Developing Assamese Information Retrieval System Considering NLP Techniques: *an attempt for a low resourced language*

Anup Kumar Barman[1], Jumi Sarmah[2], Shikhar Kr Sarma[3]

[1]Department of Information Technology, Central Institute of Technology, Kokrajhar
Kokrajhar - 783370, Assam. INDIA.
ak.barman@cit.ac.in

[2]Department of Information Technology, Gauhati University
Guwahati - 781014, Assam. INDIA.
jumis884@gmail.com

[3]Department of Information Technology, Gauhati University
Guwahati - 781014, Assam. INDIA.
sks001@gmail.com

Abstract: ***This paper engulfs the activities involved in developing a Monolingual Information Retrieval (IR) system for an Indo-Aryan language- Assamese. In a multilingual country like India, where 23 official languages exist, the task of digitizing local language contents is growing tremendously. To meet the need of each individual's relevant information, monolingual Information Retrieval in own language is very essential. The work aims to develop a search engine that retrieves relevant information for the fired query in one's respective language. Various Linguists, Researchers collaborated with the work, provided valuable information and developed various important resources. Many informative resources, language resources, tools & technologies were research, analyze, develop and applied in implementing the overall pipeline. The search engine is frame worked on open search platforms- Solr and Nutch with NLP applications embedded in it. Computational Linguistics or Natural Language Processing (NLP) enhances the performance of the IR system. Each phase of the system is being elaborately described in this paper and explained step-wise. This work is a remarkable contribution to Assamese language technology and an important application of NLP.***

Keywords: Information Retrieval, Natural Language Processing, Assamese Language

(Article history: Received: 30th November 2019 and accepted 22nd December 2019)

I. INTRODUCTION

In today's era, data, information, facts and knowledge are given prime concern than it was in two, three decades ago. Thanks, Internet! It is now possible to access anything, anytime, anywhere for gaining knowledge regarding any concept. Whenever we access any information from the online web repositories, it is the search engine that comes at this point. But, how do the search engines find the relevant information they provide us? This is where the concept of "Information Retrieval" comes.

Information Retrieval is the phenomenon of data storage, fetching, presentation and access to those data items. Extracting the user's expected information from a large text collection based on the query is the goal of an Information Retrieval(IR system). The number of web users is growing at a fast pace nowadays. Any information can be retrieved by web users anytime and at any place in this globe. But in a country like India, only 10% of the population speak English and 90% are not aware of the digitalized information on the Web as information is available in the English language. Language creates a great barrier for many people to access the digital world. There are traditionally two types of Information retrieval system: Monolingual Information Retrieval System (MLIR) which refers to that system that can retrieve the relevant information in the same language as the query fired by the user whereas Cross-Lingual Information Retrieval System (CLIR) is a sub-field of Information Retrieval dealing with retrieving information in the language different from the source (fired query) language. Our IR system facilitates the user to retrieve data in fired query language- Assamese.

The technology behind the IR system is based on two concepts- numerical and linguistic. The numerical concept reflects various numeric values collected from a document considering them as a bag-of-words. The linguistic concept may be language dependent or language independent but these both take care of the various linguistic phenomena and the underlying meaning of natural language text to fulfil the users need. Various similarity functions are used on the numerical or statistical values to produce the rank list by the numerical approach. The researches on both the IR approaches are initiated in a parallel manner even though in some time of IR research history the statistical IR has got more attention. From the last three, four decades a slightly different era of IR research has started incorporating both numeric and linguistic approaches to develop a more sophisticated IR system [1]. Boolean Model [2], Vector-Based Model [3], Probabilistic Model [4], Inference networks [5], Linear feature-based model [6] are treated as the statistical approach in IR technology. The various NLP applications with best accuracies incorporating with the IR system can increase the retrieval performance by providing the linguistic information along with the statistical data. This work also concentrates on investigating the NLP application of the Assamese language to feed the linguistic information to the IR system aimed to develop for this particular language.

The information of a various domains is scattered in the web in different formats. To collect all this information efficiently and in a faster mode, the IR system uses some web crawler algorithms. To retrieve fast and relevant content for a search query, Indexing is an essential component of an IR system. Indexing helps the IR system to get efficient searched results without consuming more time and with less computing. The ranking also called sorting is to display the top matched results of the fired query. The fired query is mapped with the index database (received after indexing) to get the results in sorted order according to the high-rank degree. After the three necessary and fundamental components of an IR system, the final output of the IR system is the searching technique. It is now becoming essential to get the searched results in a faster mode and an organized manner.

