Kurdish stemmer pre-processing steps for improving information retrieval

Article *in* Journal of Information Science · January 2017

DOI: 10.1177/0165551516683617

Arazo M. Mustafa
University of Sulaimani
12 PUBLICATIONS **118** CITATIONS

Article

Journal of Information Science
1–13
© The Author(s) 2017
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0165551516683617
jis.sagepub.com

Kurdish stemmer pre-processing steps for improving information retrieval

Arazo M Mustafa
School of Computer Science, College of Science, University of Sulaimania, Kurdistan, Iraq

Tarik A Rashid
Software and Informatics Engineering, College of Engineering, Salahaddin University-Erbil, Kurdistan, Iraq

Abstract
The rapid increase in the quantity of Kurdish documents over the last several years has created a need for improving information accuracy and precision in text classification and retrieval. Language stemming is an imperative pre-processing step for increasing the possibility of matching terms in a document in text classification tasks. Stemming helps reduce the total number of searchable terms within a document or query. This article proposes an active approach for stemming Kurdish Sorani texts to reduce variations of words to single terms or stems. The outcomes of the process, described in this article, demonstrate that decreasing the dimensionality of feature vectors in documents will increase the effectiveness of retrieval when the stemming process is used. This process applied for Kurdish Sorani can be adapted and applied in Kurdish Kurmanji as well for greater efficiency and effectiveness in digital text classification and applications.

Keywords
Kurdish stemming; list of Kurdish stop words; stemming approaches

1. Introduction

The roots of words are important for text searching to improve information retrieval in such applications as search engines for the World Wide Web. The process of converting words into their roots is called stemming. This is a vital process in digital text classification. It seems that text classification requires a language stemming taking place prior to any compression algorithm or implementation. Stemming helps decrease the space required for storing the configurations or indices of terms in the document. It also helps decrease the computational load of the used system.

Stemming approaches have already been developed in different languages such as English, Iranian and Arabic [1–4]. However, there has been little or no research work developed for building stemmers in the Kurdish language. It is of interest to note that stemming is language dependent; in other words, an English stemmer cannot be used in the Arabic language; likewise, an Arabic stemmer cannot be useful for other languages. This article is part of ongoing research where we are developing an information retrieval system for Kurdish Sorani texts, and the stemming technique is an important tool which supports the development of various natural language processing applications. It can be used in information extraction, search engines [5], automatic indexing, unstructured documents [6], machine translation, evaluation, spell checking and so on. Stemmers are language-specific tools. The contribution of this article is to describe the design of a stemming tool featuring all pre-processing steps in one package to be useful and efficient for all applications needed by Kurdish Sorani data sets. The design of a stemming algorithm requires a significant level of linguistic expertise which authors of this article bring to this project. With more and more online information being available in Kurdish Sorani,

Corresponding author:
Tarik A Rashid, Software and Informatics Engineering, College of Engineering, Salahaddin University-Erbil, Kurdistan, Iraq.
Email: Tarik.rashid@su.edu.krd

ا ە ب پ ت ج چ ح خ چ ر ڕ ز ذ س ش ع غ ف ڤ ک گ ل ڵ م ن و ۆ وو هـ ی ێ

Figure 1. The Kurdish alphabets.

a need was felt to develop tools for Kurdish language processing for the approximately 40 million people who speak Kurdish in Iraq, Turkey, Iran, Syria, Lebanon, Armenia, Georgia, Kyrgyzstan, Azerbaijan, Kazakhstan and Afghanistan. Thus, improvement of the word search algorithm for this language is the actual and interesting task. The Kurdish language is in the Indo-European family of languages. This language is spoken in Kurdistan, a large geographical region spanning the intersections of Iran, Iraq, Turkey and Syria [7]. The Kurdish language is generally divided into two dialects: Sorani and Kurmanji [8]. The writing script of Sorani Kurdish alphabets reads from right to left. This is similar to Persian, Arabic and the Urdu languages. The Kurdish alphabet consists of 33 characters as shown in Figure 1. Researchers [7, 9–11] have identified the morphology of the Kurdish Sorani dialect, which has not been otherwise classified.

The main contribution of this experiment is to provide a comprehensive analysis of Kurdish Sorani texts for a number of levels of information retrieval–related issues, particularly (1) using linguistic expertise to design a stemming-step module to strip prefixes, suffixes and postfixes from the given word by steps until to catch potential roots; (2) applying the stemming before stop words removed during the operation of the Kurdish stemming process; (3) grouping Kurdish Sorani words to experiment with further stemming methods and use both over- and under-stemming to investigate the extent and significance of the performance; (4) demonstrating the effectiveness of the hybrid steps for pre-processing compared with not using the hybrid steps. The Kurdish stemming steps, described in this article, are aimed at extracting the root from a given word via removing prefixes, suffixes and attachments at the beginning and ending of words, respectively.

This article is organised as follows: approaches of stemmers are described in section 'Classification of stemming algorithms'. Next, in section 'Related works', we explain related works for stemmers in different languages. Then, in section 'General Kurdish formulation', we describe general Kurdish stemming formulation. In section 'Proposed approach for Kurdish Sorani texts', the proposed stemming architecture for Sorani texts is presented. Then, results and evaluations are described in section 'Computational performance'. In section 'Analytical discussions', analytical discussions are presented. Finally, the conclusion points and future works are outlined in section 'Conclusion and future work'.

2. Classification of stemming algorithms

Diversity in word forms originated from both derivational and inflectional morphology, and the stemming process enables a system to shrink these inflected or derived words to their base root [12,13]. Thus, word stemming is the most central step in the pre-processing phase, and it has been commonly required and applied in many arenas such as natural language processing, information retrieval systems and text mining. Generally speaking, stemming approaches are classified into three groups, namely, truncating or so-called affix removal approaches, statistical approaches and mixed approaches. Each is explained below:

1. *Truncating or affix removal approaches.* In these approaches, the affixes that are contained in words are removed. Affixes are categorised into two types: prefixes and suffixes. The affix removal stemming type was standardised by Lovins [14]. The Porter algorithm is considered as an example of this kind [1]. The stemming algorithm in this article is close to this approach.
2. *Statistical approaches.* These approaches are widely used in languages, for that reason, they are identified as language independent. An *n*-gram stemmer is regarded as an instance of this approach. It is a string-similarity approach in which a word inflation is converted to its stem. An *n*-gram is basically an *n* string; they are regularly adjacent characters that can be removed from continuous text segments [15].
3. *Mixed approaches*. It involves three approaches of stemming and is debated in Jivani [15]. These types can be explained as follows:
   3.1. *Inflectional and derivational approach.* A modest example of this approach is the Krovetz stemmer [16]. The approach is also recognised as dictionary-based stemmers. It first gets rid of the suffix and then returns the root of a word via the checking process in a lexicon for any recording. The lexicon similarly completes any alterations which are necessary as a result of spelling exclusion and transfigures a stem generated into an actual word whose denotation is comprehended.
   3.2. *Corpus-based stemming approach.* This approach is suggested by Xu and Croft [13]. They used word variants concurrence to overcome the limitations of the Porter stemmer.
   3.3. *Context-Sensitive stemmer approach*. This is suggested by Peng et al. [17]. The stemming is performed through statistical modelling on the query side. This is very different from the typical technique in which stemming

is performed before indexing a document for a web search. They proved that the number of bad expansion is reduced, and this will improve the precision all at once.

