ORIGINAL RESEARCH ARTICLE UHD JOURNAL OF SCIENCE AND TECHNOLOGY

Kurdish Kurmanji Lemmatization and Spell-checker with Spell-correction

Hanar Hoshyar Mustafa, Rebwar M. Nabi

Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq

ABSTRACT

There are many studies about using lemmatization and spell-checker with spell-correction regarding English, Arabic, and Persian languages but only few studies found regarding low-resource languages such as Kurdish language and more specifically for Kurmanji dialect, which increased the need of creating such systems. Lemmatization is the process of determining a base or dictionary form (lemma) for a specific surface pattern, whereas spell-checkers and spell-correctors determine whether a word is correctly spelled also correct a range of spelling errors, respectively. This research aims to present a lemmatization and a word-level error correction system for Kurdish Kurmanji Dialect, which are the first tools for this dialect based on our knowledge. The proposed approach for lemmatization is built on morphological rules, and a hybrid approach that relies on the n-gram language model and the Jaccard Coefficient Similarity algorithm was applied to the spell-checker and spell-correction. The process results for lemmatization, as detailed in this article, rates of 97.7% and 99.3% accuracy for noun and verb lemmatization, correspondingly. Furthermore, for spell-checker and spell-correction, accordingly, accuracy rates of 100% and 90.77% are attained.

Index Terms: Kurdish Language, Kurmanji Dialect, Kurdish Lemmatizer, Kurdish Spell-checker and Spell-correction, Kurdish Dataset

1. INTRODUCTION

The topic of text processing has drawn the interest of numerous scholars as a result of the rising prevalence of digital texts in modern life. The amount of research in the domain of Kurdish text processing seems to be rather minor, despite significant efforts with some of the most popular languages, such as English, Persian, and Arabic.

Commonly, the language experts divided the used languages of the world over families which are by ascending: Indo-European, Sino-Tibetan, Niger-Congo, Austronesian, and some other families. The Indo-European family is the biggest family which speaks by the majority of Europe, the lands where the Europeans migrated, as well as a large portion of South-west and South Asia. This family divided into sub-families [1]. Kurdish language dialects are part of the north-western branch of the Indo-Iranic language family. The Kurdish language is an independent language that has its own linguistic continuum, historical origins, grammar rules, and extensive live linguistic skills. The "Median" or "Proto-Kurdish" language is where the Kurdish language originated. Approximately 30 million people in high land of Middle East, Kurdistan, talk numerous dialects of Kurdish [1].

Kurdish is referred to be a dialectical continuity, which means that it has a variety of dialects, it actually has four primary dialects (groups) and sub dialects, including (Kurmanjí or Kurmanji Zhwrw and Badínaní) in the north of Kurdistan and Sorani or Kurmanji Khwarw in the center

Access this article online

DOI:10.21928/uhdjst.v7n1y2023.pp43-52 | E-ISSN: 2521-4217 P-ISSN: 2521-4209

Corresponding author's e-mail: Hanar Hoshyar Mustafa, Technical college of Informatics, Sulaimani Polytechnic University, Sulaimani 46001, Kurdistan Region, Iraq. E-mail: hanar.hoshyar.m@spu.edu.iq

Received: 23-10-2022 Accepted: 05-01-2023 Published: 22-02-2023

of Kurdistan (Sulaimani and Mukrayani). Kurmanji and Sorani are indeed the two main dialects [2]. Additionally, the other two important divisions of Kurdish language are Goraní (Hawrami, Zazayee and Shabak) and Luri (Mamasani, Kurmanshani and Kalhuri). Furthermore, these are categorized into dozens of dialects and sub-dialects [3]. This paper focuses on the Northern Kurdish dialect which is (Kurmanji or Kurmanji Zhwrw) dialect which has the biggest number of speakers in comparison to other Kurdish languages dialects [4]. Several studies have been done related to common languages such English [5], [6], Arabic [7]-[9], and Persian [10]-[12]. Moreover, there are few studies which are consummated regarding Kurdish language [13], [14], despite it, a huge gap can be seen in the case of Kurdish Kurmanji dialect; therefore, this study has been aimed to serve this gap due to Kurmanji dialect in the case of creating lemmatization and spell-checker with spell-correction system. Hence, in the future, this study can be used in several applications that include data translation, sentence retrieval, document retrieval, and also can be extend and upgrade to more powerful similar systems.

This study presented a toolkit, which consists of a lemmatization system and a spell-checker with spell-correction for Kurdish Kurmanji. The aim of the lemmatization is to find a root or dictionary form (calls a lemma) for a specific surface form. It is crucial to be able to normalize words into their most basic forms, particularly for languages with rich morphology such as Kurdish language, to better assist processes such as search engines and linguistic case studies.

Spell-checking algorithms are one of the lemmatizer's most commonly used applications. With using a spell checker, the system suggests a rating of suggested corrections for each possibly incorrect word.

This study presented a combination algorithm which are n-gram language model together with Jaccard Similarity Coefficient for the spell-checker and spell-correction system. Furthermore, a rule-based method on the Kurdish Kurmanji morphological rules is used in creating the lemmatization system.