Assamese is one of India's 23rd official languages spoken by nearly 15 million people. It belongs to the Eastern group of Indo-Aryan language family. The main objective of developing the Monolingual IR system is to narrow down the gap that exists in the search of information. Developing a search engine would help the local masses to retrieve information in their native spoken language i.e., through which they communicate with one another. The search engine will facilitate human-machine interaction without any language barrier. The IR system will enable us to retrieve information on topics like historical places, temples, national parks of Assam along with some historical Indian places. IT researchers, linguists, data entry operators have worked on developing NLP resources to aid and build an efficient IR system. An Assamese IR system will facilitate internet users to access the digital world freely in their native language.

Some of the commonly used IR systems are Google, Yahoo!, and Bing etc. A general figure describing the IR system is given below in Figure 1.

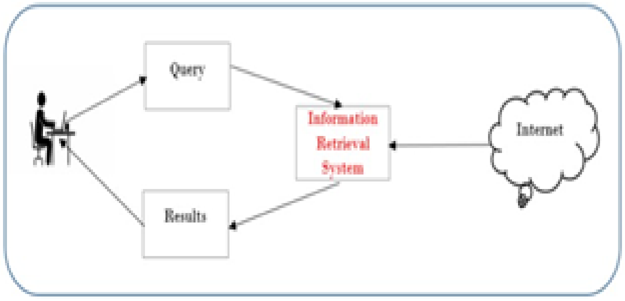


Fig. 1. The general figure describing an IR system

This paper gives a detail explanation of the various phases of the Assamese IR system. The paper is followed by an introduction to the various web systems, IR techniques in Section 2. Section 2 also describes the commonly used open source search frameworks for developing the IR system and also various evaluation methods to determine the efficiency of the IR system. Moreover, Section3 discusses various problems and challenges that came across during developing the monolingual IR system. The development pipeline of the Assamese IR system is mentioned in Section 4. The paper is concluded discussing the performance of the developed IR system in Section 5.

II. RELATED STUDY

The researches on the evolution of today's IR system have completed almost five or six decades. An interactive, efficient, commercial and intelligent IR system is the outcome of the IR research, which has now become an integral part of the people's life. The data search technique on a computer system started in the late 1940s. The advancement of computer technologies in terms of processing and data storing yields the improvement in IR performance. The concept of gathering, storing and retrieving information automatically has made a huge gap between today's IR and the traditional library system. Today's query-centric IR system can serve the people with various structured, unstructured and semi-structured information. The rate of increasing digitized information now can be conceptually mapped with the famous Moore's Law- "The number of transistors in a dense integrated circuit doubles approximately every two years" [7]. The high-speed Internet and the very huge amount of information made a chaotic situation for the information hungry people for finding his or her desired piece of information. It encouraged the IR researchers for developing a highly efficient web-search system and make the situation appealing. The development of the web-search systems can be categorized to three phases/ timeline.

Phase1 (1960-1970): The work is done on IR until 1950 concluded that computers were the ultimate device for IR. In the initial period of the 1960s, the researches on IR concentrated on- can a computer system enhance the IR system accuracy? A designated group led by Gerard Salton initiated the research on IR at Harvard University. They achieved some promising results and establish the fact that the abroad spectrum of researches is yet possible in IR. One of the IR research aspects is the standardization of ranking algorithms. They used the concept of vector to represent the queries and documents which was proposed by Switzer [8]. [9] Suggested the cosine similarity can be used to measure the similarity between query and document vectors. The concept of Feedback [10] was also another introducing aspect of IR research at this time-phase. It enhances the retrieval result iteratively manner by applying the previous search experience. This aspect is used in our modern search engine Google by linking related articles. The concept of document clustering and the machine learning used in IR is also another research finding of this era. The term association in query expansion also improve the IR accuracy by matching more number of documents with a user's query. Stemming acts as a baseline process for the term association concept by extracting the root form from the various inflectional form of a word. The book by [9] depicts the previous IR researches in the last decades. During this timeline, some companies emphasize on developing IR systems for some Govt. agencies. In 1966, an IR system was developed by the pioneer company – Dialog which was the first company devoted towards working on an IR as an initiation for NASA [11].

