FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 6 No. 1, March, 2022, pp 291 - 295
DOI: https://doi.org/10.33003/fjs-2022-0601-899

AN IMPROVED HAUSA WORD STEMMING ALGORITHM

***Sirajo Musa, G. N. Obunadike, & Muhammad Muntasir Yakubu**

Faculty of Computer Science and Artificial Intelligence, Federal University, Dutsin-Ma

*Corresponding authors' email: **sirajomusa19@gmail.com**

ABSTRACT

The explosion of scientific publications in different domains coupled with the introduction and socialization of the internet experienced in the last few decades has made information more available than ever before. Consequently, digital storage capacity has been consistently doubling to reflect this geometric increase in information. In view of this, Information Retrieval (IR), nowadays considered the dominant form of information access has become even more critical. However, the problem of using free text in indexing and retrieval arising from spelling mistake, alternative in spelling, affixes and abbreviations has continued to bedevil the field of IR. To mitigate this problem, Stemming Algorithm was introduced in the 1960s. Stemming is an automated process of stripping all word derivatives of their inflectional affixes in order to obtain stem of the word. Because stemming is language specific, there are stemming algorithms designed specifically for most of the major languages in the world. With a speaker population of about 150 million Hausa language stands in need of a better stemming algorithm. This research is an attempt to improve upon the existing Hausa word stemming algorithm. Affix stripping method of conflation with reference lookup was used. Using Sirsat's evaluation method, this research achieved 96.9% as Correctly Stemmed Word Factor (CSWF), Index Compression Factor – 74.76%, Words Stemmed Factor (WSF) – 70.44% and Average Word Conflation Factor – 59.47%.

Keywords: Hausa Language, Information Retrieval, Natural Language Processing, Stemming

INTRODUCTION

The primary objective of Information retrieval is to analyse documents and extract information that satisfies the user's needs. Information needs refers to high level concepts that describe the material of interest presented by the user in the form of direct questions such as "what is stemming" or direct term as "stemming" or when the user is unsure and just say "an algorithm for reducing words to their stems". These questions are termed as query, and they are user's input to the Information Retrieval System. The System's output will be document titles or a set of document titles which are ranked according to relevance of the document to the user's query and in accordance with the logic of such ranking. However, words in natural language are characterized by various morphological variants which leads to vocabulary mismatch thereby making it difficult to retrieve relevant information. To address this concern, stemming is employed. According to Lovins (1968), Stemming is "a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes" (Lovins, 1968). Stemming could be manual or automatic. When a regular expression is used to achieve this objective, stemming is manual. Stemming is said to be automatic when stemming algorithm is utilised.

Lovins (1968) pioneered the work on stemming algorithm. She introduced an algorithm which used a lookup on a table of 294 endings, 29 conditions and 35 transformation rules arranged on a longest match principle. The Stemmer removes the longest suffix from a word. Immediately the ending is removed, the word is recorded in a different table that make several adjustments to convert the stem into a valid word. Owing to its nature of single pass, the algorithm removes maximum of a single suffix from a word. The algorithm is fast and can handle removal of double letters from a word. Rakesh, K., & Vibhakar, M. (2016). For instance the word "getting" is stemmed as "get" thereby getting rid of the double 't' in a single operation. It can also handle irregular plurals like "mouse" and "mice", "indexes" and "index" etc. However, despite these capabilities, the stemmer was criticized for being too complex for its time and having overheads for time and using too much space. A. Ismalov et. Al (2016). It also has the shortcoming of missing many affixes in the table of endings. This is attributed to the technical vocabulary used by the author. Owing to these draw backs, several stemming algorithms have been developed subsequently, like Dowson (1974), Porter (1980) and Paice/Husk (1990) etc.

Dowson (1974) introduced his stemmer. This Stemmer which is an extension of Lovins (1968) Stemmer has more comprehensive list of suffixes (about 1200). Dowson's stemmer, like its predecessor was also a single pass and therefore very fast algorithm. The suffixes are stored in reverse order indexed by their length and last character. Dowson Stemmer is fast in execution, covers more suffixes than Lovins but very complex and lack standard reusable implementation.

