaluate spell-checking tools in various studies, including those conducted by researchers like Gerdjikov et al. (2013); Mitankin et al. (2014); Sariev et al. (2014). The WER metric was also used in a study by Evershed and Fitch (2014).

After training and evaluating our model on the test set, our spell checker demonstrated high proficiency in various aspects of spelling correction, as shown in Table 3.

The BLEU score, a measure of how well the corrected text matches the reference text in terms of n-gram overlap, is 83%. This high score indicates that the model is capable of producing text that closely aligns with the reference text in both lexical choice and grammatical structure.

Metrics	Scores
BLEU	0.83
WER	0.08
CER	0.03
ChrF++	0.94

Table 3: Performance measures of the spell checker

The WER of 0.08 signifies that, on average, only 8% of the words in the corrected sentences differ from the reference sentences. Similarly, the CER of 0.03 indicates that the corrected sentences have, on average, only 3% character-level differences from the reference sentences. These metrics highlight the effectiveness of the spell checker in accurately identifying and correcting errors at both the word and character levels.

Furthermore, the ChrF++ score of 94% demonstrates a high level of similarity between the corrected sentences and the reference sentences, considering various factors such as precision, recall, and character-level F-score.

4.1. Error Analysis

In addition to the performance metrics mentioned above, it is crucial to conduct a comprehensive error analysis to gain deeper insights into the behavior of our spell checker. We provide a qualitative evaluation of our model on a selection of misspelled Wolof sentences in Table 4. This table presents corrected sentences alongside their corresponding references.

Predictions	References
Ngir **ya** ma def ántalpareet	Ngir **yaa** ma def ántalpareet
Allemañe dëkk bou mag la	**Almaañ** dëkk bou mag la
Woorlu askan wi ñuy jot ci téere yi	**Wóorlu** askan wi ñuy jot ci téere yi

Table 4: Qualitative evaluation

An examination of errors on a subset of the test data has revealed three primary categories of recurring errors produced by our model.

The first group of errors revolves around the correction of long vowels in words. In the Wolof language, distinguishing between long and short vowels significantly impacts word meanings. However, our model consistently struggles to accurately determine when to substitute a short vowel with a long one, resulting in incorrect corrections.

The second group of errors is related to named entities. Named entities, which often deviate from standard Wolof writing conventions, introduce considerable confusion for the model. In some instances, the model incorrectly assumes that these

named entities are erroneous and attempts to rectify them. In other cases, when specific named entities are indeed misspelled and not part of the vocabulary, the model suggests incorrect corrections.

The third group of errors is associated with accent management. Accents play a crucial role in distinguishing and pronouncing words in Wolof. Our model consistently faces challenges when accurately identifying and reinstating missing accents in words.

These findings underscore the need for further refinement of our spell checker, particularly in addressing the complexities of vowel length, handling named entities, and preserving accents within the Wolof language. Moreover, it is essential to explore potential solutions for mitigating these recurring errors, such as incorporating contextual language comprehension and enhancing the model's ability to discern linguistic nuances.

4.2. Test of significance

To establish the statistical significance of the results derived from our evaluation of the spell checker, we conducted a significance test, comparing our model against the sole existing Wolof spell checker[6] accessible online. The objective of this test is to determine the robustness of the observed performance metrics, ensuring that they are not merely a product of random chance.

Our initial step involved the random selection of 100 Wolof sentences from our constructed corpus. Following this preliminary stage, each chosen sentence was input into both correction systems to observe and analyze the proposed corrections.

Subsequently, the correction proposals generated by both systems underwent evaluation by a native Wolof speaker. The evaluator was kept unaware of the source of each correction to maintain impartiality. The applied grading system was as follows: a score of "3" was assigned to sentences that were perfectly corrected and aligned with the reference sentence. A score of "2" was reserved for corrections that, despite minor errors, preserved the original sentence's intended meaning. Lastly, a score of "1" was given to corrections that were entirely incorrect or inadequate.

In order to summarize the evaluations conducted on all the sentences, we have compiled the results in Table 5, which offers an overview of the distribution of scores attributed to each system.

Given the ordinal nature of the evaluations, we opted for the Wilcoxon signed-rank test as the most appropriate statistical tool to discern whether a statistically significant difference exists between the two systems.

