Int. j. inf. tecnol. (August 2022) 14(5):2355–2360
https://doi.org/10.1007/s41870-022-00942-9

ORIGINAL RESEARCH

A Trie based lemmatizer for Assamese language

Basab Nath[1] · Sunita Sarkar[1] · Surajeet Das[1] · Arindam Roy[2]

Received: 7 September 2021 / Accepted: 8 April 2022 / Published online: 31 May 2022

Abstract Lemmatization is a process which can be used to derive the headword or root word from its inflectional forms. So natural language processing applications can use lemmatization as a pre-processing step. A lemma is simply the 'Dictionary form of a word'. This paper aims to develop a tool for the purpose of lemmatization of words belonging to the Assamese language, using a Trie based approach. In this paper we have explored certain challenges related to lemmatization of Assamese language and also we have implemented a hybrid system containing multiple sections dedicated to elevate these challenges. Our implemented hybrid system operates on the basis of longest prefix match, character skipping algorithm and morphological analysers conforming to different rules in order to deal with the irregularities present in the Assamese language. Our proposed hybrid lemmatizer tool correctly lemmatizes 21,655 words out of 26,521 words, thus achieving an accuracy of 81.65%.

Keywords Trie · Lemma · Corpora · Indo Wordnet

✉ Sunita Sarkar
sarkarsunita2601@gmail.com

Basab Nath
basabnath@gmail.com

Surajeet Das
surajeet310@gmail.com

Arindam Roy
arindam_roy74@gmail.com

1 Department of Computer Science and Engineering, Assam University, Silchar, India

2 Department of Computer Science, Assam University, Silchar, India

1 Introduction

Language is an important tool for communication. The domain of natural language processing deals with the interaction between computers and human languages, how a computer processes natural language (human language) data, in particular. In the lexical information bases like lexicon, Word Net [1], and so on where a surface word is experienced in a raw text, its significance can't be derived from its entrances which are normally root words with their morphological and semantic depictions. So the significance of this surface word must be acquired by computing its suitable root word through lemmatization. In this manner, lemmatization is a fundamental requirement for any sort of semantic processing for dialects [2].

Lemmatization is not the same as stemming as in lemmatization it is not required to reduce the word to its stem form without bothering about POS, it actually produces the words to its normalized form. For example in stemming the word *studies* stemmed to *studi* whereas in case of lemmatization the word is translated into its root form that is *study,* which is the original form [2].

In this paper we have proposed a hybrid system comprising multiple sections like longest prefix match, character skipping algorithm and a morphological analyser conforming to different rules, dedicated to retrieving root words of different variants of inflectional words in Assamese. But where the input or the morphological variant has no similarity with their corresponding lemma, a dictionary is formed where the morphological variants and their corresponding lemma are stored in the form of<key, value>.

Assamese language is a state language mainly spoken by the people in the region of Assam and besides in Bangladesh and Bhutan otherwise called asamiya. There are about 15–20 million individuals who communicate in the language of Assamese. Most of the regions along the Brahmaputra valley communicate in Assamese. There are parcels of likenesses among Bengali and the Assamese language aside from a couple of contrasts. This can be demonstrated from the antiquated content Charya Padas which is even today being guaranteed both by the Old Assamese and Old Bengali [3]. In the seventh century Assamese which is also a part of Indo-Aryan script advanced having its underlying foundations from the Sanskrit language. In Indo Aryan Languages, Assamese, Bangla, Oriya and so forth are gotten from Modern Apabhransha. These dialects are initially derived from the Magadhi-Prakrit. Hema Saraswati who was one of the most famous and seasoned Assamese essayists composed the most well known ''Prahlada Charita'' towards the finish of the thirteenth century AD. Another conspicuous figure in Assamese writing was Madhava Kandali who had a place in the fourteenth century and he composed the popular epic Ramayana in the local language [4] (Figs. 1, 2, 3, 4, 5, 6).

