IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.6, Issue.10, Oct. 2018 E-ISSN: 2347-2693

Pre-processing Phase of Automatic Text Summarization for the Assamese Language

Gunadeep Chetia[1*], Gopal Chandra Hazarika[2]

[1]Centre for Computer Science and Applications, Dibrugarh University, Dibrugarh, India
[2]Dept. of Mathematics, Dibrugarh University, Dibrugarh, India

[*]Corresponding Author: gunadeep@dibru.ac.in, Tel.: +91-99544-53339

Available online at: www.ijcseonline.org

Accepted: 16/Oct/2018, Published: 30/Oct/2018

***Abstract*—** Pre-processing is the first and important phase of automatic text summarization. Pre-processing helps in normalizing a text document and generating a structured representation of the text. Major pre-processing tasks include segmentation, tokenization, stop-word removal, stemming and lemmatization. In this paper, we discuss these pre-processing tasks required for automatically summarizing Assamese text documents. Both Stemming and lemmatization play an important role in the pre-processing phase of morphologically rich highly inflected language like Assamese. We present a corpus based approach for stemming the Assamese words using n-gram similarity matching technique. We also propose a hybrid method for lemmatization of the Assamese verbs to obtain the grammatically correct root of a verb. Assamese verbs are the most inflectional compared to other word categories. Stemming alone is not sufficient to find the original roots in case of Assamese verbs. So, after segmentation, tokenization and stop-word removal we first apply stemming to all the words in the text document irrespective of their grammatical categories and then apply lemmatization to only the Assamese verbs. For identifying the Assamese verbs we use a look-up dictionary which contains a list of possible stems along with the corresponding lemma of the verbs.

***Keywords*—**Pre-processing, Summarization, Stemming, Lemmatization, n-gram

I. INTRODUCTION

Automatic text summarization is the process of shortening a long text document with software, considering and retaining the most important points of the original text [1]. With the rapid increase in the amount of available information on the Web every day, there is a growing need to produce focused summaries containing important points. Summaries save time of the reader and help in deciding whether to read a full document or not. They are also useful in indexing and question answering system. Pre-processing generates structured representation of text which is an important phase of automatic text summarization systems [2]. Pre-processing involves various steps in preparing the text for processing. It transforms the text into an object with minimal linguistic features. Pre-processing has mainly two objectives: word normalization and lexicon reduction in a text. The pre-processing phase of automatic text summarization mainly includes Text segmentation, Tokenization, Stop-word Elimination, Stemming, and Lemmatization. Pre-processing may optionally include annotation through grammatical tagging or Parts-of-Speech tagging, Named Entity Recognition (NER), Extraction of terms and keywords etc.

In this paper, we describe the major pre-processing tasks applied to the automatic text summarization of the Assamese language. Assamese is an East Indian Language of Indo-European origin, spoken by more than 15 million people in India [3]. Being a morphologically rich and highly inflected language, the pre-processing stage of text summarization in Assamese requires a great deal of effort which is different from English and other resource-rich languages. Although many works related to automatic text summarization have been carried out for English and some other languages, little work has been reported for the Assamese language [4,5]. However, with the recent growth of Assamese content on the web and digital media, it has become important to develop automatic summarization system for the Assamese language.

The rest of the paper is organized as follows: section II presents a brief overview of all the pre-processing steps carried out for automatic summarization of Assamese text, section III describes a corpus-based approach for stemming using n-gram similarity matching technique, section IV contains a methodology for lemmatization of the Assamese verbs, section V presents the results and discussion and in section VI we draw the conclusion with future directions.

II. PREPROCESSING STEPS

A. Text Segmentation

Text segmentation is an essential pre-processing step for extractive summarization. Segmentation divides an input text into separate textual contexts. Ideally, a context should express a relatively independent and standalone piece of information. In extraction-based summarization, sentences may be considered as contexts. Sentences may be identified by detecting sentence boundaries which involves detection of punctuations and quotations. Sometimes punctuation ambiguities make sentence boundary detection rather hard. In Assamese, the punctuation symbols that mark the end of a sentence are “।”(*daar*i or Devanagari *danda*, U+0964), “?”(U+003F) and “!”(U+0021). We have applied a rule-based approach which uses a database of regular expressions denoting segments that contain punctuation marks, but does not indicate sentence boundaries like factorial notation (!), symbols inside single or double quotes etc. [6].

B. Tokenization

Tokenization involves splitting the sentences into words. In Assamese, words in a sentence are separated by spaces and punctuation marks. So, using a regular expression the words can be easily identified from the sentences.

