Hindawi
Scientific Programming
Volume 2023, Article ID 7625785, 10 pages
https://doi.org/10.1155/2023/7625785

Research Article

Morphology-Based Spell Checker for Dawurootsuwa Language

Dawit Tadesse Gamu[1] **and Michael Melese Woldeyohannis**[2]

[1]*Department of Computer Science, Mizan Tepi University, Mizan Teferi, Ethiopia*
[2]*School of Information Science, Addis Ababa University, Addis Ababa, Ethiopia*

Correspondence should be addressed to Dawit Tadesse Gamu; dawittadesse@mtu.edu.et

Received 16 November 2022; Revised 24 April 2023; Accepted 29 April 2023; Published 10 May 2023

Academic Editor: Sadiq Hussain

Processing of textual information by using word-processing tools is extremely increased due to the presence of misspelled or erroneous words. In order to minimize these misspelled words from digital information, different spellchecker tools are needed. A plenty of works are performed in technological favored languages like English and European languages but not for an underresourced language like Dawurootsuwa. The primary idea behind a morphology-based spellchecker is to use a dictionary lookup approach with morphological properties of the language to reduce dictionary size while also handling word inflection, derivation, and compounding. Two distinct tests were carried out in this work to evaluate the performance of a morphology-based spellchecker: error detection and error correction. The Hunspell dictionary format was utilized to construct the root words in this study, which included a total of 5,000 root words and more than 2,500 morphological rules along with 3,156 unique words for testing. The experimental result showed the overall spell error detection performance of 90.4% and the overall spell error correction performance of 79.31%. Moreover, we are working further towards developing a real word spelling checker that incorporate more numbers of language rules.

1. Introduction

Communication is the act of two or more people sharing a shared understanding through the exchange of ideas, feelings, facts, and information [1]. This affects every facet of daily life including work, home life, and social relationships. In today's business world, nobody can complete their task successfully unless they communicate effectively with their employees, clients, suppliers, and customers. The people who have mastered the skill of communication are the most popular business people in the world [2]. The use of technology in communication is increasing at a rapid pace, controlling human life conditions; the strength of technical growth can be seen in many text processing products such as Microsoft Word and Office 365 [3, 4].

Natural language processing (NLP) as integral part of artificial intelligence (AI) facilitates the communication in various ways through text, speech, and video forms using different applications [5]. Among these applications, a spellchecker is one of the most important tools for detecting and correcting spelling errors in text documents produced in certain languages [6]. Spelling error detection is concerned with detecting misspelled words in the language, whereas spelling error repair is the most probable correct word for identified misspelled words. The lack of spell checking tools may result in bad spelling in the writing system which may delay the communication between writers and readers [7]. In order to handle this issues, spelling detection and correction tools are extremely important for a wide range of core NLP applications such as text authoring, OCR, postediting, or preediting for parsing, machine translation and intelligent tutoring systems, and others [8].

NLP tools such as spell checking must be localized into native languages in order to improve the usability and accessibility of computing devices and allow people to express themselves in their original languages. Among these tools, spellchecker and correction tools are now extensively used for the technological supported language like English [9], Arabic [10], European (French and Spanish) [11, 12], and Asian (Chinese and Japanese) [13, 14]. On the contrary, underresourced African languages, which contribute around 30% of the world language, highly suffer from the lack of a spellchecker for respective languages [15]. This is especially true for Ethiopian language like Dawurootsuwa. As a result, some spellchecker tools are made for our local languages like Amharic [16, 17], Afaan-Oromo [8], Kaffi-Nonoo [18], and Tigrigna language [19] but none for the Dawurootsuwa language.

Dawurootsuwa is one of the morphological rich and complex underresourced Omotic language families spoken primarily in the Dawro zone of the SNNPR in the southwest of Ethiopia [20, 21] with an approximate number of speakers of 838,000 [15, 22]. The language is also used in the primary, secondary, and tertiary levels of education. In addition to formal education, a huge amount of electronic data is being produced every day in religious areas, government offices, and media. Furthermore, new word and concept are being added and formed by affixation and compounding from the dynamic nature of the language. For example, the Dawurootsuwa root word "ush" means "drink" in English, and the language has more than 100 forms of new words. This morphological complexity of the languages leads the users to create errors during writing different material.

The most difficult issue for Dawurootsuwa from a technical point of view is developing a spelling checker application because of morphological complexity of the language. Furthermore, due to linguistic variations, the current cutting-edge word processing programs do not provide built-in spellcheckers for all languages [23, 24]. Using the most popular spell detection and repair method, various researchers came up with distinct strategies [24, 25]. These approaches for spell error correction include edit distance, similarity key, rule-based, probabilistic, neural networks, noisy channel model, and N-gram and dictionary lookup for detection. Among these approaches, the study adopted a morphology-based approach to design and develop a Dawurootsuwa spellchecker. Besides this, a single word in Dawurootsuwa forms a complete sentence. The word "ushide'enna and" forms a multiple word in English of "he is not drinking."

Morphological complexity and richness motivated the development of a spellchecker for the Dawurootsuwa language. Therefore, there is a need to collect, preprocess, and prepare a corpus for Dawurootsuwa and to conduct an experiment on detection and correction of errors. Thus, the main aim of this study is to design and develop a morphology-based spellchecker for Dawurootsuwa that overcomes the problem of language resource.