2 Literature review

In 2006, Chrupala [5] introduced a way to deal with lemmatizing word structures. They introduced a basic information driven context delicate approach. To accomplish this undertaking he computed the Shortest edit script (SES) between turned information and output string. To get the output(lemma), all the changes made by SES must be applied to the input information. In 1968, Lovins [6] portrayed a model where he developed a list of 294 suffixes where each suffix is connected to 29 conditions in addition

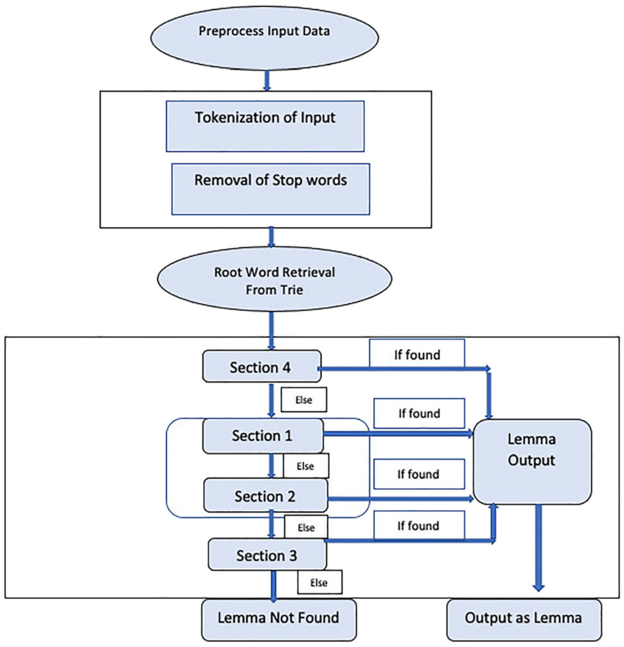


Fig. 1 Flowchart of the hybrid lemmatizer tool

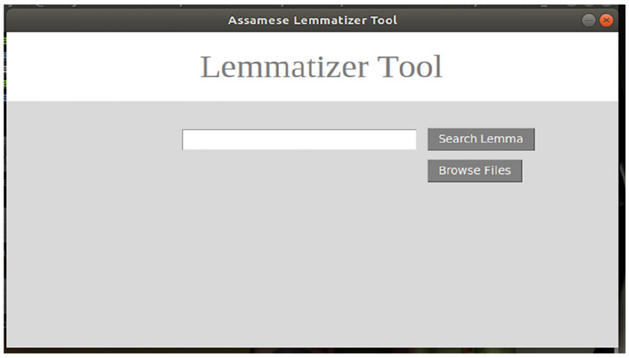
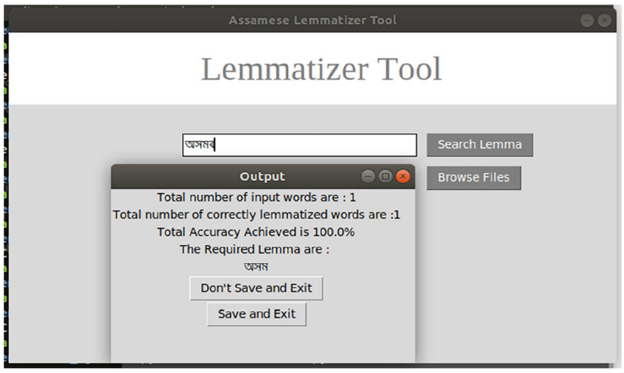
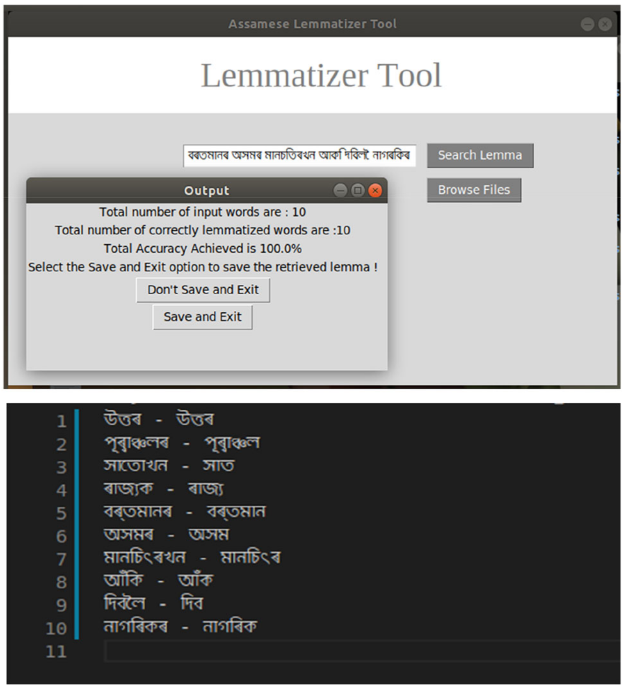


