IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

An automatic identification of function words in TDIL tagged Bengali corpus

Subrata Pan[1*], Diganta Saha[2]

[1]Department of Computer Science and Engineering, Jadavpur University, Kolkata, India
[2]Department of Computer Science and Engineering, Jadavpur University, Kolkata, India

[*]*Corresponding Author: neruda0101@yahoo.com, Tel.: +91-82405-08069*

Available online at: www.ijcseonline.org

***Abstract*—**Function words are quite high in textual information as compared to content words; where dimensionality is a critical challenge. Performance of text processing task deteriorates due to the presence of the function words in textual context. So, elimination of these words is an important activity in text processing to reduce the computational complexity and improve accuracy in the system. Many researches are performed for standard function words identification for English, Arabic, Chinese, Punjabi, Hindi, etc. In Bengali language processing, a limited number of standard function words are available. To address this limitation, we propose a computer based automatic system for identification of high scored function words from TDIL tagged Bengali corpus, Govt. of India. Total corpus consists of total 670,831 words and 134,884 distinct words. Our proposed system identifies 8 set of function words i.e. total 33,985 function words are identified in Literature domain of monolingual tagged corpus. At the end of our experiment, we achieved 290 standard function words as per their computed rank.

***Keywords*—**Bengali Text Processing, Function Words, Bag of words, NLP

I. INTRODUCTION

Pre-processing is a significant phase in the text processing. It renovates the unprocessed textual information in an effective format for text processing task. This pre-processed format is suitable for employing different types of text processing methods [1]. A chunk of unprocessed textual information is constructed by the use of content and function words. A word can be considered as function word and content word. Function words are words that have small lexical sense or have ambiguous sense and express grammatical relationships among other words within a sentence. In any textual context noun, verb, adjective and adverb words are considered as content words and function words are prepositions, pronouns, determiners, conjunctions, auxiliary verbs, etc. Sense of a Bengali word like “কাছে” [adjective] can be used in various forms and signifies their own sense like “কাছে কাছে” [adverbs], “কাছে পিঠে” [verbs], “সমীপে” [preposition], “আমার কাছে” [pronoun], etc. Function words in textual context express a grammatical relationship with content words in sentences. Function words are also known as grammatical words or grammatical morphemes. Stopwords are subset of function words. In computing, function words are filtered out during processing of any natural language text/data. In text processing task, function words (inflected/not) carries a vigorous part for construction of sentences rather it doesn’t carry an important significance. Function words in textual context explain or create a grammatical or structural relationship into which the content words may fit. But, the presence of function words in any form deteriorates the performance of text processing. Many researches were performed for identification of function words for English, Yoruba, Chinese, Arabic, Punjabi, Hindi languages. A reasonable amount of standard function words are available for these languages. In Bengali language processing, an inadequate number of standard function words are available for text processing. Also, function words are chosen manually for Bengali text processing.

To overcome the availability and identification strategy of Bengali function words, we propose a system for automatic identification of function words (AFWI) over TDIL tagged Bengali corpus [2], MeitY - Govt. of India. TDIL corpus consists of 49 dataset including 7 monolingual and 2 parallel tagged domains. Usages of our proposed system are in the field of Word Sense Disambiguation, Search Engine, Information Retrieval System, Multiword Expressions, Bengali Text Processing, etc.

This paper consists of 5 sections. Following the introduction, related works are briefly reviewed in Section 2. Proposed methodology is discussed in Section 3. Section 4 includes experimental result and the discussion. Conclusion and future scope is presented in Section 5.

II. RELATED WORK

Function words express a structural association with the content words in textual contexts. Many researches are performed for identification of standard function words. H. Saif, et.al. [3] are performed function word identification method using term frequency, inverse document frequency, singleton words, mutual information and term based random sampling approaches for Twitter Sentiment classification. They achieved effective results using several off-the shelf function word elimination methods for polarity classification of tweets data. While [4] are proposed some methods for identification and evaluation of function word lists from Terrier retrieval platform using four TREC collections. They presented term based random sampling approach for identification of function word. In the following of identification, they accomplished an appraisal between their proposed approach and four baseline approaches inspired by Zipf's law to achieved lower computational overhead. W.J. Wilbur [5] proposed methods for identification of function words from MEDLINE subset in Biotechnology area. Statistical testing with vector retrieval methodologies like cosine coefficient and document-document similarity is used to identify 10% function words.