Porter (1980) introduced his Affix Stripping Stemmer with a view to correcting the shortcomings of the Lovins (1968) and Dowson (1974) algorithms. The algorithm consistst of 5 steps and a series of 60 rules which use partial matching concept to reduce the complexity of the algorithm. In the overall, Porter Stemmer is a huge improvement over its predecessor (Lovins Stemmer). However, because Porter Stemmer consists of 5 steps compared to that of Lovins which has only 2, Porter stemmer is slower in performance. Because of its popularity and simplicity, several researches have been conducted in different languages applying Porter's suffix stripping stemmers.

Paice/Husk Stemmer (Paice, 1990) introduced an iterative stemming algorithm using the same rules and suffixes in every loop.

Malay stemmer was first introduced by Darwis et al. (2012). He utilized Malay word register to handle ambiguity problem and to use an extensive affix stripping approach to eliminate or drastically reduce understemming and overstemming errors. This stemmer consists of 121 affix-stripping rules. Several other stemmers were attempted to improve existing Malay stemmer. This approach achieved a success of 99.8%.

Dictionary-based stemming, light-based stemming and unsupervised methods have been used to stem Arabic word. In dictionary-based approach, the root word is determined based on linguistic lexicons. Three stages are employed viz; pre-processing, entity recognition and stem identification. In order to identify and remove redundant spaces, sentence boundaries are marked during the pre-processing stage. The text to be stemmed is then tokenized upon which it is sent to the morphological analyzer. The entity recognition stage recognises Arabic personal names. Stem is finally arrived based on suffix, prefix and stem. The light-based approach consists of 3 stages viz: word identification, word segmentation and pattern matching. In word identification, common words and non-derivational nouns are identified. In word segmentation, the words are divided into their components (stem and affixes are separated); affix, stems and clitics based on some rules. The words are finally extracted by matching them with some patters in the final stage (pattern matching). An enhanced Arabic Stemming algorithm has been proposed by Juzaiddin and Aziz (2011) and Alhanini et al (2011). The aim of this algorithm was to ameliorate the inadequacies of light-based and dictionary-based stemmers developed thus far. This stemmer has 96% accuracy compared to light-based and dictionary-based which have 85% and 88% respectively.

Stemming Hausa word begun in 2015 with the work of Muazzam et al. (2015) which was augmented by the work of Bimba et al. (2015). Muazzam et al. (2015) modified Porter algorithm with rules that handle exceptional cases that occur in the language. Bimba et el. (2015) used 78 affix stripping rules applied in four steps and a reference lookup to improve recall and precision of the work of Muazzam et al. (2015). Accuracy of 78.3% and 87.3% was achieved by Muazzam et al. (2015) and Bimba et al (2015) respectively.

Information Retrieval

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers). Most of the times, the information that will meet the need of the users is buried in tons of other (irrelevant) information and the only way to get it is to search for it. Unfortunately natural language, unlike computer language, is complex and ambiguous. Words often have morphological variants which results in vocabulary mismatch. Same word may have different suffixes thereby making it difficult to the IR to identify and recognize it as a match. In addition to this, use of abbreviation, difference is spelling styles (U.S. and U.K.) also contribute to vocabulary mismatch. To minimize vocabulary mismatch and reduce size of index file, stemming algorithm is used. With stemming, morphologically variant words are conflated and represented as a single term. For example “retrieve”, “retrieves”, “retrieval”, “retrieving” and “retriever” are all stemmed to “retriev”. The resultant word may not be in the dictionary of the language. It is however acceptable provided it will aid in the process of conflating the related terms.

Information Retrieval Model

The query and relevant documents are represented in the same way in order that documents selection and ranking is formalized by a matching function which returns a retrieval status value (RSV) in respect of each document. Some IR Models are discussed below:

1 Boolean Model:

It is the oldest and simplest retrieval model. This IR model utilizes set theory and Boolean such that documents are treated as set of terms and queries are essentially Boolean expression on the terms. Boolean queries are queries using AND, OR and NOT to join query terms. Each document is viewed as a set of terms which either matches or not the Boolean conditions set in search queries. In this model, documents are not ranked on relevance. Example of application of Boolean Retrieval is email search. Most search engines do not use Boolean model.