2. Related Works

A spellchecker is a computer program or function that can detect potential misspellings in a block of text by comparing it to a database of acceptable spellings [26]. To make perfect spelling for a document, a writer must add a correct sequence of characters in a coherent fashion; otherwise, spelling errors may occur. A spelling error is a word that is misplaced for the language's rules or does not meet the language's character sequence regulation [27]. The spell checker is one of the most important plugins and foundational NLP applications for many languages, particularly morphological rich, complicated, and underresourced languages [8, 25, 28]. Spell checking research in computational linguistics has a long history dating back to 1961 when the first handwriting recognizer with spellchecker capabilities was developed [25, 29]. However, the first spellcheckers were widely available in the late 1970s, and the tools were primarily built on mainframe computers for the IBM company by a group of linguists.

Many researchers have developed and implemented numerous approaches for detecting and correcting computationally incorrect words in electronic text [30]. A recent study effort demonstrates that no word processor exists without spelling checkers and correctors; even thesaurus and grammar checkers are now considered essential components of word processors [31]. This is true for numerous languages that are significant and widely spoken and have commercial relevance languages such as English, European, Arabic, and Asian languages. Furthermore, recently, some researchers conducted research for specific languages like English, Arabic, and Indian languages. The development of mistake detection and correction for the Arabic language was performed using a language model based on the Gigaword and Al-Jazeera corpora, yielding 93.64 percent and 92.78 percent, respectively [32]. Other research works for Arabic language were conducted by Hamza et al. [33], using morphological structures of the languages. They proposed a new approach which is almost independent of as dictionary, and it uses a stemmed dictionary to reduce dictionary sizes instead of using a large dictionary. The study used 2,784 misspelled words, which contain deletion, addition, and permutation errors. They conducted experimentation and compared their approach with Levenshtein's with respect to average time correction, corrected rate words, and lexicon size. The proposed approach obtained the result of 85% for addition, 81% for deletion, and 86% for permutations with an average time of 0.10 ms.

Alongside the Arabic language, spelling and grammar check for the Punjabi language was developed by using a hybrid approach [23]. The hybrid approach works sequentially; first, it checks a spell error and then corrects it. After that, it checks a grammar error and corrects it. The researchers evaluated and developed a hybrid spell and grammar checker for Punjabi languages and got an average accuracy of 83.5%. Unlike the technological favored language, a number of attempts were also made for the underrepresented Ethiopian languages by different researchers [8, 16–18]. These languages include Amharic, Afaan-Oromo, Tigrigna, and Kaffi-Nonoo.

The Amharic spellchecker was designed and developed utilizing morphological techniques to handle spell checking difficulties for Amharic nonword mistakes and obtained 97.27% accuracy [17]. Another Amharic spellchecker was created utilizing a hybrid technique in which the author created spelling checking tools for the Amharic language. As a result, the system's overall performance in error detection and correction was 98% [16]. Similarly, a morphology-based spellchecker for the Afaan-Oromo language was constructed, with 88.62% lexical recall, 100% mistake recall, and 28.62% precision.

Besides Amharic and Afaan-Oromo languages, a Tigrigna spellchecker is also developed using a rule-based morphological analyzer and an unsupervised approach [19]. These studies experimented two approaches for Tigrigna: dictionary with a morphological-based approach and an unsupervised approach using Morfessor. The system achieved 89% of recall, 87% of precision, 88% of F-measure, and 80% of total accuracy in the dictionary with a morphological-based approach. Also, the system achieved 99% of recall, 72% of precision, 73% of accuracy, and 84% of F-measure in an unsupervised morphological-based approach using the Morfessor tool. In addition, the morphology-based approach for the Kafi Noonoo language merged the dictionaries with morphological rules [18]. The performance of the system resulted with a lexical recall of 95.91%, an error recall of 100%, and a precision of 62.76%.

However, the study work cited above shown that it is not feasible to apply it directly to our target language due to the differences in the alphabet, writing system, and morphological structure of the language. Additionally, using a spellchecker system is entirely language dependent.

3. Pros and Cons of the Morphological Approach

The morphology-based approach combines both spell error detection and correction techniques. The approach is completely language dependent and designed based on the characteristics of specific language features. Some researchers developed spelling checking tools by using these approaches. According to [8, 16–18], the morphology-based approach is good for morphologically rich languages, to handle internal inflection, derivation, and word compounding. The morphology-based approach for spell checking has many advantages. The approach is efficient, and it has the ability to reduce dictionary sizes, address the word class, and handle the possible derivation, inflection, and compounding. Furthermore, according to [18], the morphology rule developed for spell checking functionality is the corner stone to develop other NLP applications. The mere drawback of the morphology-based approach needs deep knowledge of the language and time-consuming task to incorporate the overall morphological rules of the languages.

4. Dawurootsuwa Language

Ethiopia has more than 83 registered spoken languages with over 200 dialects, but the majority of the population speaks just a small subset of them all [15]. There are 56 nations in the SNNPR region of Ethiopia, each of which has its own culture, religion, and language [20]. Dawuro, which is one of these nations and is located around 480 km southwest of Addis Ababa, has its own identity, spoken languages, clothing style, eating habits, way of life, culture, and systems for collaboration and dispute resolution mechanisms [34].

Dawurootsuwa is a member of the North Ometo cluster of Omotic languages. Gok'atsuwa and Mes'atsuwa are the two primary dialects of Dawurootsuwa spoken in the highlands and lowlands close to the Wolaytta border, respectively [21]. The language is spoken by about more than one million people. Dawurootsuwa has been made available as a medium of instruction from grades 1 through 4, as a subject matter in the secondary level of education as well as the undergraduate level of the university. In addition, the language is also used as a medium of communication in the various government agencies in the zone level. Since the language is morphological rich and complex, Section 4.1 presents the writing system of the language including consonants, vowels, and punctuation used. Similarly, Section 4.2 discusses the morphology of the Dawurootsuwa language and the challenges in Section 4.3.