C. Stop-word Removal

Stop-words are the words occuring very frequently in a language that don’t carry any particular information on their own. These may be conjunctions, exclamations, particles etc. These words are removed in the pre-processing phase of automatic text summarization so that we can focus on the important words. Stop-words may be grouped into a Stop-list which may be domain dependent or independent. Since we could not find a ready list of stop-words for Assamese, we have manually prepared a list of 300 domain-independent Assamese stop-words after going through various Assamese text documents. After analyzing 1000 Assamese articles of various domains, we have found that the words in our Stop-list account for between 20% and 30% of the total words contained in the articles.

D. Stemming

Stemming is the process of extracting the base form of an inflected word. The goal of the stemming is to reduce inflectional forms of a word to a common base form known as the “stem” . The stem may not be a morphologically correct root of the word [7]. It is usually sufficient if the stemming produces the same stem for related words. For example: {calculation, calculator, calculated, calculating} all these four related words should produce the same stem ‘calculat’ after stemming. Similarly in Assamese, the words {শিক্ষকে, শিক্ষকলৈ, শিক্ষকসকলক, শিক্ষকজনক} should produce the stem ‘শিক্ষক’.

Assamese is a morphologically rich highly inflected language [8,9]. We have observed that, in Assamese, verb inflection is the most predominant. An Assamese verb can take up to 5000 inflected forms according to person, tense, aspect and modality. For example we have found that the root verb ‘কৰ’ (to do) can take up to 4424 forms by combining single affix or more than one affix in a valid sequence. Some verbs like ‘পাৰ’(to be able), ‘যা’(to go) takes suppletive forms which cannot be deduced by simple rules from the base forms. For example the verb ‘পাৰ’ becomes ‘নোৱাৰ’ in negation and the verb ‘যা’ becomes ‘গৈছিল’ in simple past tense.

Thus, it is obvious that Stemming alone is not sufficient in case of Assamese verbs to get the proper root form. After applying Segmentation, Tokenization and Stop-word removal in the input text, we first apply Stemming to all the words using a corpus-based n-gram similarity matching technique described in section III. Then we apply a technique called Lemmatization[10] to the verbs in the input text to find out the normalized form.

E. Lemmatization

Lemmatization involves finding the grammatically correct root form or dictionary form known as lemma from a word [11]. For instance, {begin, began, begun} should produce ‘begin’ after lemmatization. Similarly in Assamese {গ’ল, গৈছিল, যাম, যাব, গৈছিলোঁ, গ’লাগৈ} should produce the lemma ‘যা’.

Both stemming and lemmatization play a very crucial role in pre-processing stage of automatic text summarization. However, considering the complexity of the lemmatization many automatic text summarization systems deal with only stemming in the pre-processing step. Through our literature survey, we have found that Assamese verbs are the most inflected ones and sometimes the original root word completely gets lost in different forms. The inflected form can consists of prefix, suffix or a sequence of prefix and suffixes added to the original root form or its modified root form. Other lexical categories of Assamese, particularly nouns and pronouns also take many inflected forms, but most of them are inflected by adding suffixes to them and retaining the base word in the inflection.

Lemmatization is stricter than stemming. Lemmatization is difficult to implement as it is related to semantics and parts of speech (POS) of the word [12]. A hybrid approach for the lemmatization of the Assamese verbs have been developed which is described in section IV.

III. CORPUS BASED APRROACH FOR STEMMING

A. N-Gram Similarity Matching

For languages like English, French and Slovene, a suffix-removal method is found to be sufficient [13]. Unlike these languages Assamese is a highly inflected fusional language. We have observed that the rule-based suffix-removal approach for stemming that produces good results for English and similar languages do not show same performance in Assamese. A purely rule-based Stemmer is also difficult to build for Assamese, since there is a large number of word formation rules in Assamese with many exceptions. Being a very new language in the field of Natural Language Processing, Assamese does not have sufficient digital resources like root word dictionaries, valid suffix list which are required to stem words by Dictionary Lookup method or Suffix Stripping methods.

In our approach, we have used a method devised by Adamson and Boreham[14] to identify semantically related pairs of words and extend it to perform a corpus-based stemming for Assamese. In this method, each word is first broken down into some units known as n-grams, where n represents the number of adjacent characters in the unit. If n=2, the units are known as bigrams and if n=3, they are known as trigrams. We have done experiment with both bigrams and trigrams in Assamese and found that bigrams produce better results than trigrams.