Phase2 (1970-1980): Luhn's term frequency weighting scheme is one of the essential contributions for IR research of this era. [12] On her research work establish the concept of inverse document frequency (idf) by reflecting that the less frequent words are more influential in the retrieval scenario. [13] Explained the new idea (tf*idf) by merging the two weighting scheme- tf (term frequency) and idf (inverse document frequency). A large number of researchers along-with Salton carried out their researches on standardizing the IR techniques putting more emphasis on vector space model (VSM) [14]. In the next two decades after this phase, many IR kinds of research stand as evidence of this VSM approach. Recent IR researches also use the vectors to represent the document and query.

Phase3 (1980-1990): In this era of IR research, the fruitfulness of previous findings such as tf-idf [15] score was viewed by applying those directly or with slight modification in the IR system. The researchers tried to incorporate the term frequency with the basic probabilistic model in an effective manner. BM25, a ranking model proposed by [16] was one of the promising findings which is commonly used in the IR system now also. The Latent Semantic Indexing (LSI) approach was introduced as an advanced version of the vector space model where the Singular Value Decomposition (SVD) is applied to reduce the space complexity [17]. Another initiative to reduce the query terms is by merging the semantically similar terms which as a result reflects the positive impact on retrieval. In this scenario of retrieval, the NLP applications to provide linguistic information like word structure and their semantics, information about collocations, grammatical categories etc., have got more attention by the IR researcher. WWW was introduced in late 1990 led to the tremendous growth of the web pages and to deal with those, the research on the web-search engines also attained a new era.

A. Various IR Techniques

Relevance Feedback: This technique is also called a query refinement technique that helps the user in searching for more relevant documents from the first set of retrieved documents for a query. Basically, in this phase, the user is displayed with a set of documents at first for a query and then he/she marks those as relevant. The first 10 or 20 retrieved web documents are needed to be examined. The main theme behind this technique is to mark the important terms/expressions of a retrieved document so that the query is refined with those marked terms and the process is iterated. It aims to identify and present before the user the relevant content instead of the non-relevant ones. Employing relevance feedback on the earliest SMART systems [18] to the latest probabilistic model [19] has shown improvements when evaluated on a small test collection.

Information Extraction: Another approach to getting the most useful information from the large web repository is information extraction. The IR system can gather the web content of different genres. An information extraction process initiates from the gathered web documents. The various raw data are transformed into information by an information extraction system so that the user can get his desired information in a very efficient and easily understandable format. Generally, an information extraction system separates the relevant text patterns at first, then extract the important information from the text patterns and then form an understandable piece of information for the user from those using some logical structure. In 1970, [20] developed an information extraction system at New York University. The system intends to extract the information related to the health domain particularly preparing the patient discharge summary. [21, 22] developed the information extraction system namely FRUMP in 1979 which enables retrieving information from the unclassified domain. JASPER [23], an IE system is used to extract information from the corporate data. To analyse the text, the system uses some template driven methods that enables to achieve Natural Language Processing capabilities.

Information Filtering: The way of retrieving relevant information from a collection of documents is the information filtering technique. The main goal of an Information Filtering system is to present before the user the necessary information by removing or filtering unwanted information from the information stream by applying some automatic techniques. On presenting the information to the user, the information filtering systems apply the user-preferences based newsfeed etc. to display the wanted information

B. Open source search Frameworks

There are lots of open-source frameworks available for developing a search engine. These frameworks provide multiple features for constructing the inverted index from the crawled web contents used for the IR system. All these frameworks provide the facility of customization for building an index and making it compatible with the needed application. This section describes some of the commonly used.

Apache Lucene: Lucene[24] is the most popular open source indexing software. It is not the complete search framework but an indexing library. It generates an inverted index from the documents in the crawler database. Apache Lucene is required to be plugged with a crawler (Apache Nutch) to index web documents. This indexing library is written entirely in JAVA by Doug Cutting and is a high-performance indexing software.

Apache Solr: Solr[25] is an undertaking indexing framework built on top of Apache Lucene. Some major features of Solr include- full document searching, real-time indexing, distributed indexing, dynamic clustering, integrating databases, rich document handling (e.g., Word, PDF). Besides indexing, Solr also has the features to add, modify and delete the documents present in the index database. Solr is considered the most popular enterprise search engine. Solr operates as a stand-alone full content search server. It adopts the Lucene Java library at its base for full-text indexing and search.

Apache Nutch: Nutch[26] is an open-source crawler software built in JAVA. This library for the crawling purpose was initiated by the Apache Lucene project. Nutch has a highly modular architecture for crawling web documents, allowing its developers to build plug-ins for text retrieval, querying, clustering. Nutch has a highly scalable crawler framework.