Automatic extraction of domain specific function words from a large labelled corpus is proposed by M. Makrehchi, et.al. [6]. Experiment is performed through backward filter level performance and sparsity measurement of training data. Evaluation is done by using seven feature ranking measures: F-Measure, IG, Max (χ2), Mean (χ2), Max (OR), IDF, Random and validated by 5-fold cross validation scheme. However, entropy based algorithm is proposed to identify function words for Yoruba Language text [7]. Two set of corpuses of total 756,039 Yoruba words were used in diacritized and undiacritized versions to achieved 256 function words from diacritized texts and 189 function words from undiacritized text. The full text is reduced by 65.91% and 67.46% after elimination of function words from diacritized and undiacritized texts. Another function word construction method is proposed in identification of high frequency Persian words [8]. Using aggregate based methodology, authors identified a list of terms having low inverse document frequency and afterwards calculate entropy measurement for each word to build a list of high entropy valued words. Also, another automatic aggregated methodology is performed for identification of function words in Chinese language [9]. Function words are constructed through word segmentation, statistical model, information model and aggregation. Statistical analysis performed on a corpus of 423 English articles in TIME magazine and it contains 245,412 occurrences of words. Experiment was performed on TREC 5 and TREC 6 Chinese corpora which contain news reports of Xinhua and People's daily newspaper. H. Lili and H. Lizhu [10] are proposed an automatic approach for extraction of function word in text classification based on weighted Chi-squared statistics on 2∗p contingency table. They performed experiment on the Chinese corpus of Mayor's public access line project text consists of 87,540 labelled training samples and 10,925 labelled training samples. A semantic approach is proposed in Automatic Stopword Generation using Contextual Semantics for Sentiment Analysis of Twitter [11] to identify and remove function words automatically from twitter data. They performed experiments on 6 twitter dataset OMD, HCR, STS-Gold, SemEval, WAP and GASP. After identification and elimination of functional words, 0.42% accuracy and 0.94% F-measure values are computed. For evaluation of result, they performed binary sentiment classification using MaxEnt classifier and observed fluctuations after elimination of function words. To identify automatic function word list for Persian information retrieval systems using similarity function and part-of-speech information [12]. Experiment is performed to identify function word through determination of most efficient set of part-of-speech tagging by using PSWG algorithm. PSWG algorithm was performed in the basis of mean probability, variance probability, stability of distribution and entropy based measurement. PSWG identified a set of noun, adjective, unknown, number and verb tagged words as functional words. Also, an automated method is proposed for Punjabi Language texts [13]. Authors performed their experiments using 10,000 news articles from Punjabi newspaper to identify a list of function words on the basis of frequency count, mean probability, variance probability and decision variables methods.

From the above review, it is observed that in Bengali language text processing lacks the availability of standard function words. To fill-up the gap, we focus our research in this paper towards identification of function words in automated way. Following the identification, ranking of these words are computed for identification of standard function words. Our proposed methodology is described in consequent sections.

III. METHODOLOGY

In our paper, we used the following baseline methods for function words identification:

1. Term Frequency (TF)

Based on George Kingsley Zipf′s Law, TF is identified as the number of times a certain term (W) appears throughout a specific dataset (D). Every document is different in length; it is possible that a word will appear much more times in larger documents with compared to smaller ones. So, term frequency (TF) is calculated as:

$$\mathrm{TF} = \frac{\text{Number of times W appears in D}}{\text{Total number of W in D}} \tag{1}$$

2. Inverse Document Frequency (IDF)

IDF visualizes the importance of a term (W). Inverse document frequency is calculated as:

$$\mathrm{IDF}_W = \log_e \frac{\mathrm{NDOC}}{D_W} \tag{2}$$

Where, NDOC is the total number of datasets in the corpus and D_W is the number of documents containing term W.

3. Kullback-Leibler (KL) Divergence Measure

Suppose token_c is denoted as total number of tokens in the full corpus, F is denoted as word frequency of the query term in the corpus, l_x is denoted as sum of the length of the document set and wf_x is denoted as frequency of the query term in the document set. To determine importance of each term in the dataset, KL divergence measure [14] is expressed as:

$$I(W) = P_x \cdot \log_2 \frac{P_x}{P_c} \tag{3}$$

Where,

$$P_x = \frac{wf_x}{l_x} \text{ and } P_c = \frac{F}{\mathrm{token}_c}$$

4. Mutual Information

The mutual information method [14] is a type of supervised method to compute the mutual information between a given word and a document class. Mutual information between two random variables: term (W) and document class (C) is calculated as:

$$I(W;C) = \sum_{w \in W} \sum_{c \in C} P(w,c) \cdot \log \frac{P(w,c)}{P(w) \cdot P(c)} \tag{4}$$

Where, w and c is denoted as term and document class. I (W; C) is denoted as mutual information between term and document class. w = (0, 1) is the set in which a term w occurred (W = 1), or not occurred (W = 0) in a given document D. c = (0, 1) is the class set in which the D belongs to class c(c = 1), or not belongs to class c(c = 0).