3. Related works

Generally speaking, stemmers have mainly been developed for the English language. The Porter stemming approach [1,2] is used to remove suffixes from a word form in anticipation that the 'stem' will be found. Thus, it is designed for enhancing the performance of informational retrieval systems, if selected word groups are conflated into a unique 'root' of a word. The aim of stemming is to reduce the size and complexity of the data in a text document. However, in some cases, the stemming process cannot produce a correct stem in terms of linguistics. It is worth noticing that due to the progress of the World Wide Web, and the increased number of non-English users, many research efforts for developing stemming approaches for other languages, in particular the Arabic language, have been presented [3,18–20]. Those approaches are applied via truncating both prefixes and suffixes from words to yield the roots of these words.

Nevertheless, few stemming efforts in the Kurdish Sorani dialect have been attempted. Recently, by Salavati et al. [21], a stemming approach for two dialects of the Kurdish language, Sorani and Kurmanji, was developed, but only for getting rid of suffixes. The main aim of this article is to suggest a simple stemmer approach for Kurdish Sorani texts and provide a list of stop words. The list of stop words contains non-useful words, which are deleted from a document after beginning the stemming process. The Kurdish stemming procedure attempts to remove inflectional and derivational affixes (prefixes, suffixes and postfixes) to combine word variants into the same stem root.

4. General Kurdish formulation

The syntactic structure of the Kurdish Sorani dialect depends on the basic word or root. The root word produces both nouns and derivational verbs via adding affixes to the base (root). It is worth mentioning that affixes in the writing system of Sorani can be in the form of nouns that are attached to their roots; for instance, 'دارتاش' means carpenter, which can be further described as 'تاش' which is the root and 'دار' which is the noun. Or prepositions can be attached to roots such as 'ياريدا', which means in play, which can be further explained as 'يارى' which is the root and 'دا' which is the preposition 'in'. Generally speaking, in the Kurdish language, affixes can take three forms:

1. Prefixes are attached to the beginning of a word.
2. Suffixes are attached to the end of a word.
3. Postfixes are attached to the end after suffixes.

Therefore, the words in the Kurdish language can get quite complicated if all these affixes are attached to their roots, for example, a word 'لەياريگايەكان' (layarigakan) means 'from playgrounds'. Table 1 shows a word and its affixes.

Accordingly, from the above example, it can be seen that the Kurdish Sorani dialect has a comparatively complex morphology. Thus, affixes can be removed from a word; then, the stemmed word or the root is produced.

5. Proposed approach for Kurdish Sorani texts

In this section, the proposed Kurdish Sorani stemming approach of the pre-processing stage is presented. Clearly, the complex morphology of Kurdish Sorani makes it very difficult to develop and handle the processing of natural language for digital information retrieval. Accordingly, the stemmer is a tool that can be useful for normalisation and stop word removal. In addition, it is useful for information retrieval in combating the problem of vocabulary mismatch. This problem commonly occurs in numerous applications and can be tackled during the pre-processing stage.

The architecture of the proposed approach for Sorani texts is shown in Figure 2. When the texts are collected, they are sorted and coded in a variety of ways that make evaluation difficult. Thus, before they can be subject to electronic

Table 1. Example of Kurdish affixes

Word	Root (base)	Prefix	Suffix	Suffix	Postfix
لەیاریگایەکان	یاری	لە	گا	یـ	ەکان
Layarigakan	Yari	La	Ga	I	Akan
in play grounds	The word 'play'	From	PlaceMarker	For the existence two vowel(ا and ە)	Plural definite Marker

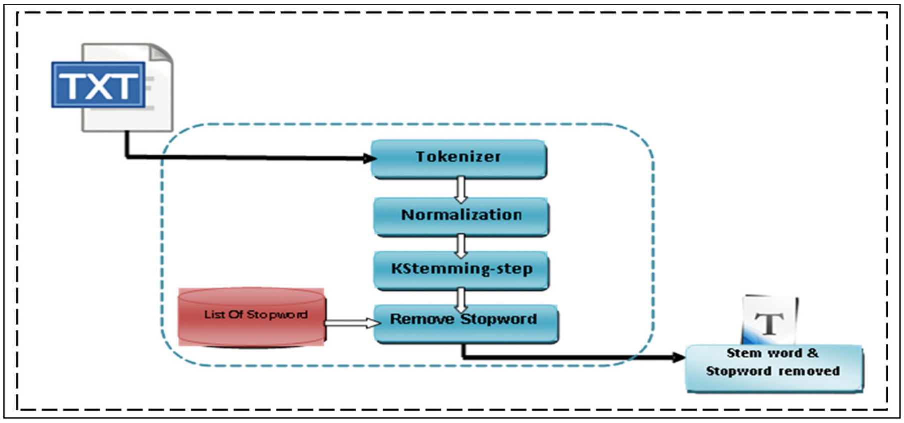
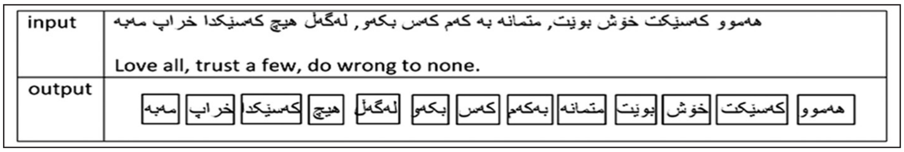


Figure 2. Architecture of Kurdish Sorani text pre-process.



input	هەموو کەسێکت خۆش بوێت, متمانە بە کەم کەس بکەو, لەگەڵ هیچ کەسێکدا خراپ مەبە Love all, trust a few, do wrong to none.
output	هەموو کەسێکت خۆش بوێت متمانە بەکەم کەس بکەو لەگەڵ هیچ کەسێکدا خراپ مەبە

Figure 3. Example of tokenising process.

searching, all documents and texts must be standardised and encoded into Unicode (UTF-8). The proposed approach involves four main steps which can be clarified by the following subsection:

5.1. Tokenisation

Tokenisation is regarded as a basic, and significant step in the natural language process [22,23]. The simple reason for this process is to transform a text stream into tokens by segmenting the texts into smaller units of meaning. The process of tokenisation involves breaking down the sentences in the text document file into words delimited via tabs, different lines or white spaces [22,23].

Tokenisation results in a list of valuable semantic tokens. Therefore, this process helps using a given word in the next phase. Figure 3 shows an example of this stage.