Fig. 2 Overview of GUI application of Lemmatizer



Total number of input words are : 1
Total number of correctly lemmatized words are :1
Total Accuracy Achieved is 100.0%
The Required Lemma are :
অসম

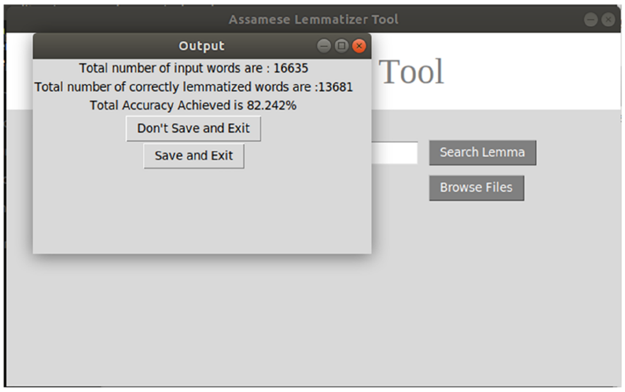
Fig. 3 Test 1, where a single word is given as input



Total number of input words are : 10
Total number of correctly lemmatized words are :10
Total Accuracy Achieved is 100.0%
Select the Save and Exit option to save the retrieved lemma !

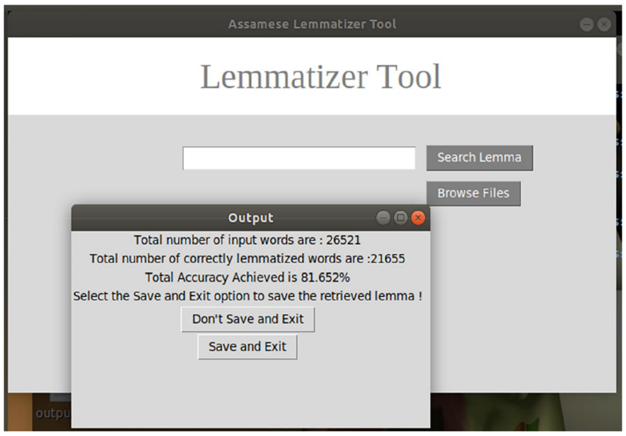
```
উত্তৰ - উত্তৰ
পূৰ্ব্বাঞ্চলৰ - পূৰ্ব্বাঞ্চল
সাতোখন - সাত
ৰাজ্যক - ৰাজ্য
বৰ্তমানৰ - বৰ্তমান
অসমৰ - অসম
মানচিত্ৰখন - মানচিত্ৰ
আঁকি - আঁক
দিবলৈ - দিব
নাগৰিকৰ - নাগৰিক
```

Fig. 4 Test 2, where a sentence is given as input



Total number of input words are : 16635
Total number of correctly lemmatized words are :13681
Total Accuracy Achieved is 82.242%

Fig. 5 Test 3, where a paragraph of 16,635 words is provided as input



Total number of input words are : 26521
Total number of correctly lemmatized words are :21655
Total Accuracy Achieved is 81.652%
Select the Save and Exit option to save the retrieved lemma !