5. Information Entropy

Claude Shannon postulates that entropy is a measure of randomness [15]. High random behavioral terms or low entropy valued terms are highly informative. Function words carry small information; they are high entropy valued words (terms). Consider W(w) denotes entropy of a given term w of a given set of n documents, $f_i(w)$ denotes frequency of term w in document i and n denotes number of documents. Entropy is calculated as:

$$W(w_j) = \sum P_{i,j} \cdot \log \frac{1}{P_{i,j}} \tag{5}$$

Where,

$$P_i(w) = \frac{f_i(w)}{\sum_{i=1}^{n} f_i(w)}$$

6. Mean of Probability

Suppose M is denoted as distinct terms, N is denoted as total documents, each term is denoted as a set W_j = (W_1,W_2, .., W_M) and each document is denoted as a set D_i = (D1, D2,..,D_N). For each term W_j, frequency of term in document D_i is denoted as $F_{i,j}$ [13] . Due to variable length of documents, length of document is to be normalized. Afterwards, calculation of probability $P_{i,j}$ for the term W_j in document D_i is expressed as:

$$P_{i,j} = \frac{F_{i,j}}{\sum_1^M W_j} \tag{6}$$

For each term W_j, the mean of probability (MP) among various documents is calculated as:

$$\mathrm{MP}(W_i) = \frac{\sum_{1 \le i \le N} P_{i,j}}{N} \tag{7}$$

7. Variance of Probability

The stability of distribution of term is measured by using variance of probability (VP) as:

$$\mathrm{VP}(W_j) = \frac{\sum_{1 \le j \le n} \left(P_{i,j} - \overline{P_{I,J}}\right)^2}{N} \tag{8}$$

8. Decision Variable

Decision variable (D_i) is measured as aggregation of mean and variance probability. Decision variable (D_i) for terms W_i can be expressed as:

$$D_i = \frac{\mathrm{MP}(W_i)}{\mathrm{VP}(W_i)} \tag{9}$$

In our proposed method, identification of function words are implemented through program controlled boundary pivots to generate 8 lists of function words. Afterwards, a unique, optimized and high-scored function word list is generated. Our proposed methodology is performed through the following seven sections: Text cleaning of part-of-speech (POS) tagged Bengali text, Tokenization of words, unique word identification, Bag of words (BOW) generation, AFWI algorithm, Optimization of words and Word score generation. Process diagram of our proposed method is as follows:

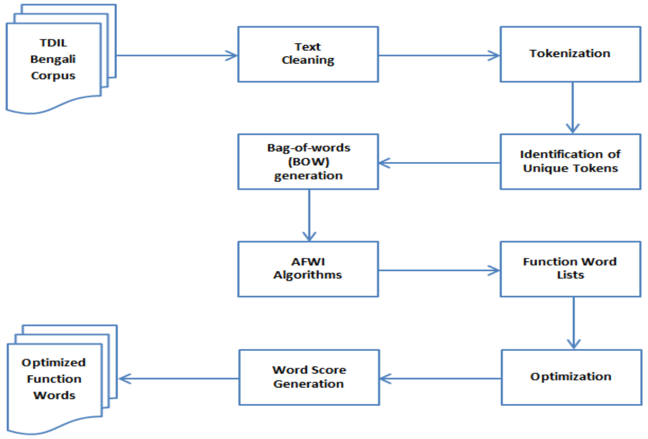

Figure 1: Proposed model for automatic construction of function words

Consider a dataset (*d*) contains n number of corpuses (*c*). Each corpus consists of r number of domains (*e*) and t number of documents (*f*). Each document in a corpus consists of p number of words (*w*) and q number of POS tag (*p*). Then,

$$d = \{c_1, c_2, c_3, \ldots\ldots, c_n\} \quad (10)$$

$$c = \sum_{i=1}^{r} e_i \times \sum_{j=1}^{t} f_j \quad (11)$$

Each document can be expressed as,

$$f = \sum_{r=1}^{p} w_r \times \sum_{s=1}^{q} p_s \quad (12)$$

We depicted our proposed model in Figure 1, which takes TDIL dataset as input and produces standard function words on the basis of their computed rank.

Steps of our model are as follows:

1. TDIL Bengali Dataset

The Bengali corpus is used in our work is developed under the Technology Development for the Indian Languages (TDIL) Project, Govt. of India [16]. TDIL dataset contains two corpuses: Monolingual and Parallel tagged. Monolingual-tagged corpus contains 7 domains: Art and Culture, Economy, Entertainment, Literature, Politics, Sports and Tourism. Parallel-tagged corpus contains 2 domains: Agriculture and Entertainment. TDIL dataset contains a set of total 9 domains, 49 datasets, 670,831 words and 134,884 distinct tokenized words with their POS tagging. For example, occurrences of POS tagged word like গাড়ি \N_NN occurred 50 times, চারদিক \N_NST occurred 11 times, আগে \PSP occurred 136 times in the corpus. Sample of dataset is as follows:

Table 1. TDIL Bengali Dataset

শিল্প \ N_NN - \ RD_PUNC কলা \ N_NN মানে \ N_NN মুখ্যমন্ত্রীর \ N_NN কাছে \ PSP নাচ \ N_NN , \ RD_PUNC গান \ N_NN , \ RD_PUNC ছবি \ N_NN আঁকা \ N_NN , \ RD_PUNC বাঁশের \ N_NN বাখারি \ N_NN থেকে \ PSP গুঁড়ো \ JJ মশলা \ N_NN তৈরি \ N_NN সবই \ QT_QTF | \ RD_PUNC

2. Text Cleaning

Any textual document is mixed with punctuation symbols, typographically mistaken characters, and alphanumeric characters. These are not played an important role in text processing and it tainted the performance of text processing. But, these symbols have highest TF in the dataset. So, it is desirable to be cleaned in pre-processing stage in text processing. Some symbols and characters are cleaned in our work like #, /, ,, �, ?, “, ‘, ’, @, ৳,\%, &, *, (), -, =, <>, ?, !, ।, ..., —, =-, `, ;, :, ``, =---,etc.