In the bigram model, firstly a bigram string is generated for each word taking two adjacent characters, e.g.
Calculation => ca al lc cu ul la at ti io on
Calculator=> ca al lc cu ul la at to or

In this example, the word "calculation' is decomposed to 10 bigrams and the word "calculator" is decomposed to 9 bigrams. A pair-wise comparison of the bigrams will give the number of shared bigrams of both the words. With this information we can calculate the similarity measure of the two words using Dice Coefficient (DC) [15] which is defined as

$$DC = \frac{2z}{x + y} \qquad (1)$$

where z is the number of common bigrams between the two words, x is the number of bigrams in the first word and y is the number of bigrams in the second word.
Thus, we calculate Dice Coefficient for each pair of words in the corpus and a similarity matrix is obtained as shown in Table 1.

Table 1: Example of Similarity Matrix

	W1	***W2***	***W3***	***W4***
W1	***1***	***0.2***	***0.7***	***0.6***
W2	***0.2***	***1***	***0.4***	***0.3***
W3	***0.7***	***0.4***	***1***	***0.4***
W4	***0.6***	***0.5***	***0.3***	***1***

Here W1, W2... represents the words in the corpus.
We then programmatically map the corpus into a graph by considering each word as a node of the graph. Two nodes are connected if they have a Dice Coefficient value equal to or greater than a predefined threshold value. An example of such graph is shown in Figure 1. The words are then clustered by finding out the strongly connected components from the graph [16]. Thus, each cluster contains a list of similar words based on bigram similarity.

B. Data and Experiment

For stemming we have developed a corpus of Assamese words collected from dictionaries and through parsing Assamese text from different online sources like Assamese Wikipedia, Blogs and e-magazines. We first remove all the duplicate words and sort the corpus according to Assamese alphabetical order using a customized sorting program. Words starting with separate alphabets are stored in separate files for faster processing. We apply n-gram technique and generate bigrams for each word. Dice Coefficient is calculated for each pair of words in a file and similarity matrix is obtained. For Assamese, we experimented with different threshold values for Dice Coefficient and found that threshold value of 0.6 is suitable. We find out all the strongly connected words and group them to form clusters of related words.

When a word from an input text is to be stemmed, we make use of the clusters obtained through n-gram similarity matching. We first search the input word in the clusters and if it is found we consider that cluster where it exists. In that cluster, we extract the longest matching unit of all the words in the cluster by string comparison from left to right. This longest matching unit is taken as the stem of the word.

If the input word is not found in any of the existing clusters, then n-gram similarity matching is carried out again and the word is either accommodated in the appropriate cluster or the existing clusters are modified. Then stemming is performed as described above.