2. Probabilistic Model

In probability model, the ranking function is found based on whether a document is relevant to a search query. This is called probabilistic ranking principle. Two main parameters in this model are P(REL) and P(NREL) – probability of relevance and probability of non-relevance of a document. Probability that a document is relevant is given by Equation 1

$$P\left(\frac{REL}{d}\right) = \left(P(REL) * \frac{P\left(\frac{d}{REL}\right)}{P(d)}\right) \quad - \quad (1)$$

In this model, document d is classified as relevant if $P\left(\frac{D}{REL}\right)P(REL) > P\left(\frac{D}{NREL}\right)P(NREL)$

3. Vector Space Model

In this model, the documents and query are represented as a vector and based on their similarity, documents are retrieved. The vectors can be either binary in Boolean VSM or weighted in Non-binary VSM which can be used for ranking.

Stemming

Lovins (1968) defined stemming as “a computational procedure which reduces all words with the same root (or, if prefixes are left untouched, the same stem) to a common form, usually by stripping each word of its derivational and inflectional suffixes“. Stemming can be performed in any of the following four ways:

(i) Affix Stripping

Affix stripping is based on the idea that inflected languages produce inflections on word at the right side of such words to change their meaning. The words “computer”, “computers”, “compute”, “computed” “computing” are all variants of the word “comput”. Rules are therefore made to truncate the suffix in a word so that the root of the word is arrived at. In the example above, if we truncate “ed”, “ing” “d” and “s”, we would be left with “compute”. This word is stemmed as “comput” by Porter algorithm (Porter, 1980). In stemming, the resulting stem needs not be in the dictionary of the language. It is nonetheless sufficient that it has led to the conflation of related terms.

(ii) Table Lookup

In this method, all words and their stems are kept in a table. All terms in query and indexes are stemmed using a look up from the table. The major problem of this method is storage overheads and extensive use of a particular language to the stems of each word.

(iii) N-gram

An n-gram is a sequence of consecutive characters extracted from a word. The more similar words are the higher the proportion of n-grams they will have in common. The items can be syllables, letters etc.

(iv) Successor Variety

This method depends on the frequency of letter series to segment a word into its stem by using a word corpus in the process of stemming. Successor variety of a word is the number of letters that follow it in words in a body of text.

Natural language, unlike computer language is characterised with variants of the same word. Many a times these morphologically variant words have the same semantic interpretations and could be considered same for the purpose of information retrieval. By conflating these terms and representing them with a single stem, index file size and vocabulary mismatch is reduced thereby enhancing efficiency of Information Retrieval.

Errors in Stemming

Because Stemming is based on rules and such rules can fail, errors are sometimes encountered in the process. Two errors are associated with stemming as follows:

(i) Over stemming

When too much of a word is removed, it is said to be over-stemmed. It is also an instance of over-stemming when two words belonging to different stems are stemmed to the same stem thereby causing the conflation of unrelated terms. The consequence on Information Retrieval performance is retrieval of non-relevant materials. Over-stemming is also known as false positive. As an example consider the words “university”, “universities”, “universal” and “universe”. Porter Stemmer stems all these words as “univers”.A better result would be 'universi' for “university” and “universities” and 'univers' for “universal” and “universe”two. Porter stemmer is rule-based and rules can fail in some situations.

(ii) Under stemming

Under-stemming is when too little of a word is removed thereby preventing related terms from being conflated and hence failure to retrieve relevant documents. It is also called under stemming when words that are belonging to the same root are not stemmed so. This is also known as false negative.For example although data and datum are from the stem, they are stemmed as ‘dat’ and ‘datu’ respectively.

METHODOLOGY

This research is an application of Porter Stemmer developed by Martin Porter in 1980 to stem Hausa Words. Similar work had been carried out by Bimba et al in 2015. The material to be stemmed (text) is cleaned of all non-alphabetic and special characters like punctuation marks, numbers, brackets etc. The text is then converted to lowercase and subsequently tokenized.

The reference lookup used by Bimba et al is significantly enhanced in this research. Some root words were added in order to enhance the strength of the stemmer. The use of lookup was found to reduce stemming errors. The stemmer makes heavy use of the lookup. Before invoking virtually every step, a reference is made to the lookup. Hence the need to enrich the lookup.Bimba’s stemmer used 1500 words. Additional root words have been introduced to augment the existing ones especially in areas where ambiguity could be encountered in the language like words starting with ‘ma’ or ‘ba’. The work methodology for this work is s shown in Figure 1.