Table 1: Distribution of consonants and vowels.

Types	Characters
Consonants	p b p' m w t d n l r D s z s' š t^s c j c' y k g k' h ?
Short vowel	a, e, i, o, u
Long vowel	aa, ee, ii, oo, uu

4.1. Writing System. The Dawurootsuwa language employs the Latin-based AbiChiDi alphabet made up of 35 letters [35]. Five of the thirty-five basic letters are known as vowels, five are known as double consonants, and the remaining are known as consonants. Letters with two consonants combined into them are known as double consonant letters. Unlike the English language, Dawurootsuwa follows the subject-object-verb (SOV) word order. Table 1 presents the distribution of the consonant and vowels in the Dawurootsuwa languages.

Similar to the English language, vowels both create and produce sounds. Dawurootsuwa vowels can be divided into short and long varieties consisting of a total of ten vowels. Five short vowels (a, e, i, o, and u) are written in single form, and five long vowels (aa, ee, ii, oo, and uu) represented as two vowels are joined together. Short and long vowels can convey different meanings depending on the writing system used. For instance, the words “mentsaa” means “buffalo,” while “mentsa” means “break.”

In addition, except apostrophe, punctuation marks used in both Dawurootsuwa and English languages are the same and are also used for the same purpose [35]. The apostrophe mark (‘) in English shows possession, but in Dawurootsuwa, it is used in writing to represent a “glitch.” It plays an important role in the Dawurootsuwa reading and writing system. This is used to write a word in which two vowels come together most of the time, for example, “lo’aa.”

4.2. Morphology. Dawurootsuwa is an agglutinative, purely suffixing language that uses a variety of morphemes to produce complex words [20, 21]. A single morph can represent multiple grammatical components such as tenses, cases, and gender definiteness in a single element. In addition, verb roots and stems always end in consonants. The root verbs may be monosyllabic or polysyllabic, but they do not end in a vowel. Unlike the root, most nouns end with a vowel.

The vowel disappears during suffixation. Dawurootsuwa nouns have an identifiable root, however, which end in a vowel. For example, the root word “na” means “child,” where “na’a” means “boy” and “natta” means “girl.” Moreover, the noun root word does not give meaning by itself. There are also other kinds of morphemes which are used to represent plural numbers [20]. Plural numbers can be indicated by the suffixes “-tu,” “-atu,” and “-etu” depending on phonological factors. The most common derivational suffixes for nouns are “–uwa,” “-asaa,” and “iya.”

Dawurootsuwa morphology is classified by seven cases, namely, nominative, accusative, dative, locative, commutative, ablative, and vocative [20, 21]. Nominative is used to marks the subject, and it is marked by “-i” or “-y.” In addition, for nominative nouns whose stem end in “-u,” the “u” vowel gets extended; for example, “kuttu-u” means “chicken-NOM.” Dative marks an indirect object in a sentence and are represented by “w” and “oo.” For example, “wod’iyawantt-oo” means “killers DAT.” Oblique marks suffix “-ssi,” and for example, “saakettennawa-ssi” means “for health-OQ.” The accusative case is used to mark an object, which is marked by “-a” for masculine and “-o” for feminine. For example, “kana-a” means “dog.” The directional case is used towards something by adding suffix “-kko.” For example, “Doktoriya-kko” means “to doctor-DIR.” Ablative is used for motion away from something by suffixing “-ppe” or “-appe.” For example, “Addis Ababa-ppe” means “from Addis Ababa-ABL.” The commutative case denotes the concept “with” and is marked by the suffix “-nna.” For example, “Doktoriya” means doctor” and “Dokto-riyan-nna” means “with doctor.” Perlative is used for the object through which an action goes and accepts suffix “-nna.” For example, “Maskootiya-nna” means “through the window-PER.” Vocative nouns in Dawurootsuwa are derived by suffixing the elements “a” for masculine and “-e” for feminine nouns. For example, “at-e” means “mother” and “aw-a” means “father.”

4.3. Challenges. Obtaining all of the morphological features of the Dawurootsuwa language is difficult due to the nature of the language. The majority of the words are complicated and challenging to read and write. For example, the word “Sed’d’imed’d’ee” means “be proud,” whereas “Sod’d’i-sod’d’ee” implies “fear,” and writing as well as reading these terms is difficult.

In Dawurootsuwa, there are several derivational suffixes. Some morphemes from common derivational suffixes are difficult to distinguish from derivational suffixes. For instance, the derivational suffix “asaa” is sometimes employed as a word, meaning “person,” and other times, it is used as a deviational morpheme. For example, the word “haasayee” which means “he talks” is changed to a noun by removing “ee” and adding “asaa” meaning “speaker.” The biggest problem with this is that it is not apparent what influences whether a verb stem takes “asaa” to produce a noun.

Additionally, the suffix “iyaga” is unknown in Dawurootsuwa, albeit it occasionally appears as a suffix morpheme. By adding the suffix “ga” to an adjective that already has the suffix “iya,” the root word “lo’a” which means “good” becomes “lo’iyaga,” which signifies “not bad.” In the above example, “good” changes to “not bad,” and however, languages are not known to follow this criterion.

5. Data Preparation

Unlike Ethiopian languages like Amharic, Afaan Oromo, and Tigrigna, resources for Dawurootsuwa are severely underresourced and difficult to obtain in digital format. Textual data and lexicon are the two primary data required for spell checking.