Based on the literature and to the best of our knowledge, no study has been conducted regarding the spell-checking and lemmatization systems in Kurdish Kurmanji Dialect. Therefore, our study can be the base for further studies for Kurdish Kurmanji dialect.

2. RELATED WORK

There has been a huge amount of research that has been conducted regarding the word lemmatization, spell-checker, and spell-correction in several common languages, such as English, Persian, and Arabic. However, when it comes to Kurdish language, a large absence can be observed, especially in lemmatization and spell-checking with spell-correction system in Kurdish Kurmanji dialect.

In the case of lemmatizer in English language Lemma Chase which is a lemmatizer is created [5] address the problems of the most widely used lemmatizers currently available, this research presents a lemmatization model. This model accounts for the nominalized/derived terms for which no lemmatizer currently in use is able to produce the proper lemmas. Identifying the morphological structure of any input English word, and in particular understanding the structure of the derivational word, is the main issue in developing a lemmatizer. Finding the derivational suffix from morphing words and then extracting the dictionary base word from that derived word is another crucially difficult problem for a lemmatizer. Some derivative terms are not handled by well-known and well-liked lemmatizers to retrieve their basis words. Lemma Chase, the mentioned lemmatizer, accurately retrieves the base word while taking into account the word's Part of Speech, several classes of suffix rules, and effectively executing the recoding rules utilizing the WordNet Dictionary. All of the derivational and nominalized word forms that are present in any standard English dictionary are successfully used by Lemma Chase to construct the base word form.

In addition, there have been numerous studies on spell checkers in Arabic. For instance, Build Fast and Accurate Lemmatization for Arabic [7] which is a study that covers the need for a quick and precise lammatization to improve Arabic Information Retrieval (IR) outcomes and the difficulty of developing a lemmatizer for Arabic, since it has a rich and complex derivational morphology. Introduces a new data set that can be used to verify lemmatization accuracy as well as a powerful lemmatization algorithm that works more accurately and quickly than current Arabic lemmatization techniques.

Numerous studies have been published on the use of spell checkers and spell correction in Persian as well. For example, Automated Misspelling Detection and Correction in Persian Clinical Text [10] is an article that explains the creation of an automatic method for identifying and fixing misspellings in Persian free texts related to radiology and ultrasound.

Three distinct forms of free texts associated to abdominal and pelvic ultrasound, head-and-neck ultrasound, and breast ultrasound reports are utilized using n-gram language model to accomplish their aim. For free texts in radiology and ultrasound, the system obtained detection performance of up to 90.29% with correction accuracy of 88.56%. The findings suggested that clinical reports can benefit from high-quality spelling correction. Significant cost reductions were also made by the system throughout the documentation and final approval of the reports in the imaging department.

Kurdish stemmer pre-processing for improving information retrieval conducted by researcher in [13]. This article introduces the Kurdish stemming-step method. It is a method that links search phrases and indexing terms in Kurdish texts that are connected by morphology. In actuality, the occurrence of words demonstrates a supportive role for the classification process. Even though it was planned to produce more or fewer errors to demonstrate the complexity and difficulty of words in the Kurdish Sorani dialect, the handling of similarity changes was implemented, which helped to boost matching among words and decrease the storage requirements. However, the stemmer used in this work was capable of resolving most of these issues. There are many stop words with added affixes in Kurdish Sorani writings. Therefore, by combining these commonly occurring stop words, it can be stemmed. In addition, it was determined that employing partial words during the pre-processing stage was preferable.

Likewise building a Lemmatizer and a Spell-checker for Kurdish Sorani presented by [14]. This study also presented a lemmatization and word-level error correction system for Kurdish Sorani. It suggested a hybrid strategy focused on n-gram language modeling and morphological principles. Systems for lemmatization and error detection are referred to as Peyv and Renus, respectively. The Peyv lemmatizer is created based on the morphological rules, and for Renus, it corrects words both with using a lexicon and without using a lexicon. It indicates that these two basic text processing methods can lead the way for more study on additional natural language processing applications for Kurdish Sorani.

Last but not least, intensive literature search has been conducted but no studies have been found considering the Kurdish Kurmanji Dialect. Therefore, this article's primary goal is to propose a lemmatization and word-level spell checker with correction method for a Kurdish language dialect known as Kurmanji. The benchmark of this paper is [14] which is useful for the research study, despite the different algorithms used in spell-correction tool, the lemmatization tools are nearly similar in using the methods and approaches, both studies suggest a hybrid strategy based on n-gram language model and morphological principles. This study employs the Python programming language to process data as well as to create a word processing system that performs lemmatization and spell checking with spell correction at the word level.

3. METHODS AND DATA

This section describes dataset collection, data preparation, and algorithms as well as approaches which have been used in lemmatization and spell checker.

3.1. Dataset Collection

A model dataset was produced in order to carry out this study. The dataset was created by reading books and articles written in the Kurdish Kurmanji dialect, which were then manually recorded and added to the dataset. Kurdish Kurmanji dialect words include verbs, nouns, conjunctions, stop words, pronouns, imperative words, superlative words, and question words. There are around 1200 words in the dataset. Fig. 1 depicts the dataset's data amounts in a pie chart. This split results from the differing morphological rules for nouns and verbs, which affect how nouns and verbs are lemmatized. The third dataset has a large number of words that do not accept any affixes. Furthermore, it contains a few special terms with only one or two letters. Some of the conjunction words, for instance, are written with only one or two letters.