Fig. 6 Test 4, where a paragraph of 26,521 words is provided as input

to 35 rules of transformation. This paper was considered to be the first stemming model which was grown explicitly for Natural language applications. A suffix with exact condition for an input word is checked and removed. For lemmatization in 2008, Plisson et al. [7] proposed a rule based methodology which was back then most acknowledged. This method depends on word endings. To get the ultimate lemmatized word structure, their method evacuated or added suffixes. Their paper explained why the Ripple Down Rules (RDR) approach is ideally suited to the task of lemmatizing words from Slovene free text. In order to find out the lemma from a word in 2009, Jongejan et al. [8] proposed a strategy which naturally creates lemmatization rules. This was trained on English, Icelandic, Norwegian, Polish, Slovene and Swedish, German, Greek, Danish, Dutch, language pairs. It is observed that Dutch and German affix algorithms tend to use suffix-only rules when dealing with OOV terms. They have also found that lemmatized words perform better than the average. Other terms, such as infix and suffix, frequently necessitate morphological alterations in more than one position. In 2003, Ramanathan and Rao [9] in his work for building a Hindi stemmer, he used a truly physical suffix list. He likewise performed the longest match stripping. This paper also included an in-depth discussion of the systematic approach that was used to develop the suffix list, as well as the linguistic reasoning that was used to include certain suffixes in the final result. This work is considered as the most punctual work for Indian dialects. In 2012, Dabre et al. [10] they detailed the development of a morphological analyzer for the Marathi language, and also stated that it may be applied to other languages that use suffix stacking as well. Morphotactics is meticulously documented, with all generalizations and exceptions taken into consideration, after which a typical FSM-type tool is used to perform the necessary analysis. In another methodology Paul et al. [11] proposed a Hindi lemmatizer. To get a legitimate root structure suffixes are stripped by different standards and essential expansion of character(s) is done. In 2013, Faridee et al.[12] proposed a bengali lemmatizer technique. A GRALE is a lemmatizer based on Graph used to create the lemma for Bengali language which consists of two phases. In one phase successive suffixes are separated automatically and in the subsequent advance, case suffixes are recognized by a human. In 2012 El-Shishtawy et al. [13] used unmistakable Arabic language data resources and proposed Arabic lemmatizer count. He makes exact lemma structures which help IR purposes and a most extraordinary precision of 94.8% is represented. In 2018, Chaudhary et al. [14] proposed a novel approach for constructing Hindi lemmatizers in Devanagari script. Inflected words are divided into smaller components and then used to generate root words. Their hybrid approach also showed that a system can handle inflected words with unknown suffixes. In 2018, On the basis of a machine-learning-based approach and a lemmatization dictionary, Freihat et al.[15] in their paper introduced a new lemmatizer tool for Arabic language, with excellent accuracy in comparison to the former. Their outcomes show that the overall performance of the lemmatization pipeline is greater than 98 percent. In 2019, Kanereva et al.[16] described sequence-to-sequence lemmatization with morphosyntactic as a new way of informing the model about a word's context. It was shown that their method outperforms all the baseline systems in terms of relative error reduction with 52 different languages across 76 treebanks, compared to the best overall baseline system, on average by 19 percent. In 2018, Preeti Dubey [18] in her paper compared and analyzed the similarities and differences in grammatical and inflectional structures between Hindi and Dogri. Observations have proved useful in determining the best method for this system, setting the rules for inflectional analysis, discovering collocations, ambiguity of words, and so on. In 2021, Arjun et al.[1] have developed an Assamese stemmer, which provides solutions to numerous shortcomings as well

Table 1 Result Comparison Table

Model	Input word count	Correctly Lemmatized word count	Accuracy	Key Difference
System from published work	26, 521	19, 258	72.61%	Cannot lemmatize compound words
Our Proposed System	26, 521	21, 655	81.65%	Compound words can be lemmatized

as efficient utilization of various attributes as vocalization ambiguity, single solution etc.They used 20,000 words from 16 distinct publications, manually collecting all conceivable suffixes in the Assamese language with the help of an Assamese linguistic specialist.