The literature review states some other open source search engines that are built on top of the Apache Lucene project are Regain[27], Oxyus [28]. A search engine named Swish-e [29] is recognized as a small search engine that is designed for crawling web documents but up to a limited size. Another full text search engine built using JAVA is MC4J [30].

A comparative study among the web crawlers is researched by [31]. The study reports that Lucene consumes more time for indexing purpose but index occupies less storage space. On the other hand, MC4J consumes less amount of time for the indexing task. This comparative analysis is made considering all certain parameters of the crawler. To date, no comparative analysis assuming all parameters of the web crawler is studied.

All these frameworks are built for generating a monolingual index. Here we have applied the Solr and Nutch framework for developing a search engine for the Assamese language.

C. Evaluation methods in IR

This section discusses different standard evaluation methods used in IR systems. It is needed to be mentioned here that the effectiveness of an IR system is dependent on the test data. One tester can take some custom form of test data for evaluation of the IR system whereas others can take some different forms of test data, thus we cannot predict which test data can perform better in a general way. For such reasons, a standard form of test collection is needed to be used by all to determine the performance of the IR system. The test collection should comprise of the following: documents, queries denoting the information need, relevance of each document concerning each query in the set. Cranfield, TREC, FIRE are some standard test collection set.

Precision: The precision metric measures the preciseness of the retrieval of IR. It measures the number of relevant documents among the retrieved documents. The precision metric doesn’t care if we retrieve all the relevant documents but deal with if irrelevant documents are retrieved.

MAP score: Mean Average Precision or MAP score gives the average precision at various cut-off ranks on the retrieved results by the web search systems.

Recall: Recall can be treated as the measure of the completeness of the IR process. It measures/ determines how much of the relevant set of documents are retrieved. It doesn't matter if some non-relevant documents are retrieved in the process. Both Precision and Recall are dependent measures thus if we try to increase the precision score the recall gets decreased and vice-versa. F-score: F-score is the harmonic mean of precision and recalls measure.

III. CHALLENGES IN DEVELOPING IR

Information retrieval systems have become an important component in everybody's life today to retrieve any kind of information on WWW in a fraction of second. For developing a successful IR system for a multilingual country like India, various challenges are faced at each level of developing the IR system. This section discusses the challenges faced during the tenure of developing a Monolingual IR system for the Assamese language.

The Indian Language (IL) IR challenges involved with the retrieval of most relevant information to fulfill the user's expectation from an automated system in an efficient and interactive way. Determining the architecture of such a system and the use of various data structures to store and access a large volume of data are some important concerns for the developers. Creating a search platform combining the search technologies and the linguistic information to get the relevant answers is what needed to be taken care of while developing IR

Analyzing and processing the user's query to determine the actual need of the user is a difficult task in developing an IR system. The same query may represent a different expectation of the user and that the IR system is considered as efficient which can retrieve the set of relevant information by distinguishing the actual sense of the user's query at each search instance.

As the internet is flooding with huge storage of data in seconds, it makes the IR task more and more challenging. To retrieve information from voluminous heterogeneous data arising from various platforms like Twitter, Facebook, etc., makes developing the IR system challenging. Various statistical Machine Learning techniques can be applied for the development of the IR system but those measures alone cannot develop an efficient IR system. Some linguistic techniques can be applied along with statistical measures for achieving retrieval efficiency. Selecting the appropriate linguistic application and fixing it in the phase where it needs to get embedded is not always straightforward.

Various open source software are applied to develop IR systems. Open source search frameworks are the software with the source codes where the developers can inspect them, customize and enhance it according to their needs. Configuring the soft codes to develop an IR system for a specific language is a tough job. The parameters are needed to be set appropriately while configuring the software, run and build in a parallel manner to develop the modules and form an IR system. This is an important concern as well as a major challenge for researchers and developers.

Implementing a linguistically motivated IR system for the resource-scarce language like Assamese preparing linguistic resources becomes an important concern. Resources like stemmer, list of MWEs or NEs to feed to the IR system are required to be developed correspondingly to achieve an efficient IR.