Grade	Existing system	Proposed system
1	36/100 (36%)	0/100 (0%)
2	51/100 (51%)	54/100 (54%)
3	13/100 (13%)	46/100 (46%)

Table 5: Systems grades

For this test, we formulated the following null and alternative hypotheses:

$$\begin{cases} H_0 & \text{: There is no significant difference between the two systems.} \\ H_a & \text{: The neural model is significantly superior.} \end{cases}$$

The Wilcoxon test, initially introduced by Wilcoxon (1945), represents a non-parametric approach widely employed for comparing two paired samples. This method is particularly useful when assumptions regarding data distribution are not met or when dealing with ordinal data. We adopted a standard significance level ($\alpha = 0.05$) for this test, considering a result to be statistically significant if the p-value falls below α. In accordance with this methodology, the results obtained for the W-statistic and the p-value are documented in Table 6.

Metrics	Scores
W-Statistic	0.0
p-Value	4.92×10^{-17}

Table 6: W-Statistic and p-Value

The W-statistic serves as an indicator of the cumulative ranks assigned to differences between paired observations, favoring our neural model. A W-statistic value of 0.0 signifies that, in the majority of the compared instances, our proposed neural system has exhibited superior performance when contrasted with the existing rule-based system.

The p-value reflects the likelihood of obtaining such a pronounced difference between the two systems purely by chance, assuming the null hypothesis to be valid. In the context of our Wilcoxon signed-rank test, the null hypothesis postulates that there is no significant difference in the performance of the two systems. An extremely low p-value, such as the one calculated (4.92×10^{-17}), provides compelling evidence against this null hypothesis (H_0), thereby reinforcing the validity of our alternative hypothesis (H_a).

5. Limitations

Our spell checking system has demonstrated good performance, as indicated by its high BLEU and

[6] https://github.com/TiDev00/Wolof_SpellChecker

ChrF++ scores, as well as the relatively low WER and CER scores. However, there are still limitations that require further investigations and improvements.

Firstly, character-level models, such as the one used in this study, are inherently complex and can be time consuming to train. This is due to the larger sequence of data they need to learn from, compared to word-level models. The computational cost of training such models can be particularly high when working with large datasets or languages with extensive character sets.

Secondly, our model may struggle with capturing long-range dependencies within the text. The dependencies between words in a sentence, which often span across several characters, can be difficult for character-level models to understand. This could potentially affect the model's performance in cases that require a deep understanding of sentence-level semantics.

Thirdly, our model lacks the advantage of leveraging pre-trained word embeddings, which capture semantic and syntactic relationships between words. As a result, the model's semantic understanding may be less nuanced compared to models that operate at the word level.

Fourthly, character-level models can be more sensitive to noise in the input data. Spelling errors, inconsistent punctuation usage, and other forms of noise can have a more significant impact on these models, which could lead to lower performance in certain situations.

Additionally, while our model is designed to handle any language that utilizes an alphabet similar to that of the Wolof language, it may struggle with languages that rely heavily on word order. This is due to the model's lack of word-level understanding, which could help in these situations.

Lastly, our model may face difficulties with disambiguation. For instance, words spelled the same but with different meanings can pose a challenge for character-level models, as these models lack access to word-level semantic information.

Given these considerations, there are several areas that could be targeted for improvement. Firstly, the model could be further trained on a wider variety of textual data in order to improve its capacity to handle of less common or more complex errors. Given our current focus on a language with limited available resources, the use of the back-translation technique emerges as a promising strategy. This approach has consistently demonstrated its effectiveness in various domains, such as Statistical Machine Translation (SMT) (Bojar and Tamchyna, 2011), supervised Neural Machine Translation (Sennrich et al., 2016a), and unsupervised Machine Translation (Lample et al., 2017).
In the context of spell-checking and correction, adopting this approach would involve training a model to intentionally introduce a substantial number of realistic spelling errors within clean text. Subsequently, the resulting corpus of corrupted text can be employed to refine our spell checking model.

Furthermore, we suggest further exploration of hybrid models that combine the benefits of both character-level and word-level processing. Such models could potentially leverage the granularity of character-level models while still maintaining a higher-level understanding of word and sentence semantics.

Lastly, considering the computational expenses associated with character-level models, it would be beneficial to conduct research on more efficient training methods. By doing so, we can mitigate the computational burden and improve the overall efficiency of the training process.