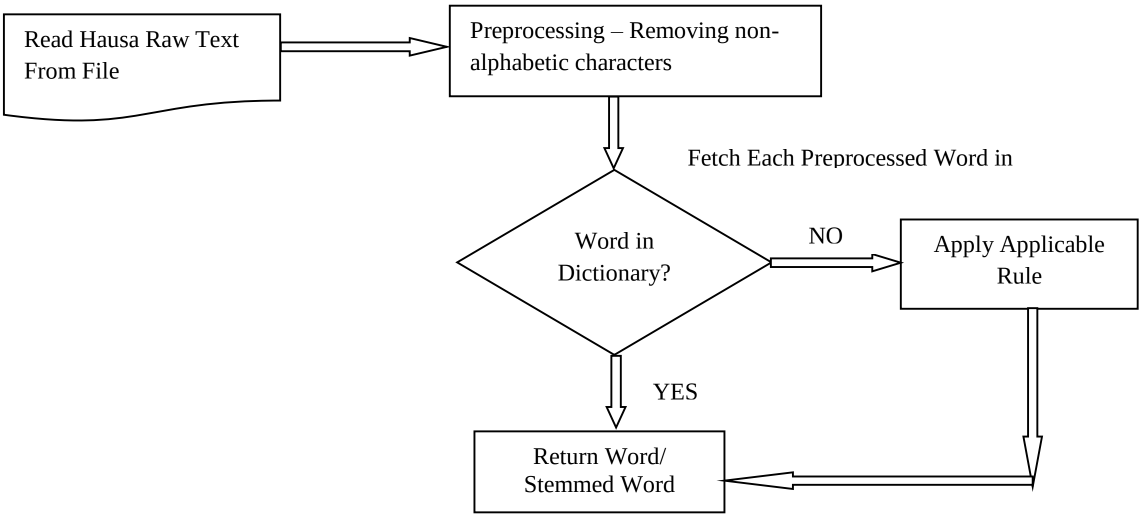
Read Hausa Raw Text From File
Preprocessing – Removing non-alphabetic characters
Fetch Each Preprocessed Word in
Word in Dictionary?
NO
Apply Applicable Rule
YES
Return Word/ Stemmed Word

Figure 1: Hausa Stemmer Methodology Diagram

List of Suffixes

Rules associated with the truncation of each of the following suffixes were specified in the stemmer:

Table 1: List of Suffixes

Sn	Suffix	Sn	Suffix	Sn	Suffix	Sn	Suffix
1	ai	21	mchi	41	obi	61	rko
2	ana	22	mci	42	ochi	62	ru
3	ao	23	mtaka	43	odi	63	sa
4	ar	24	n	44	ofi	64	shi

5	awa	25	nchi	45	ogi	65	suwa
6	aye	26	nci	46	oji	66	suwa
7	cce	27	nda	47	oki	67	ta
8	che	28	ni	48	oli	68	taki
9	chi	29	nka	49	omi	69	udda
10	ci	30	nku	50	ori	70	uka
11	fa	31	nna	51	oshi	71	una
12	fen	32	nni	52	osi	72	uwa
13	gu	33	nnu	53	oti	73	wa
14	ilu	34	nsa	54	owa	74	ya
15	ina	35	nsa	55	owi	75	ye
16	ir	36	nshi	56	oyi	76	yu
17	ka	37	nsu	57	ra	77	mme
18	ke	38	nta	57	rer		
19	ki	39	ntka	58	ri		
20	ku	40	nu	60	rin		

List of Prefixes

Rules regarding the handling of the following prefixes were stipulated in the stemmer

Table 2: List of prefixes

SN	PREFIX
1	Ma
2	Tsatt
3	Yan
4	Mai
5	Ba
6	Ta
7	Abin
8	Wuri-n

```
1    START
2    READ the raw text
3    STRIP the text of all non-alphabetic characters
4    CONVERT the entire text to lower case
5    TOKENIZE text to be stemmed
6    READ NEXT word from tokenized text
7    Length of Word >= 3?
     True: GO TO 8
     Else: GO TO 6
8    If word in Dictionary:
         Return Word and Go next step
     Else:
      Apply Rule_Step_1 To Word
9    If Rule_Step_1 Satisfied:
     IF Word in Dictionary Return Word and GO TO 6
     Else GO TO next step
10   Apply Rule_Step_2 To Word
     If Rule Satisfied:
     If Word in Dictionary Return Word and GO TO 6
     Else GO TO next step
11   Apply Rule_Step_3 To Word
     If Rule Satisfied:
     If Word in Dictionary Return Word and GO TO 6
     Else GO TO next step
12   Apply Rule_4 to Word
     If Rule Satisifed:
     If Word in Dictionary Return Word and GO TO 6
      Else GO TO 13
13   Check if Words Are Exhausted
     If Words Exhausted GO TO 14
     Else GO TO 6
14   STOP
```