3.2. Data Preparation

The most important features that indicated that the dataset was ready for analysis were its unity and quality. Furthermore,

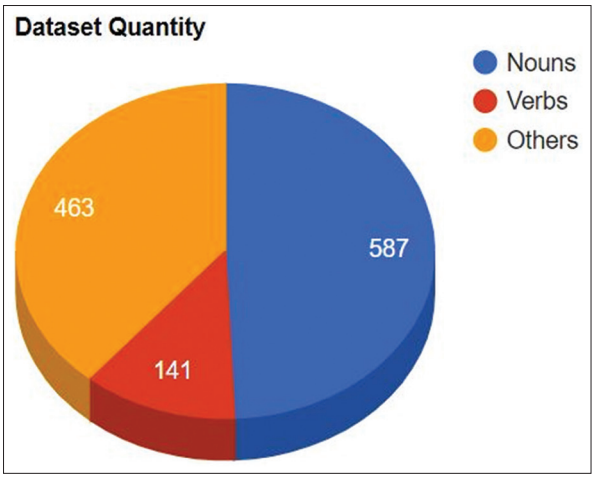

Fig. 1. Dataset quantity pie chart.

because the dataset is the primary first-hand collected dataset, it can ensure that the dataset is clean and has no duplicates. The dataset is then divided into three subsets. The first subset includes nouns. The second subset includes verbs, while the third subset contains pronouns, stop words, conjunctions, imperative words, superlative words, and question words. All of the subsets were stored in separate Excel files, each with two columns: the ID column and the data (word) column. Except for the third subset, which contains the verbs, it has four columns: ID, Chawg, Qad, and Rag. The ID column contains a unique ID for each row; the Chawg column contains the verb's base; the Qad column contains the verb's past root; and the Rag column contains the verb's present root. Table 1 presents the structure of the third (verb) Excel sheet.

3.3. Implementation

This section describes the approaches and methods used according to noun lemmatization, verb lemmatization and spell-checker.

3.3.1. Lemmatization

Lemmatizations for nouns and verbs are developed separately, after obtaining the fundamental morphological rules in Kurdish Kurmanji. Each of noun and verb lemmatization use different approaches based on the morphological rules. For the lemmatizations a pruning method is used to find out the root of the input word. In the background of the system, each process is contained in a module inside the system, as a result to eliminate complexity and increase simplicity, also to made the system more readable and understandable.

The following subsections clarify each of noun and verb lemmatizations in detail.

3.3.1.1. Noun lemmatization

According to the noun lemmatization, the noun lemmatization was created after clarifying and writing down all the rules in accordance with nouns in Kurdish Kurmanji dialect. A pruning method is used in this study. The input word to the system went through multiple stages and processes until the system found the proper root for the input noun, which is called a lemma in lemmatization process.

TABLE 1: Structure of verb-dataset

Verb dataset	Column	Include
	ID	Data ID
	Chawg	Base of verb
	Qad	Past root of verb
	Rag	Present root of verb

During the process of noun lemmatization, predefined affixes and nouns in the dataset are used to find a proper lemma for an input noun. The only condition is to enter the word with the correct spelling.

When a noun was entered, a search algorithm was used to look for it in the dataset. If the entered noun was a root without any affixes, the system determined that the input was correct and that no further processing was necessary. The output word in the outcome would be the base of the entered word. Fig. 2 shows the flowchart diagram of this process.

In other cases when the entered noun is with or attached to some affixes, in this study in the noun lemmatization module, three sets of affixes were defined. First set included prefixes that write before the noun without attaching to the noun directly, in Kurdish Kurmanji, there are some prefixes that write with a space separated with the noun. Second set included the prefixes which are write and attached directly to the beginning of the noun without any space. Moreover, the last set included suffixes which are directly attached to the end of the noun.

The entered noun went through multiple processes to find the root out. The system first removes any prefixes which are

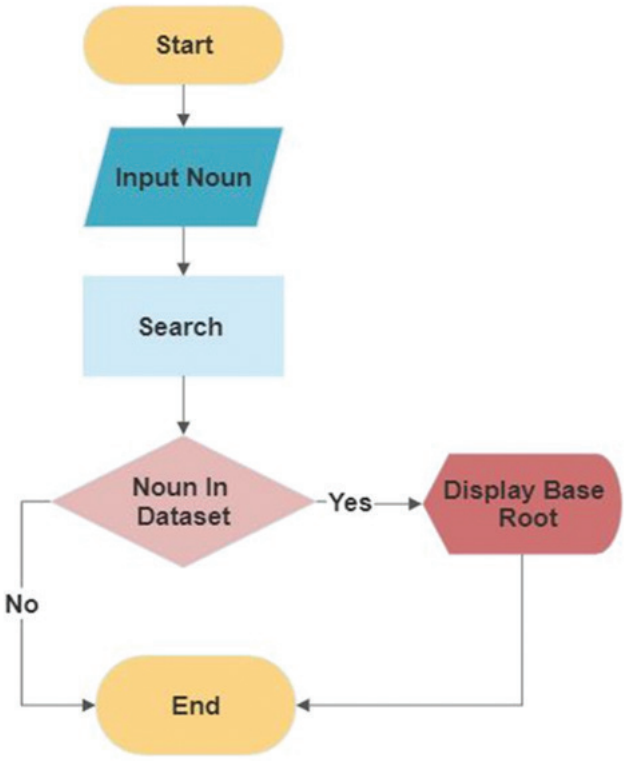

Fig. 2. Noun lemmatization first process flowchart.