5.2. Kurdish normalisation

The features of Unicode characters are considered a base for the process of normalisation. It is notable that Kurdish writing uses Arabic electronic texts. This will cause differences in letters to some extents. Plainly, this sort of discrepancy issue is generated via typescripts of multiple Unicodes, and eventually, recall in retrieving relevant information is negatively impacted. For that reason, written texts are unified to tackle this issue. This sort of unification improves subsequent steps of pre-processing. Furthermore, this similarity helps to remove affixes and relates words with each other when removing stop words. The main aim of the proposed approach is to prepare consistent words for the next steps in the pre-processing phase. Essentially, the normalisation process is conducted via the following steps to tackle Kurdish Sorani dialect:

1. Replace Arabic letter DOTLESS_YAA (ي) with Arabic letter KURDISH_YEH (ی).
2. Replace Arabic letter ALEF MAKSURA (ى) with Arabic letter KURDISH_YEH (ی).
3. Replace Arabic letter KAF (ك) with Kurdish letter KEHEH (ک).
4. Replace (‘ـه’ which is consisted of ‘ZWNJ[1] + HEH (هـ)’) with Kurdish letter AE (ە).

5.3. Kurdish stemming steps

The proposed approach in this article is used for removing affixes. This is a stemming-step analysis, which is a step-based approach to stages a word goes through before arriving at the extracted root of the word. Some conditions have been

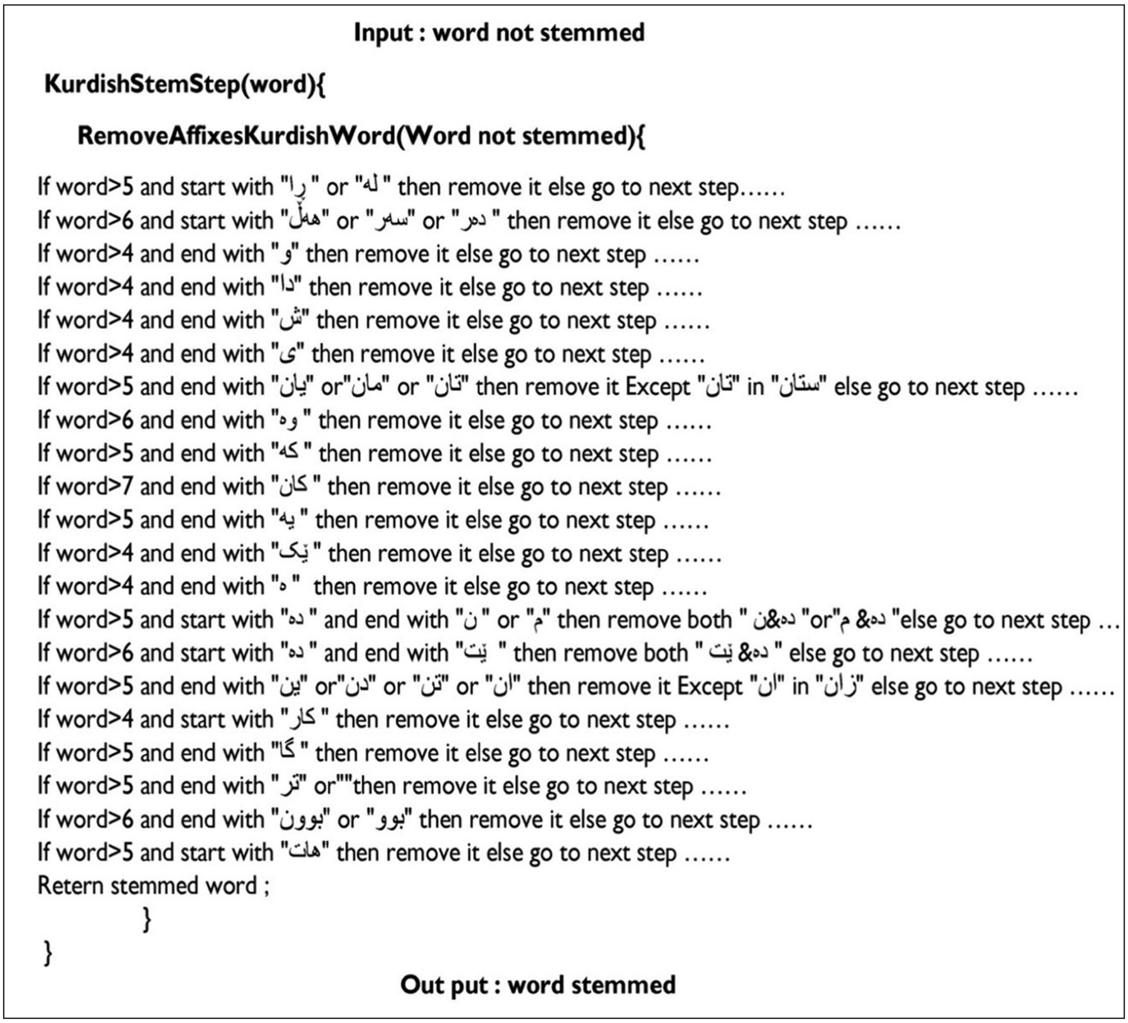
Table 2. Prefixes and suffixes removed by Kurdish stemming steps

Prefixes	Suffixes
ڕا, لە,هەڵ ,دەر , سەر,دە	دا, ش , ی , گا, یان, تان, مان,ەکان, و , ێک , ەکە , یە, وە ,ەوە , تن , ین , ان ,دن , ە ,ن, م, ێت , تر, بوون , کار, هات

Table 3. Example of stop word affixes removal

Stop word	Root of stop word	English meaning
پاشانەوە , لەپاشاندا, لەپاشی	پاشان, پاش	After
ئێوەش, ئێوەی , ئێوەمان,ئێوەیان	ئێوە	Your
چەندین , چەندە	چەند	Some
ئەوەیە , ئەوەی , ئەوەتان	ئەوە	That

Not only these stop words were repeated here, but there are many other stop words in Kurdish Sorani texts that have affixes. These are just examples to clarify.



```
                         Input : word not stemmed
 KurdishStemStep(word){
     RemoveAffixesKurdishWord(Word not stemmed){
If word>5 and start with "ڕا " or "لە " then remove it else go to next step......
If word>6 and start with "هەڵ" or "سەر" or "دەر " then remove it else go to next step ......
If word>4 and end with "و" then remove it else go to next step ......
If word>4 and end with "دا" then remove it else go to next step ......
If word>4 and end with "ش" then remove it else go to next step ......
If word>4 and end with "ی" then remove it else go to next step ......
If word>5 and end with "یان" or"مان" or "تان" then remove it Except "تان" in "ستان" else go to next step ......
If word>6 and end with "وە " then remove it else go to next step ......
If word>5 and end with "کە " then remove it else go to next step ......
If word>7 and end with "کان " then remove it else go to next step ......
If word>5 and end with "یە " then remove it else go to next step ......
If word>4 and end with "ێک " then remove it else go to next step ......
If word>4 and end with "ە "  then remove it else go to next step ......
If word>5 and start with "دە " and end with "ن " or "م" then remove both " دەن& "or"دەم& "else go to next step ...
If word>6 and start with "دە " and end with "ێت  " then remove both " دەێت& " else go to next step ......
If word>5 and end with "ین" or"دن" or "تن" or "ان" then remove it Except "ان" in "زان" else go to next step ......
If word>4 and start with "کار " then remove it else go to next step ......
If word>5 and end with "گا " then remove it else go to next step ......
If word>5 and end with "تر" or""then remove it else go to next step ......
If word>6 and end with "بوون" or "بوو" then remove it else go to next step ......
If word>5 and start with "هات" then remove it else go to next step ......
Retern stemmed word ;
            }
 }
                         Out put : word stemmed
```