Lexicon data are crucial for the morphology-based spell checking system; in this study, the lexicon data are specifically collected from the Dawurootsuwa dictionary and linguistic research works. These collected lexicon data were root words and list of affix of the languages. The root lexicon of Dawurootsuwa is prepared from the Dawurootsuwa-Amharic-English trilingual dictionary (second edition) which is organized at the linguistics of the Dawuro Zone Educational Bureau (Tarcha), and the affix lexicon is collected from the works of Dawurootsuwa morphology linguistic research. These collected dictionary words were in *.PDF file format, and for spell checking purpose, it is converted into the *.txt file. The root word was prepared from the dictionary by removing inflected words, removing phrases made of two or more words, and adding country and common person names; all these things are performed by the domain expert of the language. In this study, 5,000 root lexicons are used for root lexicon preparation and more than 3,000 morphological rules were developed for those root lexicons. Both root and affix lexicon data are prepared manually and stored in Hunspell dictionary format. The Hunspell dictionary contains two files which are *.dic for a root dictionary and *.aff for an affix lexicon with morphological rules. Moreover, both root and affix lexicon are prepared independently and stored in different classes.

Similarly, due to the lack of an annotated corpus for the language, datasets for testing were gathered from four distinct sources. The New Testament of the Holy Bible, the Wolaitta Sodo University Linguistics Department, the Dawuro Zone Educational Bureau, and Waka FM 93.4 FM are among these sources. To decide the test data size, the investigation was made on previously conducted Ethiopian spell checking research. Table 2 shows the summary of previously conducted research test data.

As indicated in Table 2, the test data size is not similar for all previous research works, and because of this, the study decided test data size by selecting sample paragraphs and sentences from collected documents. As result, a total of 232 paragraphs totaling 10,437 words were chosen at random; among them, 1,157 words were from the holy Bible, 3,110 words were from Waka FM, 5,314 words were from the educational bureau, and 850 words were from the first-year modules, to be exact. Repeated terms are deleted from the

Table 2: Test data size in previously conducted spellcheckers on local languages.

Title	Test data size
Design and implementation of morphology based spell checker [8]	1464
Automatic spelling checker for Amharic language [16]	1314
Morphology based spell checker for Kafi Noonoo language [18]	2743
Automatic Amharic spelling error detection and correction using hybrid approach [16]	500
Spell checker for Tigrigna language using rule-based morphological analyzer [19]	787
N-gram based Amharic spelling correction for query reformulation [36]	447
Average	1,209

test data, resulting in 3,156 unique words. These words are manually annotated by linguistic experts, who identify 2,976 valid and 180 incorrect words.

6. Methods

Some of the previously conducted research on spelling checker architecture does not consider normalization [8, 17, 18] and is also highly dependent on dictionary words. However, we designed a new architecture for the proposed morphology-based spellchecker for Dawurootsuwa. Basically, it works on the morphological structures of the language by reducing the dictionary size and has the ability to handle word inflection, derivations, and compounding, which are not in the dictionary. Figure 1 presents the proposed system architecture of the morphology-based spellchecker for the Dawurootsuwa language.

As depicted in Figure 1, the architecture consists of four major components: preprocessing, error detection, correction, and knowledge base. In preprocessing, two basic tasks are performed. The first task is to normalize the scripts to the same form, while other subcomponents tokenize the block of texts into tokens. After preprocessing, the existence of errors must be checked with the help of the dictionary from a knowledge base component. The component has two submodules, one error detector, and a morphology analyzer module. The error detector identifies and confirms that the tokens are misspelled or not in a given language, or it checks whether the tokens are misspelled or not. The morphology analyzer module is used to detect inflection, derivation, and compounding of words.

The knowledge base component is used for error identification and correction. The knowledge base component is made up of a dictionary, morphological rules, and an affix set of rules. The dictionary helps identify incorrect words in a given document, while the affix and morphological rule supports the error correction module through preferable suggestion proposal. The Hunspell *.dic file is used to hold just root words in the dictionary that mainly consists of main forms which are lemmas. We used 5,000 root words which are not inflected, derivated, and compounded. Each of the 5,000 root words is assigned in the . dic dictionary one word per line and contains affix information by using a slash sign (/) which uses joint root morphemes with its affix morphemes. Below is an example of the sample root dictionary for the root word from the dw. dic file which is used in our corpus:

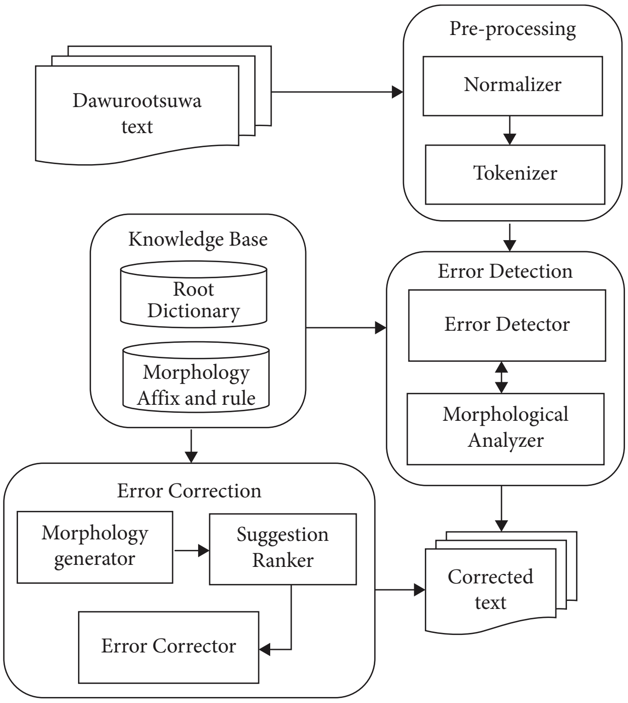


Figure 1: Morphology-based spellchecker system architecture.