3. Tokenization of Bengali texts

Tokenization is a process of splitting the text into smaller parts called tokens. In our work, we perform a Java based implementation to tokenize TDIL Bengali dataset. Example of words after tokenization is as follows:

Table 2. Tokenized words

Words	Separator	POS Tag
শিল্প	/	N_NN
মুখ্যমন্ত্রীর	/	N_NN
জৈবিক	/	JJ
দিয়েছেন	/	V_VM_VF
কাছে	/	PSP

4. Identification of Unique Words

After tokenization process, a large number of duplicate words exist in extracted dataset. Existence of duplicate words creates a bottleneck situation in the system and overall performance is to be tainted. A Java program is implemented in our work for identification of unique words. Statistics is as follows:

Table 3. Unique words in Monolingual tagged corpus

Domain	Total words	Unique words	Identification
	Before tokenization	***Tokenized***	***%***
Art and Culture	29,238	9,932	33.97
Economy	29,546	7,299	24.70
Entertainment	63,441	16,040	25.28
Literature	53,215	14,190	26.67
Politics	70,043	12,156	17.36
Sports	64,017	15,140	23.65
Tourism	50,678	13,012	25.68

Table 4. Unique words in Parallel tagged corpus

Domain	Total words	Unique words	Identification
	Before tokenization	***Tokenized***	***%***
Agriculture	154,243	20,750	13.45
Entertainment	156,410	26,365	16.86

5. Generation of Document–Term Matrix

BOW or Vector–Space model [17] is a representation in NLP. A document is denoted as multi–set of words. The BOW model [18] is used in document classification where an occurrence of each word is used as feature. We implement a Java program to generate the Document-Term Matrix using

BOW model along with the support of a Java based tool [19]. Based on term frequency (TF), inverse document frequency (IDF) and generated Document-Term matrix; TF–IDF score is calculated in our work.

Table 5. Document-term matrix and TF-IDF score

Words	**POS**	**D1**	**D2**	**D3**	**D4**	**D5**	**TF-IDF score**
অভিযোগ	N_NN	10	1	2	0	4	0.012685759
অভিষেক	N_NNP	2	6	3	0	0	0.008493059
অসাধারণ	JJ	0	6	4	3	1	0.010447095
আগে	PSP	20	18	14	12	11	0.054461308
আছে	V_VAUX	2	22	41	11	18	0.068258173

6. **AFWI algorithm**

We draw the following flow chart for realization of AFWI algorithm. Our proposed system takes TDIL Bengali Dataset as input and produces 8 set of function words. Representation of Low, middle and high terms are realized as lower, average and higher valued terms which are anticipated in the following flow charts.

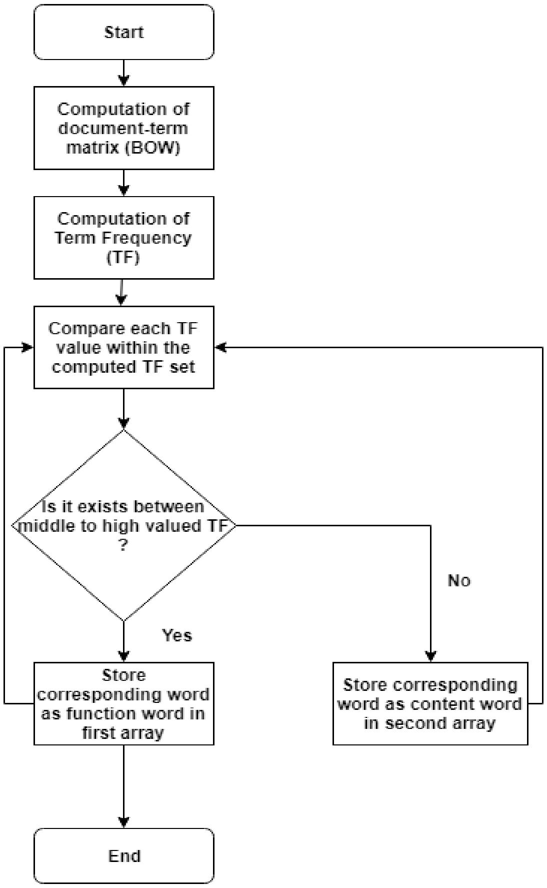
Start
Computation of document-term matrix (BOW)
Computation of Term Frequency (TF)
Compare each TF value within the computed TF set
Is it exists between middle to high valued TF ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 2: Flowchart for identification of function word set 1