Figure 4. The overall process of the Kurdish stemming-step module.

re-adjusted for the letters which are affixes positioned at the end of words in the Kurdish Sorani dialect. Accordingly, this stemmer is specified to the words that have several affixes (e.g. a word that has ‘prefix’+‘root’+‘suffix$_1$’+‘suffix$_2$’+⋯+‘suffix$_N$’). Thus, a given word can go through a group of simplified guidelines depending on conditions to get the root of word. This approach depends on collections of potential prefixes and suffixes, which are commonly utilised in Kurdish text documents. Table 2 demonstrates Sorani prefixes and suffixes which require exclusion.

This approach starts validating a word that ends with affixes. The Kurdish stemming-step module is designed to strip prefixes, suffixes and postfixes from the given word to catch potential roots. The given word will be checked through all the steps in the Kurdish stemming-step process to map the string of letters that are positioned at the beginning or the end of the root of the word. It is worth noticing that the Kurdish stemming-step technique does not use a dictionary for checking the root. For instance, the word (لەهەنگاوەکانیان) is reduced to its root (‘هەنگاو’ means ‘phase’) right through step 1 which removes prefixes (لە) and then goes through the next step for mapping it up until it arrives at step 7 to remove the suffix (یان), then produces (هەنگاوەکان). After that, the approach associates the suffix of the given word with suffixes in the steps followed later to arrive at the step where it finds a match and removes it from the word at step 10. Thus, the suffix (کان) is removed. Finally, in the step 13, the suffix (ە) is matched and then removed. The overall process of Kurdish stemming for a given word can be shown in Figure 4. This approach is not only used to remove affixes from nouns and verbs as it is used in other languages, but it also removes affixes from the stop words which are widely used in Kurdish Sorani (Table 3).

5.4. List of Kurdish stop words

In order to make the environment of information retrieval precise, some of the words that are regarded as not significant must be removed, due to their meanings in the Kurdish sentences. These words are common in Kurdish Sorani texts, and as a result, they intensify the noise of the results. Most of these words are prepositions or pronouns. A predefined list can be prepared to contain these words that do not serve the process of information retrieval, but which are used very regularly in the Kurdish texts (Table 4). The table shows a list that contains nearly 240 stop words, and the list of stop words is developed for two main reasons:

1. The words that correspond between a term and a document need to be kept. This depends considerably on the words that hold essential meaning. Thus, noise words should be removed. Retrieving documents which contain words such as بۆ (meaning to), ئێوە (meaning yours) and نێو (meaning in) in the identical request do not establish a relevant intelligibility. These noise words are non-significant, and they may cause damage to the performance of retrieval, as they do not distinguish between relevant and non-relevant documents.
2. Besides, the richness of the stop words in Kurdish Sorani increases the size of the feature vector. The proposed approach attempts to reduce the size of the file from 35% to 50%.

6. Computational performance

Two sets of data are collected, and different evaluation measures on these data are performed in order to study the performance of the proposed approach. Details of data collection are shown in section ‘Data collection’, and measures are discussed in section ‘Evaluation of the stemmer algorithm’. Then, the data sets are used, and the experimental results are shown in section ‘Experimental results’.

6.1. Data collection

Since there are no standard test collections available for the Kurdish Sorani dialect, the data were collected from two sources: Rudaw, which is a Kurdish daily online news and information about Kurdistan (http://www.rudaw.net/Sorani), and Nrttv, which is another Kurdish daily online news (http://www.nrttv.com). After collecting the data sets from these two sources, one document collection is created, which contains 1960 pieces of text data and a total of 43,594 words. After the tokenising process, it is reduced to 34,099 unique words. As a result of this process, these documents can easily be used for experiments and further study. Two samples of different sizes are created from the collected data:

1. *Sample I.* 34,099 unique words are taken to test stop words in the whole document.
2. *Sample II.* This sample consists of 350 groups of words (root words) and 1702 derivational words. In this sample, the words are linguistically grouped according to their roots taken from the document, and the assessment of grouping is done manually.

6.2. Evaluation of the stemmer algorithm

The evaluation process which is proposed by Paice [24] is performed on the proposed approach. Paice evaluated various English stemming approaches isolated from the context of information retrieval systems. Instead of using the traditional

Table 4. Stop word list for Kurdish Sorani dialect

Alpha	Stop word list
ئـ	."ئەوە", "ئەوان", "ئەو", "ئەوا","ئەم", "ئەنجام","ئێوە", "ئێمە","ئەگەر","ئێستا", "ئەمساڵ", "ئەمە", "ئەمڕۆ", "ئەی","ئەمان"
بـ	"بەو", "بەچەند","بەهۆ", "بەهیچ", "بە", "چۆن", "بەڵکو","بەرەو","بەپێی","بەهۆ","بەم","بەبێ", "بۆئەو","بۆچی", "بۆیە","بۆ", "بەپێی""بوون","بوو","بەڵام","بارە" ,"بەسەر","بەڵێ","بەوە","بکات" ,"بۆچی","باش","بەباش","بان","بکەن","بن", "بێ", ."بووە","بەو","بەوە" ,"بەوان","بێت" ,"بەدەست" "بەردەوام","بڕی","بدات","بڵێ"
پـ	."پاشان", "پاش", "پاشی", "پێش", "پێنج", "پێشتر", "پێشوو", "پلە"
تـ	."تا","تاکو","تاوەکو","تۆ","تایبەت","تر","تری","تەنها","تەنیا","تەواو","تۆن","تیدا","تێدا"
ج	."جار", "جگە","جا","جۆر", "جۆری"
چ	. "چی", "چوار", "چونکە","چونک", "چۆن", "چەند", "چوارەم","چیە"
ح	."حەوت", "حەوتەم"
خ	"خۆ", "خۆی","خۆمان".
د	."دواجار", "دا", "دوو", "دووەم", "دەیەم", "دووشەمم","دوای", "دەڵێت","دیکە", "دەبێت","دەکات", "دوا"
ڕ	. " ڕۆژ","ڕۆژان"
ز	. "ژێر" .,"زۆر","زۆرە", "زوو", "زووی", "زۆردەبن", "زیاتر", "زۆرب", "زۆربە"
س	. "سەد", "ساڵ","ساڵە", "سەدا", "سێ", "سەر", "سەرەتا", "ساڵی", "سبەی","سەرجەم"
ش	. "شوبات", "شەش", "شوێن", "شێوە", "شەشەم"
ق	."قۆناغ"
ک	"کەس", "کانون",,"کەچی", "کەئەو", "کە", "کاتی", "کات", "کەبۆ", "کەوات","کەم","کەمی","کەمە"
گ	."گەر", "گەورە"
ل	"لەسەر", "لەگەڵ", "لە", "لەم", "لەو", "لەوە", "لەبەر", "لەبن","لەوان","لەهەر","لەوکات","لەدوا", "لەنێو", "لەژێر", "لای", "لایەن", ."لەناو", "لێ","لەچی"
م	. "مەبەست", "من", "ملیۆن", "مان","مانگ",
ن	"نەک", "نیە", "نۆ", "نابن", "نەکەن", "نەوەک","نێو", "نێوان","نا", "نەبێت","نیشان", "نییە", "ناو", "ناوبراو", "ناوی", . "نوێ","نەخێر", "نە","نەچێ", "نەبوو"
و	."و", "وەی", "وە", "وەرز","وا", "وەک", "وەکوو","وەکو", "وەها", "وایە"
هـ	"هەینی", "هەبوو", "هەیە", "هەموو", "هەر", "هات", "هۆکار", "هۆ", "هەرچەند", "هەزار", "هەن", "هەند", "هۆی", "هیچ", "هێند", "هەفت", "هەریەک" ,"هەبێت" ,"هەروەها", "هەمان", "هێڵی", "هەردوو", "هەردوولا", "هەزارها", "هاوکات", "هەشت", "هەشتەم", . "هەتا", "هتد", "هەردەم","هەبێ", "هەمووکات". "هیچی", "هاتو"
یـ	. "یان", "یەکەم", "یە", "یی", "یەک", "یەکەمجار", "یا"