(1) usha/DEFG
(2) bonch/CD
(3) amassall/H
(4) kuttu/MO
(5) asa/Z
(6) ka′a

where in line 1, “ush” drink is the root word of the language and DEFG shows the affix class of the root word or identifies attributes of the words. Also, in line 2, “bonch” is a root word for Dawurootsuwa and CD represents the affix class of the root words. Lines 3, 4, and 5 are similar, but in line 6, there is some difference and the word “ka’a” represents the root words of the language and has no affix class; it means that it may occur without any derivation inflection and compounding. Unlike the *.dic file, the *.aff file is used to check against the root words using language rules such as inflection, derivation, and compound rules. An affix is either a prefix or a suffix attached to root words to make other words. For example, here is an example of the Hunspell morphological

```
BEGIN
  Take erroneous tokens from an error corrector
  Read the input tokens from left to right and right to left
  if tokens contain a valid root and a valid suffix then
    strip them and replay the root and suffix to the error detector
  else if tokens contain an invalid root and a valid suffix then
    strip the suffix and replay unknown morphemes to the error detector
  else if tokens contain a valid root and an invalid suffix then
    strip the root and replay a list of root and unknown morphemes to the error detector
  else
    return tokens
  end if
END
```

Algorithm 1: Morphological analyzer

rule for a single root word “wadh’a,” “ush,” and “ka’o” from the dw.aff file which is used in this manuscript:

$$\begin{array}{lllll} \text{SFX} & D & N & 4 & \\ \text{SFX} & D & 0 & \text{di}'\text{ayshin} & d' \\ \text{SFX} & D & a & \text{oppa} & a \\ \text{SFX} & D & {}'o & \text{etiawa} & {}'o \\ \text{SFX} & D & 0 & e & d'. \end{array} \tag{1}$$

The rule specified above is case sensitive and space delimited. The above rule information is described as follows. As indicated, the first line of the rule has 4 fields, which are SFX, D, N, and 4:

(i) SFX refers that this is a suffix

(ii) D refers that the name of the character represents the suffix

(iii) N refers that this character is not to be joined with prefixes

(iv) 4 refers that sequence to store this rule, 4 affix entry rules are required.

The remaining lines indicate the other information for the 4 affix entries.

(i) a indicates the string of chars to remove off before adding suffix **oppa**.

(ii) di’ayshin, etiawa, and e indicate that the string of affix characters to add (zero) in here means the NULL string

(iii) d’ is the condition which must happen before the affix can be applied which means the last character of words must be “d”.

Developing a knowledge-based morphological analyzer for a specific language is dependent on language features. In order to detect spell errors, words must be broken down into a stem. Till now, there is no morphology analyzer developed for Dawurootsuwa; by the help of the Hunspell package, we developed the knowledge-based morphological analyzer for Dawurootsuwa that supports spell checking tasks by decomposing the tokens into the affix and root (Algorithm 1). The tasks of the morphology analyzer are to decompose the tokens into root word and affix. Breaking of words into a stem and affix is carried out automatically by the help of the Hunspell package, but in this study, each and every rule is designed and developed manually. Algorithm 1 presents how the rule-based morphological analyzer works.

Algorithm 1 depicts that the morphological analyzer looks all information from the knowledge base module and strips tokens into the stem and affix. The morphological analyzer works by exact stripping performed for correctly spelled words. However, in this work, it works by striping four conditions. These conditions have the valid root and valid suffix, the valid root and invalid suffix, the invalid root and valid suffix, and the invalid root and invalid suffix. For the fourth condition, the algorithm does not make any striping, rather returns input tokens and resends to the error detector. Once the error is detected, the error correction component accepts erroneous words that are sent to a morphology generator. The morphology generator component handles different derivation, inflection, and compounding of different word class categories such as adjectives, nouns, verbs, adverbs, and pronouns. The morphological generator is constructed based on the Hunspell package knowledge-based morphological generator for Dawurootsuwa to support an error correction process by generating possible fully inflected, derivated, and compounded words. Algorithm 2 presents the knowledge-based morphological generator.

As depicted in Algorithm 2, primarily, the algorithm checks four conditions to generate the possible word sequence. The first criterion is that input morphemes exist in the knowledge base module; if both the root and suffix exist, it appends the suffix morpheme to the root morpheme and generates viable words. The method checks the existence of input morphemes in the knowledge base in the second condition; if the root and suffix do not exist in the knowledge base, it finds a possible suffix for roots and generates a possible list of words. In the third condition, the algorithm checks the existence of input morphemes in the knowledge base; if root words do not exist but the suffix does, it finds plausible roots for the suffix and generates a list of words.

```
BEGIN
    Take erroneous morphemes from an error corrector
    for each erroneous morpheme
    Classify morphemes into different error classes
    if the root and suffix are valid then
        Append the suffix to root and generated words
    else if the root is valid and the suffix is invalid then
        Generate a list of words for the valid root
    else if the root is invalid and the suffix is valid then
        Generate a list of words
    else
        Concatenate input morphemes
    end if
END
```

Algorithm 2: Morphological generator

Finally, the algorithm checks for the existence of input morphemes; if both are invalid or do not exist in the knowledge base, it concatenates the root and suffix morphemes and generates one word. However, this word is invalid for language, so the algorithm does not generate any other words in the fourth condition.