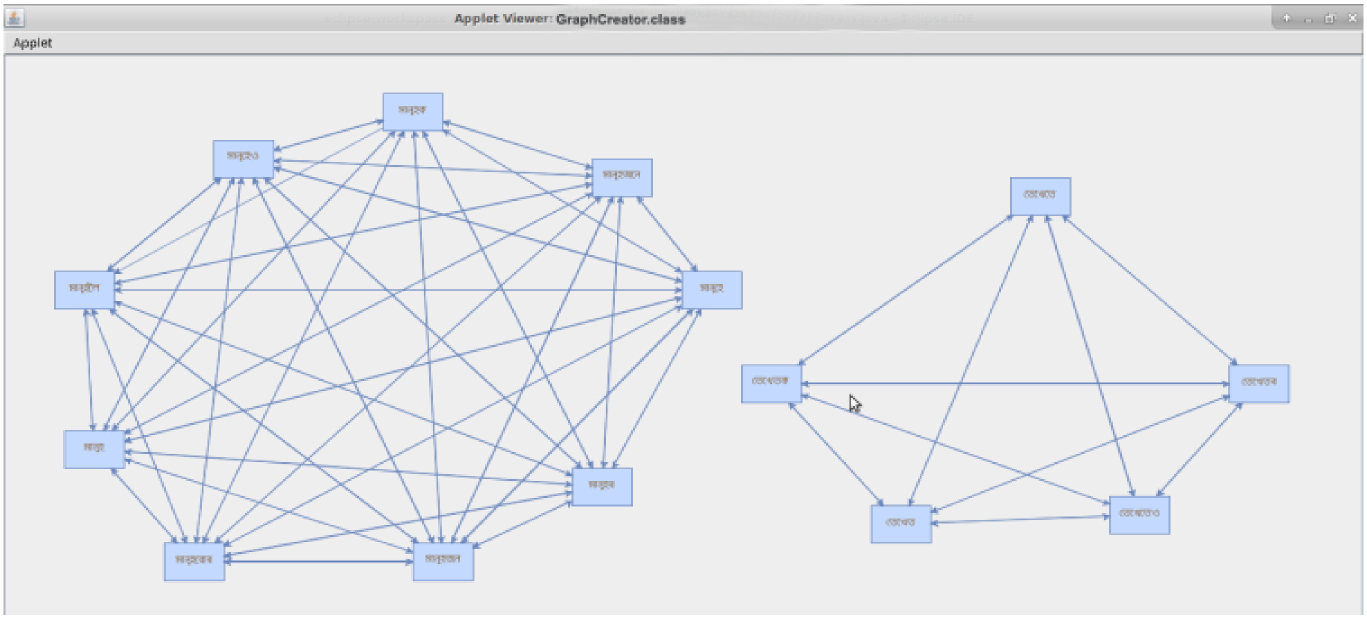

Figure 1: Example of graph conecting the words

IV. LEMMATIZATION OF VERBS

For lemmatization, we first prepare a lookup dictionary where we store all the possible stems of a verb along with its lemma. Table 2 shows the structure of the lookup dictionary with three Assamese verbs যা, ৰ and পাৰ.

Table 2: Lookup Dictionary for Three Assamese Verbs

Stem	**Lemma**
গৈ, গ'ল, যোৱা, নগৈ, নগ'ল, নোযোৱা, যা, নাযা, নেযা	যা
ৰৈ, ৰ'ল, ৰোৱা, নৰৈ, নৰ'ল, নোৰোৱা, ৰ	ৰ
নোৱাৰ, পাৰ	পাৰ

The stems that we obtain from the Stemming process described above are compared with those of the lookup dictionary. If a stem is found in the lookup dictionary, it is replaced with the corresponding lemma. Since we have stored probable stems of the verbs and not the other word categories in the lookup dictionary, this process will result lemmatization of only the verbs in Assamese.

V. RESULT AND DISCUSSION

By manually evaluating 10,000 common words stemmed through the discussed approach, we have found that 9147 words are stemmed accurately resulting Correctly Stemmed Words Factor (CSWF) equal to

CSWF=WSC/WS*100
=9147/10000*100=91.47

where ,
WSC: Number of words stemmed correctly
WS: Total number of words stemmed

This is quite a satisfactory result for a resource-poor language like Assamese. The result is also comparable with the stemmers of other Indic languages. The accuracy of our stemmer depends on the proper clustering of the equivalent classes of words. The words which are different in only one character but can have exactly the same suffixes are found to be incorrectly stemmed. For example, the Assamese words কল and কলি are two semantically different words having difference only in one character. They can also have same suffixes like বিলাক, বোৰ, কেইটা, টো etc. So the inflected words will form a single equivalent class like {কলবিলাক, কলিবিলাক, কলটো, কলিটো, কলকেইটা, কলিকেইটা, কলবোৰ, কলিবোৰ...} from which only one stem "কল" will be obtained, whereas the correct stem for {কলিবিলাক, কলিটো, কলিকেইটা,কলিবোৰ} is "কলি".

The accuracy of the lemmatization process for the Assamese verbs, in our approach, depends on the result of stemming. Since the lemmatization of the verbs is based on lookup table, it will produce accurate lemma provided stem is generated as expected. For testing, we have taken two verbs

“ধৰ”(to catch), “যা” (to go) and manually derived their possible forms according to tense, person, aspect and modality. We could find 2214 derived words from the verb ধৰ and 1645 derived words from the verb যা. First we have applied stemming to the derived words of ধৰ and found that all these derived words exactly get stemmed to root ধৰ which is also the correct lemma. The derived words of the verb যা gets stemmed to 10 different stems {গৈ, গ’ল, যোৱা, নগৈ, নগ’ল, নোযোৱা, যা, নাযা, নেযা}. Since all these stems were found in the lookup dictionary, their corresponding lemma “যা” could be correctly obtained. The performance of lemmatization depends upon how accurately we prepare the lookup dictionary.