The table contains nearly 240 stop words which are arranged alphabetically for documentation.

precision, or recall, parameters, he relied on new parameters; namely, the over-stemming index (OI) and the under-stemming index (UI), their ratios and stemming weight (SW) [25,26]. To test the stemmer, sample II is generated from different words partitioned into concept groups, each of which contains forms, which are both morphologically and semantically related to one another. A flawless stemmer can stem all words in a group to the same stem, and that stem must not then be found in any other group. An under-stemming error can occur if a stemmed group has more than one unique stem. This corresponds to an undesirable outcome on recall in information retrieval systems. By the same token, over-stemming errors can occur if a stem of a particular group also happens in other stemmed groups, in which precision is reduced. Accordingly, it is preferable to have a stemmer that can feasibly generate as few under-stemming and over-stemming errors as possible [26]. For each concept group *g*, two totals can be calculated:

1. A flawless stemmer has to merge every member of a concept group with each other. The total number of different possible words form pairs in the particular group describes the desired merge total (DMT). This can be expressed as follows

$$DMTg = 0.5Ng(Ng - 1) \tag{1}$$

where *Ng* represents the number of words in the group.

2. A flawless stemmer would not unify any member of the present concept group with any word that is not in the group. Consequently, a desired non-merge total (DNT) which counts the possible word pairs formed by a member and a non-member word for every group can be expressed as follows

$$DNTg = 0.5Ng(W - Ng) \tag{2}$$

where *W* represents the total number of words. Ultimately, both the global desired merge total (GDMT) and global desired non-merge total (GDNT) can be obtained right after summing the DMT and DNT of all groups in the word sample. It is found that some of the groups still contain two or more distinct stems when a stemmer to the sample group is applied. Thus, there are under-stemming errors to be counted in such groups. Assume that a concept group of size *Ng* contains *s* distinct stems after stemming, and the number of instances of these stems is U_1, U_2 and U_3, respectively.

The unachieved merge total (UMT) counts the number of under-stemming errors for that group can be expressed as follows

$$UMTg = 0.5 \sum_{i=1}^{s} Ui(Ng - Ui) \tag{3}$$

The global unachieved merge total (GUMT) is obtained by summing the UMT for each group. The UI is given by

$$UI = \frac{GUMT}{GDMT} \tag{4}$$

Over-stemming errors can be caused when a stemming might find cases where the same stem occurs in two or more different groups. Part of two or more different concept groups in any stem group can contain over-stemming errors which are needed to be counted through the wrongly merged total (WMT). The WMT value of that group can be zero if a group does not contain over-stemming errors. Assume a stem group of size *Ns* items which can be derived from *t* different concept groups, and the number of each original concept group within this stem group can be represented via V_1, V_2, V_3,…, V_i as shown below

$$WMTg = 0.5 \sum_{i=1}^{t} Vi(Ns - Vi) \tag{5}$$

The global wrongly merged total (GWMT) is obtained via summing the WMT for each group. The OI can be expressed as follows

$$OI = \frac{GWMT}{GDNT} \tag{6}$$

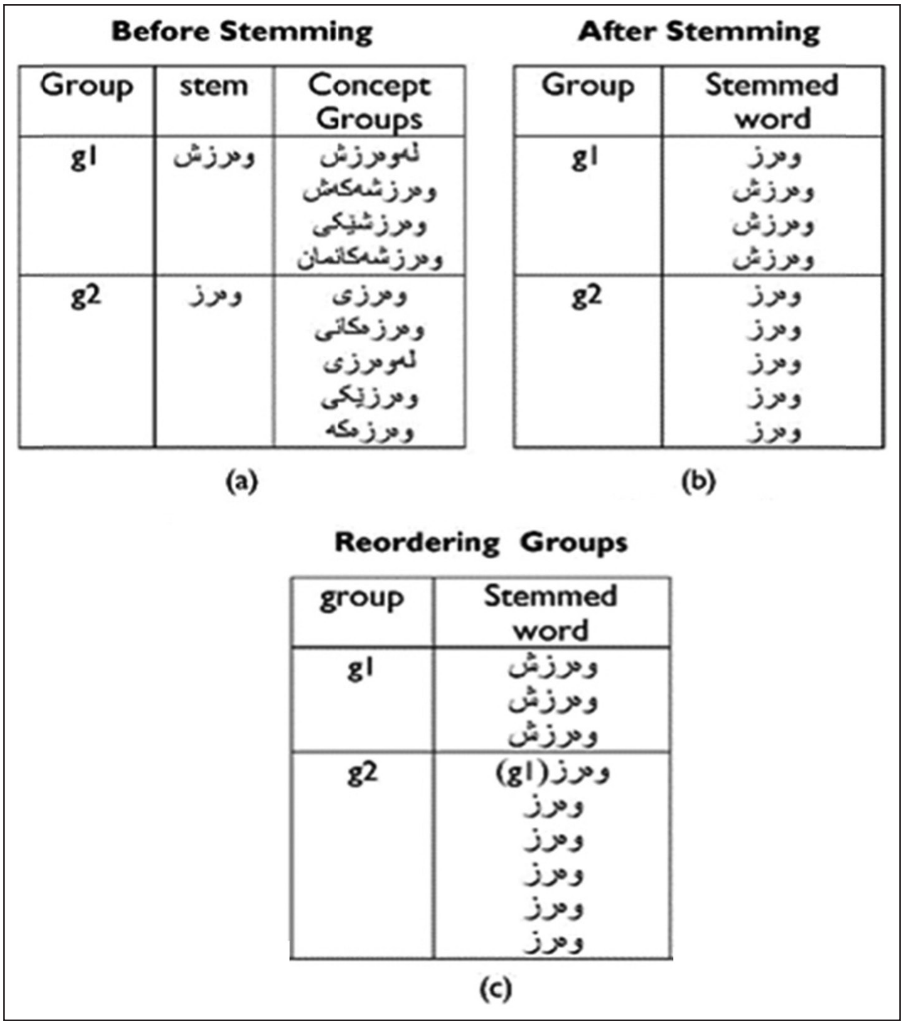
Clearly, UI and OI should be low for a heavy stemmer. The SW is defined as the ratio of these two