3 Challenges in lemmatization for Assamese language

There were a number of challenges we faced in order to come up with a method to retrieve lemma of some variants of Assamese words. The lemma of such words cannot be retrieved from Trie using longest prefix match search. Longest prefix match search can be used to retrieve lemma of those words which contain the lemma as a prefix itself. Examples of such words are নাগৰিক□ whose lemma is নাগৰিক, পূৰ্বাঞ্চলৰ whose lemma is পূৰ্বাঞ্চল, অসম□ whose lemma is অসম. But, the lemma of words such as সামীপ্য়(lemma=সমীপ), গাম্ভীযৰ্য়(গম্ভীৰ), এটীয়া(এটা), পনীয়া(পানী), ভদীয়া(ভাদ), কপহুৱা(কপাহ) etc. cannot be retrieved using longest prefix match search from Trie. Here the lemma of these words are not present as a prefix. The characters of the lemma are jumbled in the inflected word. Also, there is another variant of inflected words who have no resemblance with their corresponding lemma. Some of these words are গৈছিলো, গলোহতেনে, গলাহতেনে, গৈছে, গল, গৈছিলি, and the lemma of all these words is যা. All of these inflected forms do not even have this character 'য' (character of lemma). These are the different variants of Assamese words whose lemma cannot be retrieved using the longest prefix match search, and hence, we used a 'character skipping' algorithm and integrated different morphological analysers in the system to deal with these irregular inflected words. The working methodologies are discussed in the Methodology section.

4 Methodology

The approach we used here is a hybrid one. There are multiple sections dedicated to finding lemma of different variants of inflected words in Assamese Language. We shall be going through the entire system, section wise in order to visualize the working of the tool properly. Well, our approach here, firstly includes forming a Trie data structure by the insertion of root words taken from Indo Wordnet. There are a total of 13,130 different root words that we used to form the Trie.

4.1 Insertion of root words in Trie

The Trie is formed by inserting root words from Indo Wordnet. The Trie checks, if the word to be inserted is already stored in Trie or not, as and when a new word comes along. If the word is already inserted in Trie, then this particular word will be skipped.

4.2 Retrieving Lemma from Trie

Well, as mentioned above our proposed system is comprised of multiple sections dedicated to different variants of inflected words. They are described below:

Section 1. For words like বৰ্তমানৰ, whose corresponding lemma বৰ্তমান can be retrieved from Trie by longest prefix match search, provided the root words বৰ্তমান is inserted in Trie. The longest prefix match method can be used to retrieve lemma from Trie for words falling into the same category such as মানচিত্ৰখন (মানচিত্ৰ), দবিলৈ(দবি), নাগৰিকিৰ(নাগৰিক), পূৰ্বাঞ্চলৰ (পূৰ্বাঞ্চল), অসমৰ(অসম), ৰাজ্যক(ৰাজ্য), জলবায়ুৰ (জলবায়ু), মহাসাগৰৰ(মহাসাগৰ) etc.

Section 2. This section deals with words like এটীয়া whose root word is এটা. Here the letters of the lemma remain jumbled in the inflected word unlike in Sect. 1. So, in order to retrieve the lemma of such words, we have to use a tweaked version of the algorithm used in Sect. 1. Here, we skip the characters in the input word which are not present in Trie and continue to iterate over the input word until all the letters are processed. In this case, in order to find এটা from এটীয়া, we need to skip ী' and 'য়', since they are not stored in Trie and the obtained word is এটা which is its lemma and is stored in Trie. The rule used in this section works pretty much well for words having similar analogy to এটীয়া such as সামীপ্য়(সমীপ), গাম্ভীযৰ্য়(গম্ভীৰ) etc. We have referred to this algorithm as the 'character skipping' algorithm and we have integrated this in the lemmatizer tool. Nonetheless, in the same cases there arises an issue with retrieving the lemma using the algorithm in this section. For example, assuming a scenario where there are two root words stored in the Trie such as সমীপ and সাগৰ. We are trying to get the lemma of সামীপ্য়. With the help of the algorithm mentioned in this section, we cannot get সমীপ (the lemma of সামীপ্য়) by searching from Trie, reason being the second letter 'া' in সামীপ্য় needs to be skipped in order to get সমীপ. But due to the presence of the word সাগৰ in Trie, the algorithm will not be able to skip 'া' as it is present in this word. To elaborate the problem in this case, here 'স' is the first letter in the word and it is present in the Trie. When we move to the next node, the current letter is 'া' which is also present in Trie(due toসাগৰ). But the next letter 'ম' inসামীপ্য় will not be present in the next node as there are no words in Trie with the prefix 'সাম'. Hence, we will not be able to get the desired lemma.