6. Conclusion

The present study represents significant progress in the field of automatic spelling correction, particularly for under-resourced and under-represented spoken languages. Our model, which utilizes a transformer-based architecture has produced encouraging results across several evaluation metrics, including BLEU, WER, CER and ChrF++. These outcomes highlight the potential of advanced deep learning techniques to overcome the challenge of spelling errors, even in languages with limited available data.

Despite these promising results, our work has also highlighted certain areas of improvement that could further refine the performance of the proposed system. Our model, being character-level, exhibits certain limitations such as computational complexity, difficulty in capturing long-range dependencies, and sensitivity to noise in the input data. Moreover, the lack of word-level understanding could lead to potential difficulties with languages that heavily rely on word order or face challenges with disambiguation. Furthermore, the investigation of hybrid models, combining the benefits of both character-level and word-level processing, could be a promising direction for future work.

We hope that our findings will encourage further research in this direction, ultimately contributing to the broader goal of building inclusive and effective natural language processing tools for all languages.

7. Bibliographical References

David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana

Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, Andiswa Bukula, and Sam Manthalu. 2022. A few thousand translations go a long way! Leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

Sharefah Al-Ghamdi, Hend Al-Khalifa, and Abdulmalik Al-Salman. 2023. Fine-tuning BERT-Based pre-trained models for arabic dependency parsing. *Applied Sciences*, 13(7).

Susan Armstrong, Graham Russell, Dominique Petitpierre, and Gilbert Robert. 1995. An open architecture for multilingual text processing. In *From Texts to Tags: Issues in Multilingual Language Analysis. Proceedings of the ACL Sigdat Workshop.*, pages 30–34, Dublin.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Elan Van Biljon, Arnu Pretorius, and Julia Kreutzer. 2020. On optimal transformer depth for low-resource language translation. *CoRR*, abs/2004.04418.

David Boilat. 1858. *Grammaire de La Langue Woloffe*. Imprimerie impériale, Paris.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Eric Church. 1981. Le Système Verbal du Wolof. Technical report, Université de Dakar, Dakar.

Mamadou Cissé. 2004. *Dictionnaire Francais-Wolof*, 2.éd. révisée et augmentée edition. Dictionnaires des Langues O. Langues et MOndes, L'Asiatheque, Paris.

Thierno Ibrahima Cissé and Fatiha Sadat. 2023. Automatic spell checker and correction for under-represented spoken languages: Case study on Wolof. In *Proceedings of the Fourth Workshop on Resources for African Indigenous Languages (RAIL 2023)*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pathé Diagne. 1971. *Grammaire de Wolof Moderne*. Presence Africaine, Paris.

Pathé Diagne. 1997. *Al Xuraan ci Wolof*. Harmattan ; Sankoré, Paris : [Dakar].

Amadou Dialo. 1981. *Structures Verbales Du Wolof Contemporain*. Centre de Linguistique Appliquée de Dakar, Dakar.

Ibrahima Diouf, Cheikh Tidiane Ndiaye, and Ndèye Binta Dieme. 2017. Dynamique et transmission linguistique au Sénégal au cours des 25 dernières années. *Cahiers québécois de démographie*, 46(2):197–217.

Jean Léopold Diouf and Tōkyō Gaikokugo Daigaku. Ajia Afurika Gengo Bunka Kenkyūjo. 2001. *Dictionnaire wolof : wolof-français, français-wolof*. Institute for the Study of Languages and Cultures

of Asia and Africa (ILCAA), Tokyo University of Foreign Studies, Tokyo.

Jean Léonce Doneux. 1978. Les liens historiques entre les langues du Sénégal. *Réalités africaines et langues française: Bulletin du Centre de la Linguistique Appliquée de Dakar*, 7:6–55.

Melynda B. Dunigan. 1994. *On the Clausal Architecture of Wolof*. Ph.D. thesis, University of North Carolina, Chapel Hill.

David Eberhard, Gary Simons, and Chuck Fennig. 2019. *Ethnologue: Languages of the World*, 22nd edition edition. SIL International, Dallas, Texas.

John Evershed and Kent Fitch. 2014. Correcting Noisy OCR: Context Beats Confusion. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 45–51, New York, NY, USA. Association for Computing Machinery.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1027–1035, Red Hook, NY, USA. Curran Associates Inc.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Elodie Gauthier, Laurent Besacier, Sylvie Voisin, Michael Melese, and Uriel Pascal Elingui. 2016. Collecting Resources in Sub-Saharan African Languages for Automatic Speech Recognition: A Case Study of Wolof. *International Conference on Language Resources and Evaluation*, pages 3863–3867.