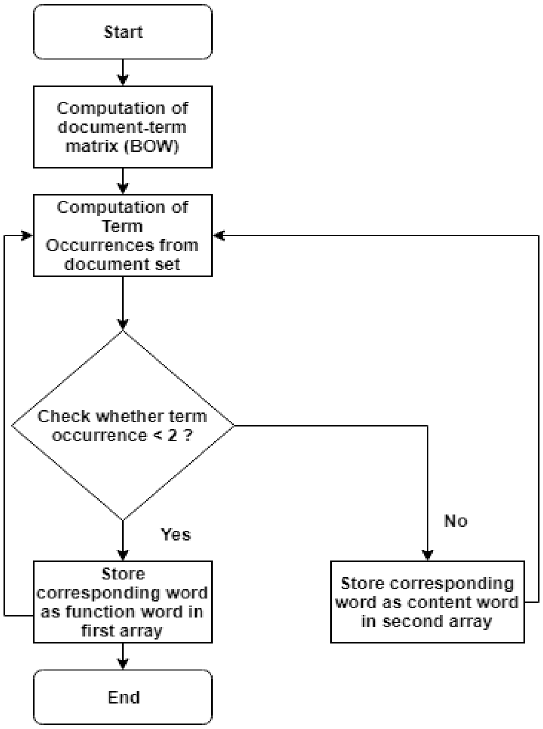
Start
Computation of document-term matrix (BOW)
Computation of Term Occurrences from document set
Check whether term occurrence < 2 ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 3: Flowchart for identification of function word set 2

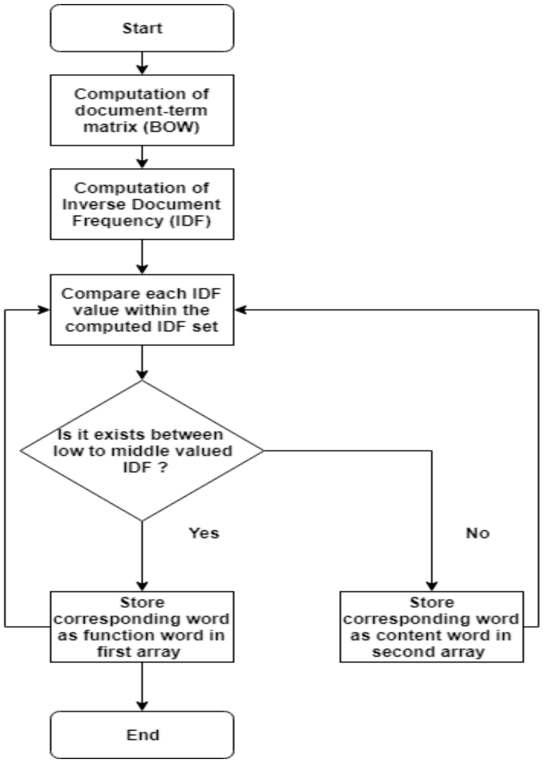
Start
Computation of document-term matrix (BOW)
Computation of Inverse Document Frequency (IDF)
Compare each IDF value within the computed IDF set
Is it exists between low to middle valued IDF ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 4: Flowchart for identification of function word set 3

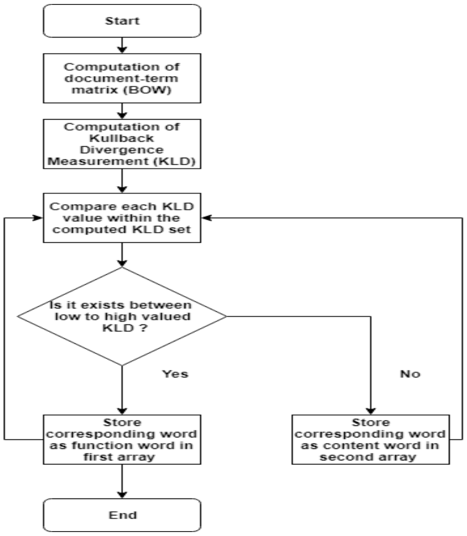
Start
Computation of document-term matrix (BOW)
Computation of Kullback Divergence Measurement (KLD)
Compare each KLD value within the computed KLD set
Is it exists between low to high valued KLD ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 5: Flowchart for identification of function word set 4