Once the morphological creation operation is completed, the proposal ranker uses LED algorithms to rank the list of all formed words. If two or more words have the same score value, the suggestion ranker uses QUERTY keyboard character distance, which is based on Euclidean distance. Following the completion of the rating task by the recommendation ranker, the ranked list of candidate suggestions replaces the misspelled words by picking and replacing potential candidates.

7. Prototype

Prototype is a model which is used to show the artifact of the design of a system. In this study, the prototype is used to check the correctness of Dawurootsuwa words by taking an input of the text and predicts the input text as correct or incorrect. For incorrect words, the prototype generates possible suggestions based on applied suggestion algorithms. The prototype was developed by the Python programming language with tkinter GUI interfacing. It contains four modules, namely, preprocessing, error detection, error correction, and knowledge base module. As a result, a user provides text by writing the text in the text area and checks the correctness of the word by clicking “check spelling” button. After the “check spelling” button is clicked, the system automatically displays the text which is “correct word” or “suggestions,” see Figures 2 and 3.

8. Experiment Result and Discussion

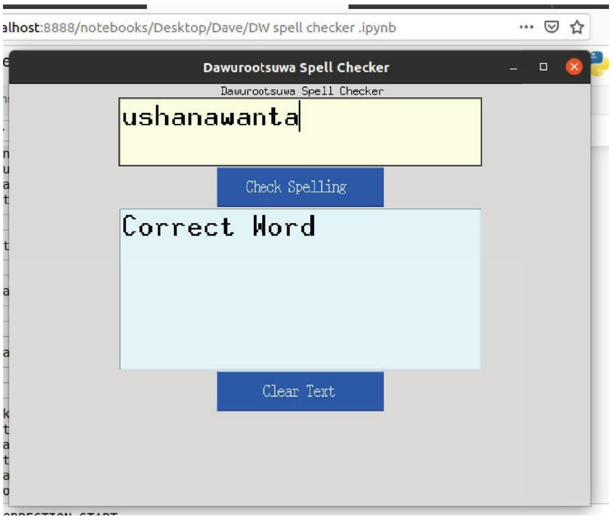
The proposed spellchecker system performs two distinct experiments: spell mistake detection and correction. The first experiment assesses the performance of spell error detection, while the second experiment was for possible suggestions of the detected incorrect words. Error detection was carried out utilizing 3,156 distinct words (2,976 correct and 180 incorrect spell). The system accepted 2,708 valid words as valid out of 2,976 correctly spelt words and reported 268 legitimate words as poorly written. Similarly, out of 180 improperly spelled terms, the system identified 145 as invalid and accepted 35 as valid. Performance evaluation was made using precision and recall of the lexical and error. Figure 4 presents the performance comparison of the lexicon and error using precision and recall of error detection.

As depicted in Figure 4, detection has a lexical precision of 98.73% and an error precision of 35.11%. Similarly, detection has a recall of 90.99% and 80.56% for lexical and error precisions, respectively. Overall, a spell error detection performance of 90.4% was obtained. This experimental result shows the promising result despite the limitation of complete root words in the knowledge base.

The suggested method obtained excellent accuracy by accepting valid Dawurootsuwa words as valid and reporting invalid Dawurootsuwa words as invalid. The system also performed well in terms of lexical recall and error recall. Recall primarily measures the system’s completeness, while precision indicates exactness. This shows that the suggested morphology-based spellchecker is comprehensive and effective. As a result, the experiment results demonstrate that the proposed morphology-based approach outperformed in the spell error detection procedure.

As there is no well-known spelling correction and also the work in [8] did not evaluate the performance of spell error corrections, this study evaluated the performance subjectively using the four criteria’s with the help of linguistic experts. These are as follows: if the suggestion appears within the first three suggestions, the score is 1, whereas if the suggestion appears anywhere, the score become 0.5. If there is no suggestion, then the score becomes 0; otherwise, it is −0.5 if the suggestion is invalid. Accordingly, 121 suggestions within the first three suggestions, 6 suggestions anywhere, and 18 incorrect suggestions were provided by systems, and there existed 13 no suggestions.

As a result of the error detection of the first experiment, 145 words were employed as test data for possible suggestion independent of the context, and three language experts



alhost:8888/notebooks/Desktop/Dave/DW spell checker .ipynb

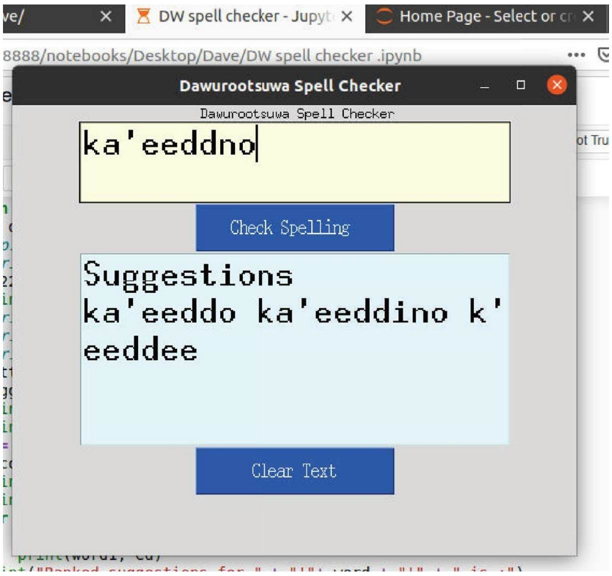
Dawurootsuwa Spell Checker

Dawurootsuwa Spell Checker

ushanawanta

Correct Word

Figure 2: Checking valid word ushnanawanata.



ve/
DW spell checker - Jupyt
Home Page - Select or cr

8888/notebooks/Desktop/Dave/DW spell checker .ipynb

Dawurootsuwa Spell Checker

Dawurootsuwa Spell Checker

ka'eeddno

Suggestions
ka'eeddo ka'eeddino k'
eeddee

Figure 3: Checking invalid word “ka’eeddnoe.”