$$SW = \frac{OI}{UI} \tag{7}$$

where *SW* is used as a parameter to measure the strength of a stemmer. A stemmer is weak when the value of *SW* is low and the stemmer is strong when *SW* is higher. Figure 5 illustrates how this evaluation method works.

6.3. Experimental results

The above sample from two sources was taken and used for assessing the quality of the suggested stemmer and stop word list. As part of the pre-processing steps, removing stop words from documents before and after applying the stemmer is assessed. First, sample I of data collection is taken to measure how many stop words are deleted correctly from documents. The results of the experiments of sample I can be seen in Table 5, which shows the average number of words per document is much reduced after stemming. This shows that the removal of the highly frequent stop words from the text after applying the stemmer is increased by 30%. Therefore, it can be suggested that the stop word list can be likewise useful in pre-process steps to improve retrieval evaluation [27]. Figure 6(a) describes the stop words that were removed from the document before and after stemming. These words are non-informative data in the documents and affect the process accuracy and retrieval. They detract from the effectiveness of the process. So, from here, it can be observed that this approach has an impact on the pre-processing steps to reduce the dimensions of feature space by eliminating noisy, irrelevant and non-informative data while retaining relevant and informative items. In another experiment on sample I, the effectiveness of grouping various types of a word to the same stem and word reduction in a document is evaluated after removing all stop words. In sample I, 14,158 unique words are obtained. This process is used to reduce the non-conflated words from conflated words. For example, if the document contains the words ‘playing’, ‘played’ and ‘player’, they are reduced to one



Before Stemming

Group	stem	Concept Groups
g1	وەرزش	لەوەرزش وەرزشەکەش وەرزشێکی وەرزشەکانمان
g2	وەرز	وەرزی وەرزەکانی لەوەرزی وەرزێکی وەرزەکە

(a)

After Stemming

Group	Stemmed word
g1	وەرز وەرزش وەرزش وەرزش
g2	وەرز وەرز وەرز وەرز وەرز

(b)

Reordering Groups

group	Stemmed word
g1	وەرزش وەرزش وەرزش
g2	وەرز(g1) وەرز وەرز وەرز وەرز وەرز

(c)

Figure 5. Illustration of the Paice evaluation method: (a) words grouped semantically, (b) after stemming process, *UI* = 0.18 and (c) reorganise the sample into groups that share the same stem, *OI* = 0.25.

Table 5. Stop words removed from documents test on sample I

Analysis of stemmers	Without stemming	With stemming
Total no. of word	34,099 unique word	29,608 unique word
Total no. of stop words removed	13,986 unique word	9495 unique word

word (the root), which is 'play'. The Kurdish stemming-step module achieved 58% of word reduction. This process affects a high dimension of features and thus in turn affects the accuracy in the classification process as well as the retrieval of the text. Figure 6(b) shows the number of words without stemming and with stemming using the Kurdish stemming-step module. Word reduction is calculated via the following equations

$$WordReduction\% = \frac{total\ number\ of\ uniqe\ word\ after\ stemmed}{total\ number\ of\ uniqe\ word} \times 100 \tag{8}$$

In addition, the Paice evaluation method (group collections) is used to evaluate sample II, which has been prepared to be consistent for this sort of evaluation. Table 6 contains the test results of the over-stemming, under-stemming and the weight measures. Although the results indicate that the Kurdish stemming-step formulation is very strong and effective with inflection and derivation, it still transforms some of the root words to incorrect stems, which is caused by both over-stemming and under-stemming errors. Overall, the occurrence of over-stemming errors is greater than under-stemming

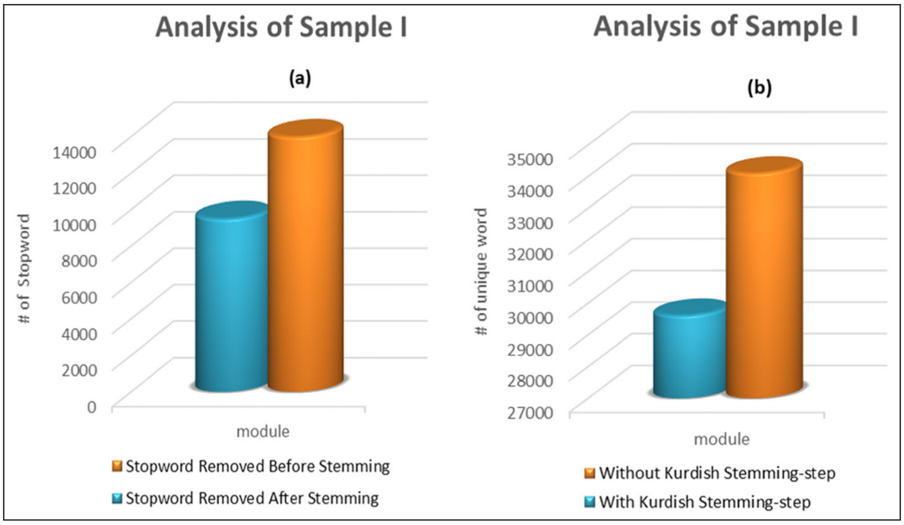


Figure 6. The experimental results of Kurdish stemming-step module: (a) the removal of stop word and (b) the word reduction.

Table 6. Results of the Paice evaluation method using sample II

Kurdish stemmer-step	Result
Over-stemming (OI)	0.34
Under-stemming (UI)	0.11×10^{-6}
Stemming weight (SW)	0.32×10^{-6}

errors. The test shows that the Kurdish stemming-step module is strong, and this appeared obviously on the SW, which is a large value. A weak stemmer produces more under-stemming than over-stemming errors, and a strong stemmer does the reverse [26].

Finally, sample II is tested after applying the Kurdish stemming-step formulation to measure how many words are stemmed correctly, which is calculated by counting the correct stem words coming from the Kurdish stemming-step module, then dividing them by the whole number of words in the collection and then finding the percentage. The Kurdish stemming-step formulation produces better indicators of correctly stemmed words and combining variant words of the same group to a correct stem; it reaches nearly 78% correct stems.