attached or not attached prefixes to the word, then search in the dataset, if there was no matching for the entered noun, the system decided that it might attached to some suffixes too, then the word went through another process which removed the possible suffixes attached to the noun, after that a search process look to find out if there was any matching word in the dataset, if any matching word found in the dataset, it would be the return root as the result. This process showed in a flowchart diagram in Fig. 3.

Although there were no words that matched, the system made an effort and forwarded the entered noun to a procedure designed to remove prefixes and suffixes one at a time. In another sense, it took away the first prefix attached and looked for a matching root; if no matching root was discovered, it took away the first suffix attached and looked once more. It continued the process until the root was discovered if a matching root had not yet been discovered and there were further prefixes and suffixes linked to the word. At the end, when there were no more affixes, the entered noun was well spelled and the noun root existed in the dataset, the system gave the correct output lemma (root) for the entered noun. However, the system would replay with the message "Input word is not in the dataset" if there was no match between the entered noun and nouns in the dataset. Fig. 4. shows the process' flowchart diagram.

Following these steps, the user sees the procedures' output, as depicted in Figs. 5 and 6. In Fig. 5, the true word (کچ) (kiç) which means (girl) with two Kurdish Kurmanji prefixes (مک) (ek) and (ا) (a) in the form of (کچەکا) (kiçeka) means (The girl who) entered. The system replayed with ("found", "کچ");

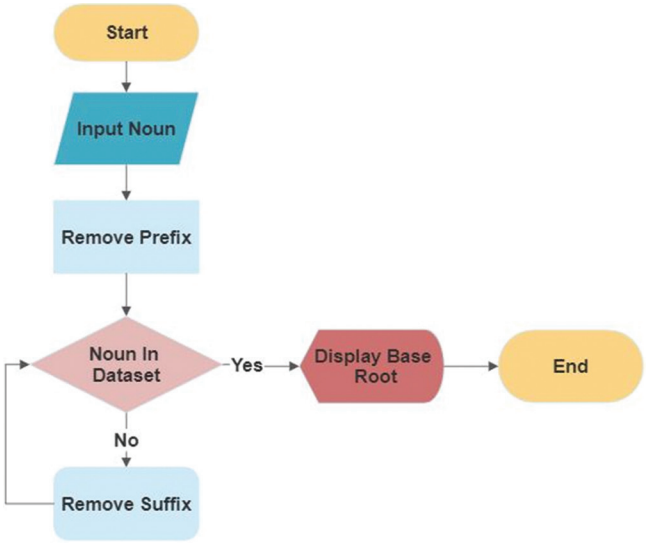

Fig. 3. Noun Lemmatization second process flowchart, Phase 1.

"found" denotes that the entered word is correct and already exists in the dataset, and "کچ" is the base root of word (کچەکا). However, in Fig. 6, the user inputted the incorrect term (کجان) (kican) in the meaning of (کچان) (kiçan) (girls) but with incorrect ending of (ج) (c) rather than (چ) (ç), followed by a correct prefix (ان) (an). Due to the incorrect spelling of the word, which confounded the system and prevented it from locating the specific base root of the word, the system replayed with the message "Input word is not in the dataset."

3.3.1.2. Verb lemmatization

Verb lemmatization also implemented in a pruning method as the noun lemmatization. After Kurdish Kurmanji dialect verb morphological rules are defined, the verb lemmatization is applied. The input verb went across several procedures until the tool selected and found the proper root.

Due to the Kurdish verb's morphology, the addition of prefixes and suffixes to the verb roots, and their ability to alter meaning, finding the root of the verb during the lemmatization process is more difficult and different than finding the root of a noun. Therefore, simply omitting the suffix is worthless.

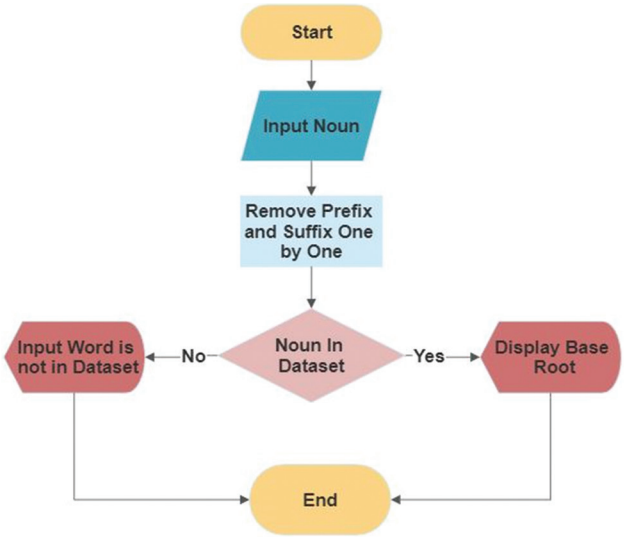

Fig. 4. Noun Lemmatization second process flowchart, Phase 2.