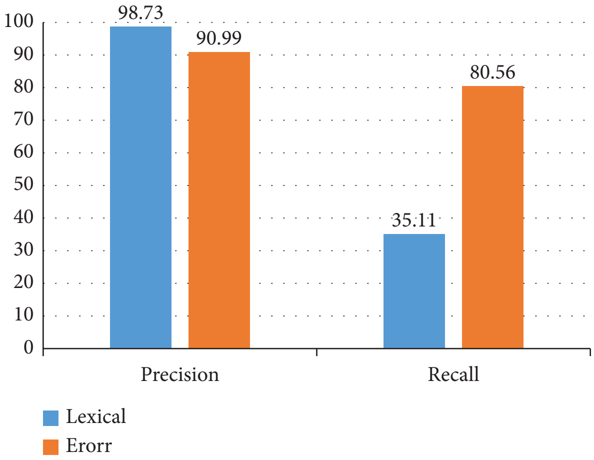


Figure 4: Performance result in error detection.

generated possible suggestions subjectively without taking the context into account. The spellchecker algorithm supplied possible suggestions for 132 of the 145 words. Similarly, suggestion adequacy provided by the rule and the domain experts was calculated, resulting in the overall error correction performance of 79.31%.

9. Conclusion and Future Works

In today’s modern world, the presence of digital resource in various disciplines and languages is continuously increasing. Some of the words in this enormous collection of data are misspelled for a variety of reasons. Spell checking for each word is necessary in order to eliminate spelling mistakes in written material. Spell checking is an important NLP application used to detect and correct misspelled words, particularly in technologically disadvantaged languages. It is widely used in word processing software as well as many other applications. In this study, we developed a morphology-based spellchecker for Dawurootsuwa, a language that is underrepresented.

A total of 5,000 root words, 2,500 morphological rules, and 3,156 unique test words were prepared as corpora for development. Two distinct experiments examine error detection and correction. The performance evaluation results for spelling error detection were 90.40%. This shows that the system performed well at detecting misspelled and correct words of the language. According to the evaluation findings from error correction, the system was able to correct errors to 79.31%. The majority of the mistakenly marked words are nouns such as location, person, and culture names. To address this, we are further working towards including the word class that is not addressed in this study besides the morphological rules.

Data Availability

The data that support the findings of this study can be obtained from the corresponding author on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank all language experts (Ashenafi Alemu, Tariku Akalu, Elias Asefa, and Danial Dawit) for agreeing to participate in this study and for making precious time to support us in data collection, preparation of root words and morphological rules, annotating words, and making manual corrections for misspelled words. This work was supported by the Education Ministry of Ethiopia in collaboration with Hawassa University and Mizan-Tepi University, Ethiopia.

Supplementary Materials

Supplementary Material 1: the table shows sample, randomly selected test data for error detection that are collected from different sources. Supplementary Material 2: the table shows sample test data for error correction, which are true negative words from the first experiment. Supplementary Material 3: the table shows language experts and system corrections for misplaced words along with score values. Supplementary Material 4: the table shows sample Dawurootsuwa affix lexicons, which are basically used for the development of the morphological structures of the language. (*Supplementary Materials*)

References

[1] L. E. Kelvin-Iloafu, "The role of effective communication in strategic management of organizations," *International Journal of Humanities and Social Science*, vol. 6, no. 12, pp. 93–99, 2016.

[2] S. Khemesh, "Effective communication techniques," 2017, https://www.knowledgecity.com/blog/effective-communication/.

[3] W. F. Cascio and R. Montealegre, "How technology is changing work and organizations," *Annual review of organizational psychology and organizational behavior*, vol. 3, no. 1, pp. 349–375, 2016.

[4] E. Green, "Word Processing tool," *Research Associate and Tutorial Fellow*, Oxford University, Oxford, UK, 2023.

[5] D. Jurafsky and J. H. Martin, *Speech and language processing*, vol. 3, Prentice Hall, Hoboken, NJ, USA, 2014.

[6] A. Yazdani, M. Ghazisaeedi, N. Ahmadinejad, M. Giti, H. Amjadi, and A. Nahvijou, "Automated misspelling detection and correction in Persian clinical text," *Journal of Digital Imaging*, vol. 33, no. 3, pp. 555–562, 2020.

[7] S. Martin-Chang, G. Ouellette, and M. Madden, "Does poor spelling equate to slow reading? The relationship between reading, spelling, and orthographic quality," *Reading and Writing*, vol. 27, no. 8, pp. 1485–1505, 2014.

[8] G. O. Ganfure and D. Midekso, "Design and implementation of morphology based spell checker," *International Journal of Scientific and Technology Research*, vol. 3, no. 12, pp. 118–125, 2014.

[9] T. A. Pirinen and K. Lindén, "State-of-the-art in weighted finite-state spell-checking InComputational linguistics and intelligent text processing: 15th international conference, CICLing," *Proceedings, Part II 15 2014*, pp. 519–532, Springer Berlin Heidelberg, Kathmandu, Nepal, 2014.

[10] N. Mohammed and Y. Abdellah, "The vocabulary and the morphology in spell checker," *Procedia Computer Science*, vol. 127, pp. 76–81, 2018.