We mitigated this problem by adding a morphological analyser in the system. The rule to elevate this particular problem is, by truncating the second letter of the word ('া' in this case). The resultant word in this case would be of the form 'সমীপ্য় ' from which its lemma ' সমীপ ' can be easily retrieved by the longest prefix match algorithm as mentioned in Sect. 1.

Section 3. This section involves adding another rule to the morphological analyser, where it deals with the words such as কাৰুণ্য়(কৰুণা), পনীয়া(পানী), ভদীয়া(ভাদ), কপহুৱা(কপাহ) etc. Here, these words are derivational suffixes whose lemma cannot be directly retrieved from Trie. So, the morphological analyser inculcates a couple of grammatical rules as per the language, so as to split the derivational inflected word into its root form.

Section 4. This section is dedicated to retrieving lemma of inflected words such as গৈছিলো, গলোহতেনে and গৈছে and their lemma is যা. As we can observe here that the inflected words and their corresponding lemma have no similarities at all or minimal similarities. So in order to deal with these types of words, we formed a Dictionary which is of the form<key, value>. Here, the key is the inflected word and the corresponding value is the lemma. Words such as গৈছিলো (যা),গলোহতেনে (যা), গলাহতেনে (যা), গৈছে (যা), গল (যা), গৈছিলি (যা), ৰুৱা (ৰো),ৰুইছিলি (ৰো), ৰুইছ (ৰো),ৰুইছে (ৰো),ৰুব (ৰো),ৰুইছা (ৰো) and many more belong to this category.

5 Results and Analysis

As rightly mentioned above, we inserted 13,130 unique Assamese words(taken from Indo Wordnet) in the Trie. For testing our lemmatizer tool, we took the help of an Assamese monolingual corpora, available on TDIL(Technology Development of Indian Languages). The Assamese text data in the corpora belongs to a variety of domains such as, agriculture, art and culture, economy, history, health, entertainment, tourism, sports and many more. We performed a variety of experiments whose snapshots are shown below:

The accuracy metric as we see here is computed using the below formula:

$$Accuracy = \frac{Number\ of\ correctly\ lemmatized\ words}{Total\ number\ of\ words\ provided\ as\ input} * 100\% \quad (1)$$

6 Comparison with existing work

Since there remain a very few published works in Lemmatization for Assamese language, we have taken a Lemmatizer system from an already published work [17], and implemented it using our corpus. This lemmatizer taken from [17], is also based on Trie data structure. The novel aspect of our work in comparison to previous work is that our proposed lemmatizer tool not only uses the standard search technique which is the longest prefix match, but also uses our proposed character skipping algorithm and a morphological analyser consisting of rules, in order to tackle irregular words in the language. A performance comparison of both this system and our proposed system are tabulated below:

Hence, as it can be seen from the experimentation results, that our proposed system outperforms the existing published work by a considerable margin, for the corpus belonging to the Assamese language.

7 Conclusion and future work

We implemented a Lemmatizer tool using Trie data structure, integrated a character skipping algorithm and various other rule based techniques to lemmatize irregular forms of words in the Assamese language, which cannot be simply split into their root form using the conventional search techniques of Trie. But there remain other variants of Trie such as Burst Trie which offers a better space complexity than a regular Trie (the one we used in our experiments). As a future work, other variants of Trie can be explored in order to improve the Lemmatizer model. The Lemmatizer tool fails to retrieve lemma of the inflected words which are not present in Trie.