Further testing to evaluate our approach in terms of its aid to information retrieval should measure the performance of the information retrieval system and compare its result with the results obtained from using both the exact-match (without stemming) and the Kurdish stemming-step module. The performance is measured by the recall and precision estimations, which are constituted in the following equations

$$Precision = \frac{|\{relevant\ documents\}\ \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|} \tag{9}$$

$$Recall = \frac{|\{relevant\ documents\}\ \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|} \tag{10}$$

$$F\ \ measure = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right) \tag{11}$$

Kurdish stemming-step module was tested on the collected documents which contain 1960 pieces of text data. The system is tested with 10 queries for searching through the collected documents. Thus, using information retrieval evaluation, which is mentioned above, the precision and recall results for each query point without stemming (exact match) and with stemming using Kurdish stemming-step module were calculated. The average recall for the Kurdish stemming-step module is 93%, which is the highest value in comparison with the one without stemming (66%). Kurdish stemming-step module achieved an average F measure of 0.89 which is showed a superiority over the without stemming. However,

Table 7. Average precision, recall and F measure

Analysis of stemmer	Precision	Recall	F measure
Without stemming	0.82	0.66	0.73
Kurdish stemming-step module	0.86	0.93	0.89

without stemming (exact match) achieved an average F measure of 0.73 with a very low recall average value where it could not retrieve all the documents in many case. These results are shown in Table 7.

Thus, this approach would be a good choice to get correct stems so as to increase search precision. It can be inferred that the Kurdish stemming-step module improves the recall and precision over the module without stemming. An ideal stemmer is a stemmer with low under-stemming and over-stemming errors. Yet, the main problem is that both errors conflict with each other; reducing one type of error can lead to an increase in the other. Heavy stemmers reduce the under-stemming errors while increasing the over-stemming errors, but light-stemmers reduce the over-stemming errors while increasing the under-stemming errors, although heavy stemmers reduce the size of the corpus significantly. For a heavy morphological stemmer, Kurdish stemming-step module considered as a heavy morphological stemmer specifically for pre-processing steps. Thus, based on all the results put forward in above tests, it appeared to be very distinct than other information retrieval–developing approaches and also yielded such relevant and better results than traditional system without stemming. Thus, the overall study led to a better, effective, efficient, reliable, relevant and excellent information system which can be user-friendly and applied anywhere on textual data sets for ease of data handling, management and access through retrieval. In near future, the same system will be examined with various data sets for more assertive solutions and conclusions.

In most previous studies, the effectiveness of stemming algorithms has been measured by two distinctive ways: (1) how precisely they map variation types of a word to the same stem; (2) how much change they bring to information retrieval systems [28,29] in terms of indexing and searching. This work evaluates performance in terms of accuracy by checking the quantity of identifiable mistakes amid the stemming of words from different content examples. This involves manual gathering of the words in every example, not to mention, programming has been created to encourage this. Next, the words are stemmed and files are then registered which represent the rate of under-stemming and over-stemming errors. These kinds of evolution are used in other languages as in previous works [29–32].

7. Analytical discussions

As can be noticed from previous sections that the strategy in this article is quite different. The way this article deals with the Kurdish Sorani texts includes a form integral to the process of pre-processing to make it more accurate in terms of performance. Thus, the following points can be discussed:

1. Subsequent number that comes before and after the word varies, and as the deletion of this number is different, this has also reminded us that the Kurdish language contains the subsequent after which it can be one or more called postfixes are attached to the end after suffixes. In addition, the use of suffixes in the work by Esmaili et al. [7] is different from this research work. Table 1 in this article is an example which explains the affixes removed from a word. The Kurdish stemming-step module is designed to strip prefixes, suffixes and postfixes from the given word by steps until to catch potential roots. Comparing this work with the work by Esmaili et al. [7], the affixes are presented in their paper just those that are stripped from a word, not even until catching potential roots. Also, observing the main stemming algorithms in the literature, we found that stemming algorithms are based on the best-known affixes and the most used ones in the morphology of the Kurdish Sorani dialect will be more efficient than those based on the top most-frequent *n*-grams in each set to extract the meaningful affixes.
2. Unlike the work by Esmaili et al. [7], in this article, the stop words are fewer and documented, certainly, after lengthy study, this is because the stemming is conducted and then stop words are removed. The most important point of this stemming approach is not only used to remove affixes from nouns and verbs as it was used in other languages including in the work by Esmaili et al. [7], but also it removes affixes from the stop words which are widely used in Kurdish Sorani dialect, for example, in Table 3, to make the process of matching words and delete them more effectively. This is not even made in other languages. This has been clarified in the structure of the work steps in Figure 2, and the results have been clarified in Table 5 which refers to the stop words that have been deleted before and after the operation of Kurdish stemming steps and the difference is very obvious.

3. Furthermore, the normalisation process is also different from that mentioned in the work by Esmaili et al. [7] in the number of letters that has addressed as well in the positions.

In the work by Singh and Gupta [33], the authors tried to explain the important aspects of text stemming and provided an extensive and useful understanding of stemming techniques. However, they kept the analysis of the current stemming techniques open in order to help researchers to think about the new lines for this research field in the future for languages such as English and alike that have a large number of researchers. Thus, the Kurdish language is very new in this field and needs extra attention from the researchers to improve extensively the performance of a pre-processing tool which can be used in the field of natural language processing, information retrieval applications, text classification and text clustering.

8. Conclusion and future work

Good stemmers can have a large impact on text classification and information retrieval. In this article, a Kurdish stemming-step approach was introduced. This is a first attempt to stem Kurdish Sorani words. It is a technique that connects morphologically-related indexing and search terms in Kurdish texts. In actual fact, the occurrence of words correspondingly displays a supportive role for the classification process. The handling of similarity changes was added, which helped to increase matching among words and diminish the storing requirements, even though it was anticipated to create more or less errors to display the complication and difficulty of words in the Kurdish Sorani dialect. Nevertheless, this stemmer resolved most of these problems. Kurdish Sorani texts comprise a wide range of stop words with attached affixes, as a result, it can be stemmed by merging these stop words that are frequently occurring. This is better for the pre-processing phase instead of using full words. This work can be extended in the future to apply to larger Kurdish affixes and to implement this approach for other Kurdish Sorani text classification to sort documents into a variety of categories (e.g. art, sports, religion and economy).

Acknowledgements

The authors thank the editor of the journal for reviewing of the manuscript. His explanations, comments and support are greatly appreciated. The authors also thank Ms Karen Nelson, a Research and Instruction Librarian from Davenport University in Grand Rapids, Michigan, and Mr Shalaw Najat Ghani (MA in TESOL), from Valparaiso University, for their continuous effort in editing the manuscript. They did a very good job, not only correcting the English language but also going deeply into the sense of the manuscript.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research received a student specific grant from School of Computer Science, College of Science, University of Sulaimania, Sulaimania, Kurdistan, Iraq.