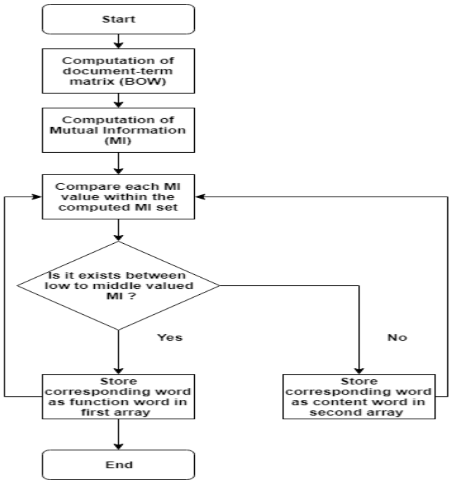
Start
Computation of document-term matrix (BOW)
Computation of Mutual Information (MI)
Compare each MI value within the computed MI set
Is it exists between low to middle valued MI ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 6: Flowchart for identification of function word set 5

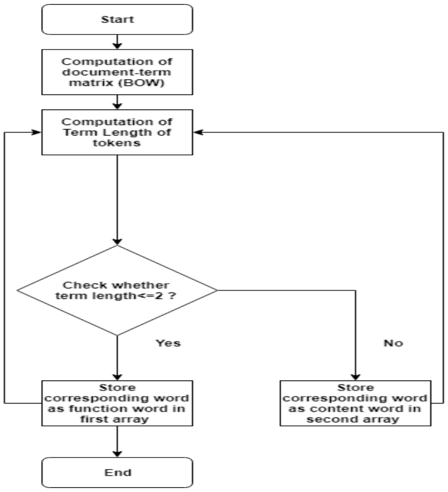
Start
Computation of document-term matrix (BOW)
Computation of Term Length of tokens
Check whether term length<=2 ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 7: Flowchart for identification of function word set 6

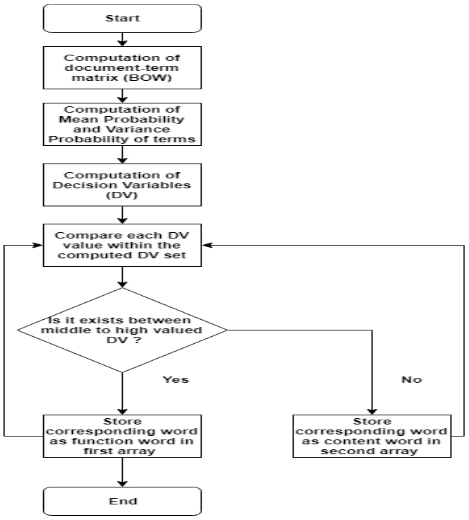
Start
Computation of document-term matrix (BOW)
Computation of Mean Probability and Variance Probability of terms
Computation of Decision Variables (DV)
Compare each DV value within the computed DV set
Is it exists between middle to high valued DV ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 8: Flowchart for identification of function word set 7

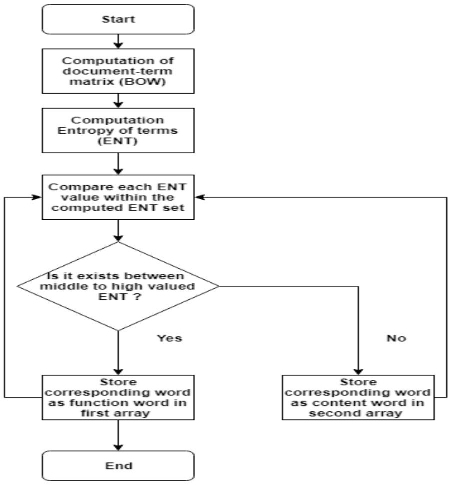
Start
Computation of document-term matrix (BOW)
Computation Entropy of terms (ENT)
Compare each ENT value within the computed ENT set
Is it exists between middle to high valued ENT ?
Yes
No
Store corresponding word as function word in first array
Store corresponding word as content word in second array
End

Figure 9: Flowchart for identification of function word set 8

Table 6. AFWI algorithm output

(***S1****: Function word set 1,* ***S2****: Function word set 2,* ***S3****: Function word set 3,* ***S4****: Function word set 4,* ***S5****: Function word set 5,* ***S6****: Function word set 6,* ***S7****: Function word set 7,* ***S8****: Function word set 8)*

Identified function words							
S1	***S2***	***S3***	***S4***	***S5***	***S6***	***S7***	***S8***
না	অংশকে	অত্যধিক	অক্ষয়	আমার	আই	আমার	অত্যধিক
আই	আমার	অনেক	অক্ষরে	আর	আছ	আর	অত্তা
করে	অংশগ্রহণ	অনেকদিন	অক্সফোর্ডে	এই	আজ	এই	অতলে
আছ	অকাতরে	অত্তনো	অকস্মাৎ	এক	আট	এক	অদ্বিতীয়
আর	অক্টোবরের	অপেক্ষা	অখণ্ড	একটা	আধ	একটা	অধিক

7. Optimization of identified function words

Our AFWI algorithm identifies 33,985 function words in Literature domain of TDIL Monolingual tagged Bengali corpus. It is observed that characteristics of identified function words contain a blend of repetition of words, semantically similar words and un-stemmed words. We perform the following steps to generate unique, stemmed and accurate function words. Complete optimization structure is composed by:

1. Identification of Unique Words

Table 6 illustrated that some function words are repeatedly identified by our proposed AFWI algorithm. Repetition of word in the result set will effect in overall system performance and accuracy measurement. We perform our unique word implementation program [III.4.] to find distinct words from the identified function words. As an outcome, 16,270 unique function words are identified. In this step, identified function words are 47.87% optimized.