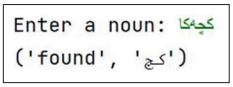

Fig. 5. Noun Lemmatization of a legitimate noun.

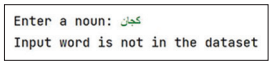

Fig. 6. Noun Lemmatization of an incorrect spelled noun.

In Kurdish language morphology, each verb has three states includes its critical state, which is called (Chawg) in Kurdish morphology; in this state, every verb ends with an (N) (ن) letter at the end of the word; the (N) (ن) is called (the N of Chawg) that determines the critical state of the verb. Another state is when the verb turns into its past state, which is called the "past root," and this is done by removing the (N of Chawg) at the end of the verb. Whenever the verb is in the past state, it can be used in the past tense. The final state is present, and it has several rules to modify a verb critical state and turn it into its present root. When the verb is changed to its present root, it can be used in the present tense [2].

When the input was processed by the system, any affix containing the verb had to be removed. As a result, three sets of affixes are defined, which include suffixes, prefixes that do not attach to the verb, and prefixes that attach to the verb directly. After removing affixes, the remaining verb had to be compared with the verb dataset in the system. As it is clarified in the verb dataset excel file, there were four columns included (ID, Chawg, Qad, and Rag), in which Chawg referred to the critical state of the verb, Qad referred to the past state, and Rag referred to the present state. After the state of the verb was recognized and found, the system returned the critical state, which is the Chawg of the verb as the base root of the entered verb. Fig. 7 shows the process of finding the root of a verb if the entered verb is already a root; no matter in which tense it appears, the system returns the base root of it. Moreover, Fig. 8 depicts the processes for locating a verb root if the entered verb is attached to some affixes; the processes are identical to those for locating a root of a noun attached to affixes in noun lemmatization.

After completing these stages, the user sees the output of the procedures, as shown in Figs. 9-11. In Fig. 9, the true word (دخۆم) (dixom) denotes the present tense of the verb (خارن) (xarin) (eat), while the prefix (د) (d) indicates the present term of the verb and the suffix (م) (m) is the pronoun that denotes (I). The system repeated ("found", "خارن") in the output, where "found" implies that the word is correctly spelled and that its present root, which is (خۆ) (xo), is available in the dataset, and "خارن" is the base root for the entered word. In addition, in Fig. 10, the past tense of the same word (خارن) (eat) is entered

```
GET InputVerb
FOR each Word in DataSet
    IF InputVerb is equal to Word THEN
        DISPLAY InputVerb
    END OF IF
END OF FOR
```

Fig. 7. Verb Lemmatization first process pseudo code.

```
GET InputVerb
CALL removePrefix RETURNING verbWithoutPrefix
IF verbWithoutPrefix is in Dataset THEN
    DISPLAY verbWithoutPrefix
ELSE
    CALL removeSuffix RETURNING verbWithoutSuffix
END IF
IF verbWithoutSuffix is in DataSet THEN
    DISPLAY verbWithoutSuffix
ELSE
    CALL removePrefixAndSuffixOneByOne RETURNING returnedVerb
END IF
IF returnedVerb is in DataSet THEN
    DISPLAY returnedVerb
ELSE
    DISPLAY wordNotInDataset
END IF
```

Fig. 8. Verb Lemmatization second process pseudo code.

```
Enter a verb: دخۆم
('found', 'خارن')
```

Fig. 9. Verb Lemmatization of correct present tense of verb (خارن) (xarin) (eat).

```
Enter a verb: خارماڤە
('found', 'خارن')
```

Fig. 10. Verb Lemmatization of true past tense of verb (خارن) (xarin) (eat).

```
Enter a verb: مەخر
Input word is not in the dataset
```

Fig. 11. Verb Lemmatization of a wrong spelled negative imperative of verb (خارن) (xarin) (eat).

```
SET frequencyOF = []
FOR each inputWordBiGram of QueryTerm
  FOR each dataSetWordBiGram of dictionaryTerm
    IF inputWordBiGram is equal to dataSetWordBiGram THEN
      INCREMENT frequencyOf[inputWordBiGram]
    END OF IF
  END OF FOR
END OF FOR
```

Fig. 12. Query term bi-gram frequency calculation pseudo code.

as (خارمەڤە) (xarmeve), which means (I ate). This time, there are two suffixes: (م) (m), which is the pronoun associated to (I), and (ڤە) (eve), which indicates that the event occurred and ended completely in the past. Once more, the system verified that the word root was correctly spelled that it was included in the dataset; it also displayed the base root of the term. Furthermore, in Fig. 11, entered the wrong negative imperative phrase (مەخر) (mexir) instead of (مەخو) (mexu) or (مەخۆ) (mexo) which means (don't eat), but with the improper ending of (ر) (r), rather than (ۆ) (o) or (و) (u). The system displayed the message "Input word is not in the dataset" according to the word's incorrect spelling, which confused the system and prohibited it from finding the precise base root of the word.