References

1. Miller GA, Beckwith R, Fellbaum C, Gross D, Miller K (1993) Introduction to word- net: an on-line lexical database. Princeton University, Cognitive Science Labora- tory, Technical report
2. Balakrishnan V, Ethe LY (2014) Stemming and lemmatization: a comparison of retrieval performances. Lect Notes Softw Eng 2 (3):262–267. https://doi.org/10.7763/lnse.2014.v2.134
3. Kakati B (1941) Assamese-its formation and development. Govt of Assam, Depart- ment of Historical and Antiquarian Studies, Narayani Handiqui Historical Institute, Guwahati, Assam
4. Sarma SK, Medhi R, Gogoi M, Saikia U (2010) Foundation and Structure of Devel- oping an Assamese Wordnet
5. Chrupala G (2006) Simple Data-Driven Context-Sensitive Lemmatization. National Centre for Language Technology Dublin City University Glasnevin, Dublin 9, Ire- land. (2006)
6. Lovins JB (1968) Development of a stemming algorithm. Mech Transl Comput Linguist 11(1–2):22–31
7. Plisson J, Lavrac N, Mladenic D (2008) A Rule based Approach to Word Lemmatiza- tion. In: Pro- ceedings of 7th International Multiconference In- formation Society, IS 2004.Institute Jozef Stefen, Ljubljana. 83–86
8. Jongejan B, & Dalianis H (2009) Automatic training of lemmatization rules that han- dle morphological changes in pre-, in- and suffixes alike.In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International JointConference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJC- NLP 09. https://doi.org/10.3115/1687878.1687900
9. Ramanathan A, Rao DD (2003) A lightweight stemmer for Hindi. In: Workshop on Computational Linguistics for South-AsianLanguages, EACL (2003)
10. Dabre R, Amberka A, Bhattacharyya P (2012) Morphological analyzer for affix stacking languages: a case study of marathi. indian institute of technology bom- bay, In: Proceedings of COLING: PostersMumbai-400076, Indi, 225–234
11. Paul S, Tandon M, Joshi N, Mathur I (2013) Design of a rule based hindi lemma- Tizer. Comput Sci Inform Technol (CS & IT). https://doi.org/10.5121/csit.2013.340
12. Faridee AZM, Tyers FM (2009) Development of a morphological analyser for ben- gali. In: Workshop on Free/Open- Source Rule-Based Machine Translation, Universi- dad de Alicante. Departamento de Lenguajes y Sistemas Informáticos, 43–50
13. El-Shishtawy T, El-Ghannam F (2012) An accurate Arabic root-based lemmatizer for information retrieval purposes. IJCSI Int. J. Comput. Sci. Issues 9(1, 3)
14. Chaudhary S, Chakma K (2018) A hybrid approach for lemmatizer for Hindi language. Int J Comput Appl 13:39–44
15. Freihat AA, Abbas M, Bella G, Giunchiglia F (2018) Towards an optimal solution to lemmatization in Arabic. Proc Comput Sci 142:132–140. https://doi.org/10.1016/j.procs.2018.10.468
16. Kanerav J, Ginter F, Salakoski T (2019) universal lemmatizer: a sequence to sequence model for lemmatizing universal dependencies treebanks. Nat Lang Eng 27(5):545–574
17. Chakrabarty A, Choudhury SR, Garain U (2014) IndiLem@-FIRE-MET-2014: An Unsupervised Lemmatizer for IndianLanguages, Indian Statistical Institute,Kolkata
18. Dubey P ((2018) The Hindi to dogri machine translation system: Grammatical perspective. Int J Inform Technol 11(1): 171–182. https://doi.org/10.1007/s41870-018-0085-4
19. Gogoi A, Baruah N, Sarma SK, Phukan RD (2021) Improving stemming for Assamese information retrieval. Int J Inf Technol 13(5):1763–1768. https://doi.org/10.1007/s41870-021-00718-7