VI. CONCLUSION AND FUTURE SCOPE

Amongst all the pre-processing tasks required for summarizing Assamese text using extractive techniques, stemming and lemmatization require a great deal of effort. Our corpus-based approach for stemming using n-gram similarity matching technique showed a good result with more than 90% accuracy. We expect that this accuracy can be improved further by refining the corpus and introducing some rule-based conditions while clustering the words. We have observed that, in our approach, the accuracy of lemmatization of the Assamese verbs depends upon the accuracy of the stemming process. We have found that for summarizing Assamese text, lemmatization is required as a pre-processing step only for the Assamese verbs as stemming alone is sufficient to obtain the proper roots of all the words except the verbs. Our corpus-based approach for stemming can be used not only for automatic text summarization but also for other language-processing tasks of highly inflected resource-poor languages. The limitation of our approach for stemming is that if two semantically different words can take same suffixes and they differ by only one character, then the proper stem may not be obtained. Though the number of such words is not large in Assamese, we believe that further study can be done to improve the performance of the stemmer in case of such words.

REFERENCES

[1] Maryam Kiabod, Mohammad Naderi Dehkordi and Sayed Mehran Sharafi, “*A Novel Method of Significant Words Identification in Text Summarization*”, Journal of Emerging Technologies in Web Intelligence, Vol. 4, No. 3, August, 2012.

[2] Joel Larocca Neto, Alex A. Freitas, Celso A. A. Kaestner, “*Automatic Text Summarization using a Machine Learning Approach*”, Proceeding SBIA '02 Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence Pages 205-215 November 11 - 14, 2002.

[3] Gordon, Raymond G., Jr. (ed.). “*Ethnologue: Languages of the World*”, Fifteenth edition. Dallas, Tex.: SIL International, 2005.

[4] Dipanjan Das, André FT Martins. "*A survey on automatic text summarization*." Literature Survey for the Language and Statistics II course at CMU 4,192-195, 2007.

[5] Prachi Shah, Nikita P. Desai, “*A Survey of Automatic Text Summarization Techniques for Indian and Foreign Languages*”, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) – 2016.

[6] Silla, C.N., Kaestner, C.A.A. “*An Analysis of Sentence Boundary Detection Systems for English and Portuguese Documents”* In: Gelbukh A. (eds) Computational Linguistics and Intelligent Text Processing. CICLing 2004. Lecture Notes in Computer Science, vol 2945. Springer, Berlin, Heidelberg.

[7] Moral, C., de Antonio, A., Imbert, R. & Ramírez, J. “*A survey of stemming algorithms in information retrieval/ Information Research*”, 19(1) paper 605.

[8] Banikanta Kakati. “*Assamese, Its Formation and Development*”. LBS Publication, G.N.B. Road, Guwahati, fifth edition, 1995.

[9] Golok Chandra Goswami, “Structures of Assamese”. Department of Publication, Gauhati University, 1982.

[10] Nitin Indurkhya , Fred J. Damerau, “*Handbook of Natural Language Processing”*, Chapman & Hall/CRC, 2010.

[11] Tuomo Korenius , Jorma Laurikkala , Kalervo Järvelin , Martti Juhola, “*Stemming and lemmatization in the clustering of finnish text documents*”, Proceedings of the thirteenth ACM international conference on Information and knowledge management, Washington, D.C., USA ,November 08-13, 2004.

[12] Plisson Joel, Lavrac Nada and Mladenic Dunja. “*A rule based approach to word lemmatization*”, Proceedings of the 7th International Multi-Conference Information Society IS. 2004.

[13] M. F. Porter “*An algorithm for suffix stripping. Program”,* 14(3): 130-137. 1980

[14] Adamson, G. W. & Boreham, J., "*The use of an Association Measure Based on Character Structure to identify Semantically Related Pairs of Words and Document Titles*", InformationStorage and Retrieval 10, pp 253-260, 1974.

[15] Akinwale, A.T., Niewiadomski, A E Cient “*Similarity Measures for Texts Matching*” Journal of Applied Computer Science Vol. 23 No. 1,pp. 7-28, 2015,

[16] Kleinberg, J. & Tardos, É. “*Algorithm Design*”, Addison Wesley, 2006.

Authors Profile

Gunadeep Chetia is currently pursuing Ph. D. in the Centre for Computer Science and Applications, Dibrugarh University, Dibrugarh, Assam, India. He has more than six years of teaching experience. He has also developed a number of tools related to language technology. His area of research is Natural Language Processing.

Gopal Chandra Hazarika is a Professor at Department of Mathematics, Dibrugarh University, Dibrugarh, Assam, India. He was the founder Director of Centre for Computer Science and Applications, Dibrugarh University. He has more than 30 years of teaching and research experience. He has published a number of research papers on Mathematics and Computer Science in reputed international journals.