3.3.2. Spell checker and spell correction

The spell checker and spell correction mechanisms collaborated in two stages in this study: First, the spell checker indicated whether the word was correct or incorrect, and second, the spell correction process corrected the word by suggesting some correct words by providing the most likely correct word forms.

After the word entered the system, it was detected if it was true or not by the spell checker's check for word frequency in the dataset (including the whole of the three files). The step of finding that the word is true or detecting the word as wrong was done based on using n-grams. The input word, which is called the query term in this paper, is fragmented into bi-grams (two grammatical units). A bi-gram is an n-gram for $n = 2$. In this study, a 2-g (or bi-gram) is a two-letter sequence of letters. The bi-grams sequences "ha," "ap," "pp," and "py," for instance, are two-letter grammatical sequences extracted from the word (happy). After the bi-gram of the query term is produced, the system calculates the gram frequencies with the bi-grams of the words in the dataset separately, which is called a dictionary term in this paper. Fig. 12 shows the process of calculating the frequency of bi-grams in the query term in comparison to the dictionary terms. After calculation, the system looked up the frequencies of the bi-gram of the query term; if one of the frequencies was equal to zero, then it detected the word as a wrong one as one of its bi-grams had no repetition in comparison with the dictionary terms, and if none of the frequencies were zero, then the word was detected as true. Hence, in the event that a query term equals one of the index terms in the dataset, this word will be selected as true, and if the word is detected as true, then the system presents "The word is true spelled" as a result.

After detecting the query term as wrong, its bi-grams are handled, and the system goes to the spell correction procedure. The wrong word is then corrected based on the Jaccard similarity coefficient method, which is popularly used to compare how close the query terms in the dataset are to one another. Here, the procedure of similarity measurement can be used to examine the most comparable terms that are structurally recorded in the dataset if a query does not match any index in the dataset. Using the Jaccard similarity coefficient [15], Equation (1) shows the rule of the Jaccard similarity coefficient.

$$\text{Jaccard sim } (A,B) = P(A \cap B)/(BUA) \quad (1)$$

Measuring the Jaccard similarity coefficient between two datasets is done by dividing the number of features that are common to all by the number of properties [15].

The mechanism worked on the query term, and dictionary terms included all three files of the dataset. The spell correction took the query term, looked for the matching dictionary term in file one, if it did not exist, then sent it to the lemmatization files, respectively, because it may be the root of a noun or a verb; also, it might be a noun or a verb with affixes, and the affixes should be removed as a result to check if it was spelled correctly or not. After checked process did go well in detail, if the word found, the system marked it as a true word. Otherwise, spell checker predicted words based on the dataset's three files, then it chose the best matching words based on the highest matching degree, which is calculated using the Jaccard Coefficient algorithm, and best matches were chosen if their matching degree were greater than the spell checker's threshold, and finally the five highest matching degree words were chosen. The threshold of this study is equal to 0.15. It has been chosen based on the accuracy of the guess for the correct word or the highest matching words in the dataset for the wrong query term. In Kurdish Kurmanji, there are words with three letters; if they are written incorrectly by missing a letter, they only have two letters. Hence, the threshold should be as small as possible to get a great and accurate result.

4. RESULTS AND DISCUSSION

This section presents the results of the algorithms in both lemmatization and spell checker tools. Also discuss the benchmarking with the benchmark study of the research.

4.1. Noun Lemmatization

To improve the efficiency and accuracy of the noun lemmatization tool, two random words were chosen with

their derivatives which were nine derivatives of (چیا) (mountain) word and 12 derivatives of (کوڕ) (boy) word. The results of lemmatization process of both words were successfully giving correct root in both nine derivatives of first word and 12 derivatives in second word. To ensure the accuracy of noun lemmatization, another 66 random words with possible derivatives were chose and entered into the system; therefore, the noun lemmatization gave correct result in 63 cases out of 66, which means that the noun lemmatization algorithm had an accuracy of approximately 95.45% in lemmatizing words. Overall, the accuracy of the noun lemmatization process was approximately about 97.7%. Table 2 presents accuracy in noun lemmatization tool.

4.2. Verb Lemmatization

To evaluate the efficiency and accuracy of the verb lemmatization tool, two sets of random verb forms were tested with the tool. The test sets included different verb forms such as present and past tense, imperative and negative imperative, passive, and negative. Regarding the verb's existence in the dataset dictionary, the verb lemmatization tool found the correct root of the input verb. Each verb in the test set was entered with all possible derivations made with specific prefixes and suffixes. The first set included 171 different forms of different verbs. The lemmatization tool lemmatized 169 of them correctly; the wrongly lemmatized ones were due to the ordering of the dataset; in the case of imperative and negative imperative of a verb, the lemmatized verb Rag was coming before the purposed verb Rag, so the system took the first verb Rag before it reached the purposed one. For example, the Kurdish verb "send" has two forms: (ناردن) (nardin) and (نارتن) (nartin) and both have the same Rag (نێر) ("nêr"). If a user entered the imperative tense of this verb, which is (بنێرە) (binêre), and expected to see the base root of (ناردن) (nardin) in the result, the system replays with the base root of (نارتن) (nartin) because it was recorded before the other form (ناردن) (nardin) in the dataset excel file. Moreover, it is due to the system, when it finds a result, it stops without going to the other verbs in the dataset.