Figure 2: Sequence of Stemmer Processing

RESULT DISCUSSIONS

To measure the performance of this stemmer, Sirsat's evaluation method was used. A total of 1786 unique words are used for the analysis and results obtained.

Table 3: Comparative Analysis of Siraj and Bimba Algorithms

	ALGRORITHMS		
FACTOR	**SIRAJ**	**BIMBA**	**IMPROVEMENT**
Total Words	1786	1786	N/A
Number of Words Stemmed (SW)	1258	1213	45
Words Stemmed Factor	70.44%	67.92%	2.52%
Number of Distinct Word After Stemming (S)	1022	1046	-24
Index Compression Factor	74.76%	70.76%	4.00%
Correctly Stemmed Words	1219	1146	73
Incorrectly Stemmed Words	39	66	-27
Correctly Stemmed Words Factor (CSWF)	96.90%	94.56%	2.34%
Correct Words Not Stemmed	528	573	-45
Average Word Conflation Factor (AWCF)	59.47%	50.04%	9.43%

From the results above, it could be seen that Number of Words Stemmed has been increased from 1213 to 1258 (+45), the Word Stemmed Factor has been increased 2.52%. The number of Distinct Words After Stemming has been lowered by 24 – in this case the fewer the better. Similarly, Incorrectly Stemmed Words has been reduced by 27, Correct Words Not Stemmed have also been reduced by 45.

CONCLUSION

In this research, Porter Affix Stripping algorithm was adopted to stem Hausa Words using a set of affix-stripping rules in a four-step sequence. Reference lookup was employed to aid stemmer recall and precision. Additional root words introduced in the research has indicated further improvement in the performance of the stemmer. The base-research has been enhanced with the ability to handle words which hitherto have not been taken care of as seen in the results analysis section. Future researchers may introduce the use of stop words in their research. The root words may also be enhanced to cover wider range of Hausa words. Development of Hausa Corpus will also definitely help the Hausa Stemmer Development initiative.

REFERENCES

Alhanini, Y., Juzaiddin, M., & Aziz, A. (2011). The enhancement of arabic stemming by using light stemming and dictionary-based stemming. *Journal of Software Engineering and Applications*, *4*, 522-526.

Bashir, M., Rozaimee, A. B., & Wan Malini, B. W. (2015). A Word Stemming Algorithm for Hausa Language. *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. VI*, 25-31.

Bimba, A., Norisma, I., Norazlina, K., & Noor, N. F. (2015). *Stemming Hausa text: using affix-stripping rules.* Springer Science+Business Media Dordrecht.

Dawson, J. (1974). Suffix removal and word conflation. *Bulletin of the Association for Literary and Linguistic Computing, 2(3),*, 33-46. *https://www.herald.ng/full-list-hausa/*. (n.d.). Retrieved October 10, 2021, from The Herald: https://www.herald.ng/full-list-hausa/

Ishmailov, A. S., Mashita, A. J., Zailani, A., & Noor Hafizallah, A. R. (2016). A Comparative Study of Stemming Algorithms for use with the Uzbek Language. *ResearchGate*.

Lovins, J. (1968). Development of A Stemming Algorithm. *Development of a stemming algorithm. Mechanical Translation and Computational Vol. 11 (1 & 2)*, 21-31.

Muazzam Bashir, A. B. (2015). A Word Stemming Algorithm for Hausa Language. *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. VI (May – Jun. 2015)*, 25-31.

Newman, P. (2000). *The Hausa language: An encyclopedic reference grammar*. New Heaven: Yale University Press.

Porter, M. (1980). An Algorithm For Suffix Stripping. *Program, 14*, 130-137.

Rakesh, K., & Vibhakar, M. (2016). Applications of Stemming Algorithms in Information Retrieval - A Review. *International Journal of Advanced Research in Information System and Software Engineering*.