Notes

1. Zero width non-joiner is denoted as ZWNJ, which is a non-printing character used in the computerisation of writing systems that make use of ligatures, when placed between two characters that would otherwise be connected into a ligature.

References

[1] Porter MF. An algorithm for suffix stripping. *Program* 1980; 14: 130–137.

[2] Porter MF. Snowball: a language for stemming algorithms, 2001, http://snowball.tartarus.org/texts/introduction.html

[3] Khoja S and Garside R. Stemming Arabic text. 1999, Lancaster: Computing Department, Lancaster University, http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps

[4] Tashakori M, Meybodi M and Oroumchian F. Bon: first Persian stemmer. In: Shafazand H, Min Tjoa A (eds.) *Lecture notes on information and communication technology (LNCS)*. Berlin: Springer, 2002, pp. 487–494.

[5] Naouar F, Hlaoua L and Omri MN. Possibilistic model for relevance feedback in collaborative information retrieval. *Int J Web Appl* 2012; 4(2): 78–86.

[6] Boukhari K and Omri MN. Robust algorithm for stemming text document. *Int J Comput Inform Syst Ind Manage Appl* 2016; 8: 235–246.

[7] Esmaili KS, Salavati S, Eliassi D, et al. Building a test collection for Sorani Kurdish. In: *Proceedings of the 10th ACS/IEEE international conference on computer systems and applications (AICCSA'13)*, Ifrane, Morocco, 27–30 May 2013. New York: IEEE.

[8] Kurdish Academy of Language. Kurdish language, http://www.kurdishacademy.org/?q=node/4 (accessed 15 February 2012).

[9] Thackston WM. Sorani Kurdish: *a reference grammar with selected readings*. Cambridge, MA: Harvard University Press, 2006.

[10] Walther G. Fitting into morphological structure: accounting for Sorani Kurdish endoclitics. In: *Proceedings of the 8th Mediterranean morphology meeting*, 2012, pp. 299–321, http://geraldinewalther.net/Geraldine_Walther/Publications_files/MMM8_GWalther.pdf

[11] Samvelian P. A lexical account of Sorani Kurdish prepositions. In: Muller S (ed.) *Proceedings of international conference on head-driven phrase structure grammar*. Stanford, CA: CSLI Publications, 2007, pp. 235–249.

[12] Tair MA and Baraka RS. Design and evaluation of a parallel classifier for large-scale Arabic text design and evaluation of a parallel classifier for large-scale Arabic text. *Int J Comput Appl* 2015; 75(3): 12–20.

[13] Xu J and Croft WB. Corpus-based stemming using co-occurrence of word variants. *ACM T Inform Syst* 1998; 16(1): 61–81.

[14] Lovins JB. Development of a stemming algorithm. *Mech Transl Comput Linguist* 1968; 11(1–2): 22–31.

[15] Jivani GA. A comparative study of stemming algorithms. *Int J Comput Technol Appl* 2011; 2(6): 1930–1938.

[16] Robert K. Viewing morphology as an inference process. In: *Proceedings of the 16th annual international ACM SIGIR conference on research and development in information retrieval*, Pittsburgh, PA, 27 June–1 July 1993, pp. 191–202. New York: ACM.

[17] Peng F, Ahmed N, Li X, et al. Context sensitive stemming for web search. In: *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, 23–27 July 2007, pp. 639–646. New York: ACM.

[18] Chen A and Gey F. Building an Arabic stemmer for information retrieval. In: *Proceedings of the 11th text retrieval conference (TREC)*, November 2002, pp. 19–22. Gaithersburg, MD: NIST.

[19] Ingason AK, Helgadóttir S and Loftsson H. *A mixed method lemmatization algorithm using a hierarchy of linguistic identities (HOLI)* (ed. A Ranta and B Nordström). Berlin: Springer, 2008, pp. 205–216.

[20] Darwish K. Building a shallow Arabic morphological analyzer in one day. In: *Proceedings of the workshop on computational approaches to Semitic languages in the 40th annual meeting of the association for computational linguistics (ACL-02)*, Philadelphia, PA, 7–12 July 2002, pp. 47–54. New York: ACM.

[21] Salavati S, Esmaili KS and Akhlaghian F. Stemming for Kurdish information retrieval. In: *Proceedings of the 9th Asia information retrieval societies conference*, Singapore, 9–11 December 2013, vol. 8281, pp. 272–283. Berlin: Springer.

[22] Jayanthi R. An approach for effective text pre-processing using improved Porters stemming algorithm. *Int J Innov Sci Eng Technol* 2015; 2(7): 797–807.

[23] Dilekh T and Behloul A. Implementation of a new hybrid method for stemming of Arabic text. *Int J Comput Appl* 2012; 46(8): 14–19.

[24] Paice CD. An evaluation method for stemming algorithms. In: *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*, Dublin, 3–6 July 1994, pp. 42–50. New York: ACM.

[25] Kraaij W and Pohlmann R. Porter's stemming algorithm for Dutch. In: *Informatiewetenschap 1994: Wetenschappelijkebijdragenaan de derde STINFON Conferentie* (ed. LGM Noordman and WAM de Vroomen), 1994, pp. 167–180. Tilburg, Netherlands.

[26] Karaa WBA and Gribâa N. Information retrieval with Porter Stemmer: a new version for English. In: Nagamalai D, Kuma A and Annamalai A (eds) *Advances in computational science, engineering and information technology*, vol. 225. Heidelberg: Springer, 2013, pp. 243–254.

[27] Hmeidi I, Al-Ayyoub M, Abdulla NA, et al. Automatic Arabic text categorization: a comprehensive comparative study. *J Inform Sci* 2014; 41(1): 114–124.

[28] Flores FN, Moreira VP and Heuser CA. Assessing the impact of stemming accuracy on information retrieval. In: *Proceedings of the international conference on computational processing of the Portuguese language*, Porto Alegre, Brazil, 27–30 April 2010, pp. 11–20. Berlin: Springer.

[29] Paice CD. Method for evaluation of stemming algorithms based on error counting. *J Am Soc Inform Sci* 1996; 47(8): 632–649.

[30] Gupta V, Joshi N and Mathur I. Design & development of rule based inflectional and derivational Urdu stemmer 'Usal'. In: *2015 International conference on futuristic trends in computational analysis and knowledge management (INBUSH ERA-2015)*, Noida, India, 25–27 February 2015, pp. 7–12. New York: IEEE.

[31] Thalji NJ. Developing an effective light stemmer for Arabic language information retrieval. *Int J Comput Inform Technol* 2016; 5(1): 55–59.

[32] Kumar D and Rana P. Stemming of Punjabi words by using brute force technique. *Int J Eng Sci Technol* 2011; 3(2): 1351–1358.

[33] Singh J and Gupta V. A systematic review of text stemming techniques. *Artif Intell Rev*. Epub ahead of print 1 August 2016. DOI: 10.1007/s10462-016-9498-2.