Moreover, it is due to the system, when it finds a result, the system stops and the result appears without going to the other data in the dataset. As a result, the accuracy of lemmatizing the first set was 98.83 percent. In the order of the other set, there were 131 different forms of different verbs with different tenses. Due to this set, the lemmatization tool lemmatized all of them, which means it gave the correct root for each of the forms. It can be said that with the two test sets, the verb lemmatization tool overall gave approximately 99.4 percent accuracy. Table 3 shows the accuracy of the verb lemmatization tool.

4.3. Spell Checker and Spell Correction

According to calculate and analyze the accuracy of the spell-checker and spell-correction tool, the process of analyzation is more complex, due to connecting the spell-checker and spell-correction tool with the lemmatization tools. As described in the above section, there was three datasets, so the spell-checker and spell-correction accuracy should be calculated according to all the datasets. The mechanism as said is to first check if the input word is correct or not, and the spell-checker tool is tested with three groups of data which are consisted in the three datasets as well. These three groups included 100 words from first dataset file, 100 nouns from second dataset file, 100 verbs from third dataset file, respectively. The result always returned true which meant the input word spelling is correct, while the data existed in the dataset. Hence, it reached to be said that the spell-checker tool returned in all cases successfully. Table 4 shows the accuracy of spell-checker tool.

For the spell-correction tool a set of random words included noun, verb and others is tested, the contained nouns and verbs included all forms with prefixes and suffixes also simple noun and verbs without prefixes and suffixes. The result shows that whenever a bi-gram of the original correct word came in the input word, it was a higher chance to get the most correct word and most similar word as a result. The more bi-

TABLE 2: Accuracy in noun lemmatization tool

Sets	True lemmatization	False lemmatization	Total	Accuracy (%)
1st set	21	0	21	100
2nd set	63	3	66	95.45
Total	84	3	87	97.7

TABLE 3: Accuracy in verb lemmatization tool

Sets	True lemmatization	False lemmatization	Total	Accuracy (%)
1st set	169	2	171	98.83
2nd set	131	0	131	100
Total	300	2	302	99.3

TABLE 4: Accuracy in spell checker tool

Sets	True spell checking	False spell checking	Total	Accuracy (%)
1st set	100	0	100	100
2nd set	100	0	100	100
3rd set	100	0	100	100
Total	300	0	0	100

TABLE 5: Accuracy in spell correction tool

Sets	True correction	False correction	Total	Accuracy (%)
1st set	55	6	61	90.16
2nd set	71	9	80	88.75
3rd set	100	7	107	93.4
Total	226	22	254	90.77

grams of the original wanted word came in the input word, the higher similarity degree get and the more accurate results acquire in the outcome. In several occasions, the incorrect lemmatization occurred because of the incorrect input word, and this led to incorrect spell-correction which at the end resulted in a low accuracy degree of the outcome result.

To provide the efficiency of the spell-correction a set included 100 of wrong random words with different forms were tested manually. First set included 61 wrong spelled nouns, the spell-corrector with the help of the noun lemmatization resulted an accuracy of 90.16% of the correction process. Second set contained 80 wrong spelled verbs, in the result spell-corrector with the use of verb lemmatization gave an accuracy of correction process with 88.75% rate. Third set consisted the wrong spelled pronouns, stop words, conjunctions, imperative words, superlative words, and question words, in 107 words the spell-correction system corrected 100 of them successfully which give accurate result as 93.4% of accuracy rate. Table 5 displays the accuracy of spell-correction tool.

As shown in Table 5, the third set had the highest accuracy rate among the other two sets, and as previously stated, some false correction cases occurred due to false lemmatization, so it must be stated that if a dataset is created with all the forms of the words in all three datasets, then more accurate results can be obtained because the spell-corrector can directly look for the right form of the input misspelled word and find it with a high degree of certainty.

5. CONCLUSION AND FUTURE WORKS

Information retrieval and text classification can benefit greatly from effective lemmatizer. In addition, incorrect words are detected and corrected by spell-checkers and spell-correction. This paper introduced the Kurdish Kurmanji lemmatizer and word-level spell-checker with spell-correction methodologies. It is the first attempt that tools of this kind have been made for Kurdish Kurmanji. A hybrid technique has been utilized for the spell-checker and spell-correction that depends on the n-gram language model and the Jaccard Coefficient Similarity algorithm, also the proposed approach for lemmatization, is based on morphological principles. The outcome demonstrated that, while applying the suggested approach, the accuracy of lemmatization for each noun and verb lemmatization was assessed, respectively, at 97.7% and 99.3%. In addition, the spell-checker and spell-correction accuracy rates were 100% and 90.77%, respectively. The experimental findings show that several false correction cases were caused by incorrect lemmatization led by misspelled input words. Furthermore, according to experimental findings, more accurate results may be obtained if a dataset is established with all the word forms in the datasets since the spell-checker will directly search for the correct form of the input misspelled word and discover it with a high level of equality. In the future, this work can be expanded to apply to a bigger dataset of Kurdish Kurmanji and utilize these approaches for NLP applications like text mining for Kurdish Kurmanji.