Stefan Gerdjikov, Petar Mitankin, and Vladislav Nenchev. 2013. Realization of common statistical methods in computational linguistics with functional automata. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 294–301, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Daniel Hládek, Matúš Pleva, Ján Staš, and Yuan-Fu Liao. 2019. Sequence to sequence convolutional neural network for automatic spelling correction. In *Proceedings of the 31st Conference on Computational Linguistics and Speech Processing (ROCLING 2019)*, pages 102–111, New Taipei City, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Omar Ka. 1981. *La Derivation et La Composition En Wolof*, volume 77 of *Les Langues Nationales Au Senegal*. Centre de Linguistique Appliquée de Dakar, Dakar.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale. 1990. A spelling correction program based on a noisy channel model. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 2*, COLING '90, pages 205–210, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Nitin Madnani, Michael Heilman, Joel Tetreault, and Martin Chodorow. 2012. Identifying high-level organizational elements in argumentative discourse. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 20–28, Montréal, Canada. Association for Computational Linguistics.

Max Mangold. 1977. *Wolof Pronoun Verb Patterns and Paradigms*. Number Bd. 3 in Forschungen Zur Anthropologie Und Religionsgeschichte. Homo et Religio, Saarbrücken.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *CoRR*, cs.CL/0010012.

Petar Mitankin, Stefan Gerdjikov, and Stoyan Mihov. 2014. An Approach to Unsupervised Historical Text Normalisation. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH '14, pages 29–34, New York, NY, USA. Association for Computing Machinery.

El Hadji Mamadou Nguer, Mouhamadou Khoule, Mouhamad Ndiankho Thiam, Mbaye Baba Thiam, Ousmane Thiare, Mame-Thierno Cissé, and Mathieu Mangeot. 2015. Dictionnaires wolof en ligne : état de l'art et perspectives. In *Colloque National Sur La Recherche En Informatique et Ses Applications*, Thiès, Senegal.

Codu Mbassy Njie. 1982. *Description Syntaxique Du Wolof de Gambie*. Les Nouvelles Editions Africaines, Dakar.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: Character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: Words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Maja Popović and Hermann Ney. 2007. Word error rates: Decomposition over POS classes and applications for error analysis. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 48–55, Prague, Czech Republic. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Stéphane Robert. 2011. Le wolof. *Bulletin de la Société de Linguistique de Paris*, 81.

J. David Sapir. 1971. West Atlantic: An inventory of the languages, their noun class systems and consonant alternation. In Thomas Albert Sebeok, editor, *Current Trends in Linguistics, 7: Linguistics in Sub-Saharan Africa*, number 7 in Current Trends in Linguistics 7 (Ed. by T. Sebeok), pages 45–112. Mouton & Co., The Hague & Paris.

Andrey Sariev, Vladislav Nenchev, Stefan Gerdjikov, Petar Mitankin, Hristo Ganchev, Stoyan Mihov, and Tinko Tinchev. 2014. Flexible Noisy Text Correction. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 31–35, Tours, France. IEEE.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR*, abs/1704.04368.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Leslie N. Smith. 2017. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017, Santa Rosa, CA, USA, March 24-31, 2017*, pages 464–472. IEEE Computer Society.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle,

Washington, USA. Association for Computational Linguistics.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.

Alexey Sorokin, A. Baytin, I. Galinskaya, E. Rykunova, and T. Shavrina. 2016. SpellRuEval: The first competition on automatic spelling correction for Russian. In *Proceedings of the Annual International Conference "Dialogue"*, volume 15, Moscow, Russia.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Marianne Starlander and Andrei Popescu-Belis. 2002. Corpus-based evaluation of a French spelling and grammar checker. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Andreas Stolcke. 2000. Entropy-based pruning of backoff language models. *CoRR*, cs.CL/0006025.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Las Vegas, NV, USA. IEEE.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80.

William André Auquier Wilson. 1989. Atlantic. In John Bendor-Samuel, editor, *The Niger-Congo Languages: A Classification and Description of Africa's Largest Language Family*, pages 81–104. University Press of America, Lanham, MD.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, USA. Association for Computational Linguistics.