[11] M. Starlander and A. Popescu-Belis, "Corpus-based evaluation of a French spelling and grammar checker," 2002, https://www.researchgate.net/publication/2832205_Corpus-based_Evaluation_of_a_French_Spelling_and_Grammar_Checker.

[12] M. Blazquez and C. Fan, "The efficacy of spell check packages specifically designed for second language learners of Spanish," *Pertanika Journal of Social Science and Humanities (JSSH)*, vol. 27, no. 2, pp. 847–863, 2019.

[13] J. Liu, F. Cheng, Y. Wang, H. Shindo, and Y. Matsumoto, "Automatic error correction on Japanese functional expressions using character-based neural machine translation," in *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, China, December 2018.

[14] Y. Hong, X. Yu, N. He, N. Liu, and J. Liu, "FASPell: a fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm," in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pp. 160–169, Hong Kong, China, November 2019.

[15] D. M. Eberhard, G. F. Simons, and C. D. Fennig, *Ethnologue: Languages of Asia*, sil International, Dallas, Texas, 2019.

[16] G. Assefa, "Automatic Amharic spelling error detection and correction by using hybrid approach," *Bulletin of Electrical Engineering and Informatics*, vol. 5, no. 6, p. 7, 2018.

[17] M. Tilahun, "Automatic spelling checker for Amharic Language," Doctoral dissertation, Addis Ababa university, Addis Ababa Ethiopia, 2020.

[18] Y. A. Fikru Tafesse Bekele, "Morphology based spell checker for Kafi Noonoo language," Thesis, Addis Ababa University, Addis Ababa Ethiopia, 2018, http://213.55.95.56/bitstream/handle/123456789/19584/Fikru%20Tafesse%20%202018.pdf, Addis Ababa Ethiopia.

[19] B. Hadis, "Spell checker for Tigrigna language using rule based morphological analyzer and unsupervised approach," Doctoral dissertation, Addis Ababa university, Addis Ababa Ethiopia, 2020.

[20] S. Hanserud, "Dawro verb morphology and syntax-A description," M.Sc. thesis, Universtitas Osloensis, Oslo, Norway, 2018.

[21] D. B. GebreGiorgis, *Language Documentation Based Lexical Study of the Earlier Dawuro Kingdom*, Addis Ababa university, Addis Ababa Ethiopia, 2016.

[22] T. Negese, *Aspect of Dawro Phonology*, Addis Ababa University, Addis Ababa Ethiopia, 2010.

[23] J. Kaur and K. Garg, "Hybrid approach for spell checker and grammar checker for Punjabi," *International Journal*, vol. 4, no. 6, 2014.

[24] E. S. Randhawa and E. C. Saroa, "Study of spell checking techniques and available spell checkers in regional languages: a survey," *International Journal For Technological Research In Engineering*, vol. 2, no. 3, pp. 148–151, 2014.

[25] G. Andrade, F. Teixeira, C. R. Xavier, R. S. Oliveira, L. C. Rocha, and A. G. Evsukoff, "Hasch: high performance automatic spell checker for Portuguese texts from the web," *Procedia Computer Science*, vol. 9, pp. 403–411, 2012.

[26] M. Mihiretu and D. Melkamu, "Investigating factors contributing to grade nine students spelling errors at Don Bosco High and Preparatory School in Batu," *Journal of Languages and Culture*, vol. 2, no. 6, pp. 103–115, 2011.

[27] A. Kusuran, "L2 english spelling error analysis: an investigation of english spelling errors made by swedish senior high school students," 2016, https://www.diva-portal.org/smash/get/diva2:1078118/FULLTEXT01.pdf.

[28] R. Kumar, M. Bala, and K. Sourabh, "A study of spell checking techniques for indian languages," *JK Research Journal in Mathematics and Computer Sciences*, vol. 1, no. 1, pp. 105–113, 2018.

[29] J. L. Peterson, "Computer programs for detecting and correcting spelling errors," *Communications of the ACM*, vol. 23, no. 12, pp. 676–687, 1980.

[30] L. Barari and B. QasemiZadeh, *CloniZER Spell Checker Adaptive Language Independent Spell Checker*, InAIML 2005 Conference CICC, Cairo, Egypt, 2005.

[31] M. Kamayani, R. Reinanda, S. Simbolon, M. Y. Soleh, and A. Purwarianti, "Application of document spelling checker for Bahasa Indonesia," in *Proceedings of the 2011 International Conference on Advanced Computer Science and Information Systems*, pp. 249–252, Jakarta, Indonesia, December 2011.

[32] M. Attia, P. Pecina, Y. Samih, K. Shaalan, and J. Van Genabith, “Arabic spelling error detection and correction,” *Natural Language Engineering*, vol. 22, no. 5, pp. 751–773, 2016.

[33] B. Hamza, Y. Abdellah, G. Hicham, and B. Mostafa, “For an independent spell-checking system from the Arabic Language vocabulary,” *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 113–116, 2014.

[34] E. A. Negash, “Developing English to dawurootsuwa machine transaltion model using rnn,” Doctoral dissertation, Addis Ababa Science And Technology University, Addis Ababa, Ethiopia, 2021.

[35] B. B. Daammana uutee, *Dawurotsuwa Tamarissia Mas’a-faa(5th Class Book)*, 5TSA KIFILIYA, Hawassa, Ethiopia, 2010.

[36] A. Alemu, “Interactive Amharic query reformulation: N-gram query spelling correction based techniques to improve retrieval effectiveness,” Doctoral dissertation, UOG, Gujrat, Pakistan, 2015.