As a contrast between this study and its benchmark. Actually, this study is done for the Kurdish Kurmanji dialect, while the benchmark was done for the Kurdish Sorani dialect, which has completely different morphological rules in so many phases to study and implement in the system. The datasets that were used were different, while this research's dataset is primary, first-hand, and organized in three subsets. In addition, there are some variances between them in terms of accuracy and the algorithms that have been used. This study achieved 97.7% and 99.3% accuracy for noun and verb lemmatization, respectively, while the benchmark achieved 95% and 89.4% accuracy of two test sets for noun lemmatization and an average of 86.7% accuracy for verb lemmatization. In addition, according to the spell-correction, this study used the Jaccard Coefficient Similarity algorithm and rated 90.77% accuracy, while the other study, as mentioned, used an edit distance algorithm and obtained 96.4% accuracy with a lexicon while, without a lexicon, the correction system had 87% of accuracy. At the end, it has to be said that the similarities can be seen in the theoretical parts and ideas, but for the practical part, a huge difference can be seen from using different programming languages; this study used the Python programming language, while the other used the Java programming language, up to and including recreating the system from the beginning to the end.

6. ACKNOWLEDGMENT

The authors would like to thank SPU for providing the opportunity, support, and funding for this study. Sulaimani, the Kurdish journalist syndicate, is also thanked.

REFERENCES

[1] Z. Kurdî, M.Û. Zarêن Wî and H.S. Khalid. "Kurdish Language, its Family and Dialects". 2020. Available from: https://www.dergipark.org.tr/en/pub/kurdiname/issue/50233/637080 [Last accessed on 2022 Aug 15].

[2] D.N. MacKenzie. "Kurdish Dialect Studies". Oxford University Press, London, 1961. Available from: https://www.books.google.iq/books/about/Kurdish_dialect_studies_2_1962.html?id=eaf2zaeacaaj&redir_esc=y [Last accessed on 2022 May 31]

[3] "Kurdish Academy of Language Enables the Kurdish Language in New Horizon". Available from: https://www.kurdishacademy.org/?q=node/41 [Last accessed on 2022 Jun 04].

[4] N.A. Khoshnaw, Z.U.Z. Sulaimaniyah. "Awer Station", 2011. Available from: https://rezmanikurde.blogspot.com/2018/01/blog-post_26.html?m=1 [Last accessed on 2022 Jun 09].

[5] R. Gupta and A.G. Jivani. "LemmaChase: A Lemmatizer". *International Journal on Emerging Technologies*, vol. 11, no. 2, pp. 817-824, 2020.

[6] D. Hládek, J. Staš, S. Ondáš, J. Juhár and L. Kovács. "Learning string distance with smoothing for OCR spelling correction". *Multimedia Tools and Applications*, vol. 76, no. 22, pp. 24549-24567, 2017.

[7] H. Mubarak. "Build Fast and Accurate Lemmatization for Arabic". vol. Proceedings of the European Language Resources Association (ELRA). Miyazaki, Japan, 2018. Available from: https://www.aclanthology.org/L18-118 [Last accessed on 2022 Jun 08].

[8] N. Zukarnain, B.S. Abbas, S. Wayan, A. Trisetyarso and C.H. Kang. "Spelling Checker Algorithm Methods for Many Languages", in Proceedings of 2019 International Conference on Information Management and Technology, (ICIMTech), 2019, pp. 198-201.

[9] A.A. Freihat, M. Abbas, G. Bella and F. Giunchiglia. "Towards an optimal solution to lemmatization in Arabic". *Procedia Computer Science*, vol. 142, pp. 132-140, 2018.

[10] 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. 2019.

[11] S. Mohtaj, B. Roshanfekr, A. Zafarian and H. Asghari, "Parsivar: A Language Processing Toolkit for Persian," in Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018. Available from: https://www.aclanthology.org/L18-1179 [Last accessed on 2022 Aug 20].

[12] A. Rashidi and M.Z. Lighvan. HPS: A hierarchical Persian stemming method. *International Journal on Natural Language Computing*, vol. 3, no. 1, pp. 11-20, 2014.

[13] A.M. Mustafa and T.A. Rashid. Kurdish stemmer pre-processing steps for improving information retrieval. *Journal of Information Science*, vol. 44, no. 1, pp. 15-27, 2018.

[14] S. Salavati and S. Ahmadi. "Building a Lemmatizer and a spell-checker for Sorani Kurdish". CoRR, vol. abs/1809.10763, 2018. Available from: https://www.arxiv.org/abs/1809.10763 [Last accessed on 2021 Aug 15].

[15] S. Niwattanakul, J. Singthongcha, E. Naenudorn, and S. Wanapu. "Using of Jaccard Coefficient for Keywords Similarity", in Proceedings of the International Multi Conference of Engineers and Computer Scientists. vol. 1, 2013. Available from: https://www.data.mendeley.com/v1/datasets/s9wyvvbj9j/draft?preview=1 [Last accessed on 2022 Apr 08].