2. Word Similarity Measurement

Several words exist with equal semantics in the obtained result set [7.1.] e.g. ’তার’ and ’তাঁর’, ’না’ and ’নি’, ’কি’ and ’কী’, etc. has equal semantics but they appears separately in the result set due to their POS tagging in TDIL corpus. We perform a Java based implementation using Levenshtein distance method to calculate word similarity score [20]. In this method, words with highest similarity score are removed. In this experiment, 407 words are removed due to their highest word similarity score. 15,817 words are selected for further optimization process, i.e. total 16,270 unique function words are 2.50% optimized. Statistics of optimization is illustrated in Table 7:

Table 7. Example of word similarity measurement

Word 1	Word 2	Word Similarity Score
কোম্পানির	কোম্পানীর	0.889
ভালোবাসা	ভালবাসা	0.875
হচ্ছিল	হচ্ছিলো	0.857
স্বর্বস্বহারা	সর্বস্বহারা	0.846
স্কুল	ইস্কুল	0.833

3. Stemming of Words

Our optimized result set [7.2.] contains several inflected words. Example of inflected words e.g. ’রবীন্দ্রনাথের’, ’সাহিত্যের’, ’ সাইকেলের’, ’অগ্রহায়ণের’. Existence of these words affects the system performance. Stemming is a process of reducing inflected words to their root. We implement a Java program to perform stemming operation on 15, 817 function words using the help of a rule based Bengali stemmer program [21]. Afterwards, we perform unique word identification program [III.4.] to remove 15,481 repetitive words and 336 unique words are selected. From 336 words, it is observed that 43 words are removed due to incorrect stemming and 293 are finally optimized i.e. total 336 unique function words is 87.50% correctly and 12.8% is wrongly stemmed.

At the end of optimization process, it is observed that 293 out of 33,985 words are selected as function words. But, the word set contains 1.02% single characters due to the effect of optimization method. These literals has a very few chance to use as function words. After removal of these literals, 290 words are finally selected as function words i.e. we achieved 99.14% optimized function words. Examples of function words are illustrated in Table 8:

Table 8. Example of optimized function words

Optimized function words
রবীন্দ্রনাথ, সাহিত্য, অন্য, করেছেন, চলেছ, টাকা, তিন, নিত, অবশ্য, এবার, কখন, হয়েছিল, আর্টিস্ট, মালবিকা, মাথায়, এসেছ, কথা, আমি

8. Word Score Generation

To generate the score of each optimized words, we use Z-Score method. It is the number of standard deviations from the mean of function word set. Word score (Z) is calculated as:

$$Z = \frac{(X - \mu)}{\sigma} \tag{13}$$

Where, X , μ , σ are denoted as occurrences of words, mean of the word set and standard deviation of the word set. High valued Z–Score indicates high weighted function word. Based on this score, high to low weighted function words are sorted.

IV. Results and Discussion

We performed our experiments through Java based implementation over TDIL tagged Bengali corpus. This corpus consists of total 670,831 words and 134,884 unique tokenized words. We execute our experiments on Intel Core2Quad 2.83 GHz processor and 8 GB RAM based system. To reduce the computation time, we perform our experiments on Literature domain of Monolingual tagged domain. After completion of experiment, our system identifies 33,985 function words. Following the identification process, we perform optimization processes on the result set to achieve 290 function words finally.

Computation for finding rank from optimized words is an essential task to entitlement standard function words. Finding rank of these words is computed on the basis of Z-score method. Higher valued Z-score indicates higher weighted function words. In this mechanism, higher to lower weighted function words are ranked. Higher Z-score valued words are expected to become standard function words. Example of function words score is illustrated in Table 9:

Table 9. Function word-score generation

Function words	Z-score	Rank
না	8.911044382	1
করে	5.988146629	2
আর	4.411752335	3
তার	4.280386144	4
এই	3.902708344	5

V. Conclusion and Future Scope

We perform our experiment on Literature domain of TDIL tagged Bengali corpus. Our system has predicted 290 standard function words out of 33,985 identified function words. Standardization is done on the basis of each word rank computation. System predicted function words may vary from human recognized function words. So, performance comparison between AFWI predicted words and human recognized words are required to improve our system. Due to unavailability of the standard Bengali function words for Literature domain, our implemented system lacks verification from such comparisons and accuracy measurement. In future, we perform such tasks to improve AFWI as domain independent system.

Acknowledgment

We acknowledge to Technology Development for Indian Languages Programme (TDIL), MeitY - Govt. of India and Indian Statistical Institute, Kolkata for their Bengali dataset support.

References

[1] F. Louise, F. Matt, "*Text Mining Handbook*", Casualty Actuarial Society E-Forum, CRC Press, pp. 1, 2010.

[2] Ministry of Electronics & Information Technology, Govt. of India, "*Technology Development for Indian Languages Programme (TDIL)*", Retrieved from http://www.tdil.meity.gov.in

[3] H. Saif, M. Fernández, Y. He, H. Alani, "*On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter*", Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Iceland, pp. 810-817, 2014.

[4] R.T.W. Lo, B. He, I. Ounis, "*Automatically Building a Stopword list for an Information Retrieval System*", Journal on Digital Information Management, Vol. 3, pp. 3-8, 2005.

[5] W.J. Wilbur, K. Sirotkin, "*The Automatic Identification of Stop words*", Journal of information science, Sage Publications Sage CA: Thousand Oaks, CA, Vol. 18, pp. 45–55, Issue.1, 1992.

[6] M. Makrehchi, M.S. Kamel, "*Automatic Extraction of Domain-Specific Stopwords from Labelled Documents*", Proceedings of Advances in Information Retrieval, 30th European Conference on {IR} Research, {ECIR}, Glasgow, UK, pp. 222-233, 2008.

[7] Asubiaro, T. Victor, "*Entropy-based Generic Stopwords list for Yoruba texts*", International Journal of Computer and Information Technology, Vol. 2, Issue. 5, 2013.

[8] M. Sadeghi, J. Vegas, "*Automatic Identification of Light Stop words for Persian information retrieval systems*", Journal of Information Science, Sage Publications Sage, UK, London, England, Vol. 40, pp. 476–487, Issue. 4, 2014.

[9] F. Zou, F.L. Wang, X. Deng, S. Han, L.S. Wang, "*Automatic Construction of Chinese Stop Word List*", Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, pp. 1010–1015, 2006.

[10] H. Lili, H. Lizhu, "*Automatic Identification of Stop words in Chinese Text Classification*", IEEE International Conference on Computer Science and Software Engineering, Vol. 1, pp. 718–722, 2008.

[11] S. Hassan, M. Fernandez, H. Alani, "*Automatic Stopword Generation using Contextual Semantics for Sentiment Analysis of Twitter*", Proceedings of the ISWC-2014 Posters and Demonstrations Track a track within the 13th International Semantic Web Conference (ISWC), Riva del Garda, Italy, pp. 281-284, 2014.

[12] Y.Z. Fard, M. Ali, M. Bidgoli, Behrouz, Rahmani, Saeed , Shahrivari, "*PSWG: An Automatic Stop-word List Generator for Persian Information Retrieval Systems based on Similarity Function & POS Information*", IEEE 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), pp. 111–117, 2015.

[13] R. Puri, R.P.S. Bedi, V. Goyal, "*Automated Stopwords Identification in Punjabi Documents*", International Journal of Engineering Sciences, Vol. 8, pp. 119–125, 2013.

[14] T. Cover, J.A. Thomas, "*Elements of information Theory*", John Wiley & Sons., 2012.

[15] Lin, Jianhua, "*Divergence measures based on the Shannon entropy*", IEEE Transactions on Information theory, Vol. 37, pp. 145-151, Issue. 1, 1991.

[16] N. Das, "*Indian Scenario in Language Corpus Generation*", Rainbow of linguistics, T. Media Publications, Kolkata, Vol. 1, pp. 129-162, 2007.

[17] G. Salton, A. Wong, C.S. Yang, "*A Vector Space Model for Automatic Indexing*", Communications of the ACM, Vol. 18, pp. 613–620, Issue.11, 1975.

[18] Z.S. Harris, "*Distributional Structure*", Word: Taylor and Francis. Vol. 10, pp. 146–162, Issue. 2, 1954.

[19] S. Roy, "*Bengali Document Ranking*", Github Inc., 2017.

[20] M. Bilenko, R.J. Mooney, "*Adaptive Duplicate Detection using Learnable String Similarity Measures*", Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 39–48, 2003.

[21] T. Nayak, "*Bengali Stemmer*", Github Inc., 2015.

Authors Profile

Mr. Subrata Pan is a research scholar of department of Computer Science and Engineering at Jadavpur University, Kolkata. He passed Bachelor of Engineering in Information Technology from University of Burdwan, Burdwan and Master of Technology in Information Technology from Bengal Engineering and Science University, Shibpur. Presently, he is working as Assistant Professor in the Department of Information Technology at Bankura Unnayani Institute of Engineering, Bankura.

Dr. Diganta Saha is an eminent Professor of Department of Computer Science and Engineering at Jadavpur University, Kolkata. His field of specializations are Machine Translation, Language Engineering, Mobile Database Management, Text Processing, Text Classification and Text Data Mining. He has 2 Book chapters, 15 Journals and 54 Conferences publications. He is carried out 7 Projects under UGC and DST.