IV. METHODOLOGY

IR system begins with firing a query in the search box by web users in need of some information. Some people fire query specifically in one or two keywords related to the domain of knowledge he/she wants to retrieve and some others don't have enough knowledge about firing queries. The words in his/her mind are fired as queries regarding the information he/she seeks to (no matter how many numbers of keywords are fired in the search box). The user looks upon the searched ranked results, if the searched results are not satisfied he/she tries to restructure or redefine the fired query. Users are not generally aware of the infrastructure of the IR system, thus the IR system is like a black box to them. What is more concerned to them are the retrieved results by the search systems. But the computer researchers always want to know how the IR system works, what are the algorithms needed to develop IR system, how the performance of the IR system could be improved. These questions in our mind led us to perform research on this topic and try our hand in developing an IR system for one of the official languages of Assam. The overview of Assamese IR system is shown in Figure 2. The IR system is implemented in two modes-offline mode and online mode. In online mode, the IR system activates on receiving query by the user. The query is processed at first- Morphological Analyser analyses the query terms by removing the suffixes, reducing it to the root word form. Later, NE list and MWE list is looked-up to identify the phrases if exists for efficient search retrieval. After query formulation IR system searches the index data structure to help the user retrieve the search results. Also, the index data are influenced with the linguistic components for efficient results. In the online mode, a set of seed URL is crawled from the Internet and the contents are fetched, parsed, indexed to store in the index data structure. Later, the output generation module helps the user in displaying ordered and most relevant search results. The pipeline is described below with the various resources developed (4.1, 4.2, 4.3) and Assamese IR System Development Module (4.4). The various resources developed are:

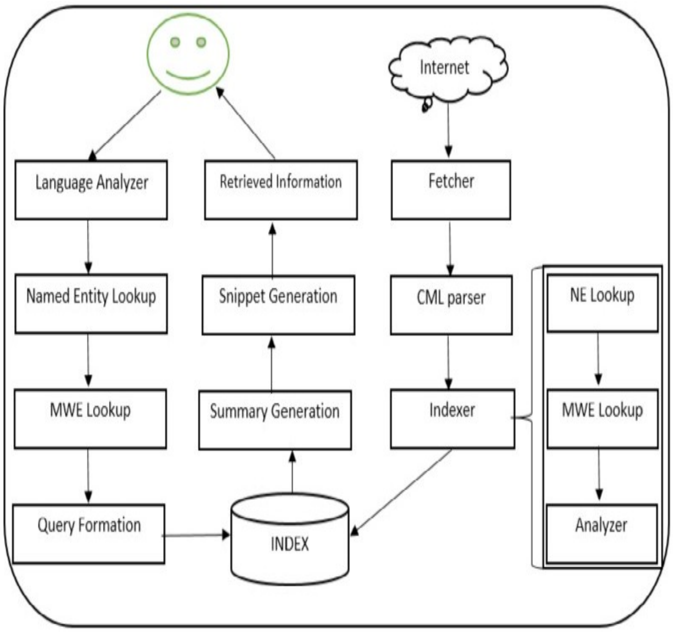


Fig. 2. Assamese IR architecture

A. Developing Language resources

Language resources basically are developed for assisting linguistics in their research work and fieldwork. Linguist develops such resources. Those resources basically refer to data-only resources such as corpora, dictionary, named-entity lists etc. Such resources can be used in a number of intellectual disciplines.

Named Entity List: Named-Entity recognition is a subtask of Information extraction which classifies information in the text to some predefined categories like Person names, organization names, location names, numbers etc. A list comprising of 105905 (One lakh five thousand nine hundred five) Assamese named entities was manually created. The list consist of Assamese Region-specific NEs and which are categorized as Organization (ছাত্র সন্থা [Caatro sontha]), Names (পঙ্কজ [Pankaj]), Festival (ৰঙালী বিহু [Rongali Bihu]), Flower (গোলাপ [Gulap]), Folk Instruments(বাঁহী [Bahi]), Food (ভাত [Bhaat]), Games (ঢোপ খেল [Dhup Khel]), Honorific title (জয়াল [Joyal]), Measurement (যোগ [Jug]), Place Name (তেজপুৰ [Tezpur]), Plants (আঁহত গছ [Aahot Gos]), Birds (ভাটৌ [Bhatou]), Religious Places (পোৱা মক্কা [Puwa Mucca]), Tourist Places (কাজিৰঙা [Kaziranga]).

Spell variation list: A spelling variant of a word occurs when a word may not have a single correct spelling and there are many of different ways in which it can be spelled. For example, if two persons spell the word “জৰ (Jor)” as “জৰ Jor)" and the other as “জ্বৰ (Jor)” than this qualifies as spelling variation. So, on searching the word as a query both the word containing documents should be retrieved by the web users. Also, if the term “কাজী (Kaji)” is spelled by two variables say x spelled as "কাঁজি (Kaji)" and y as "কাজী (Kaji)" then this signifies two different concepts and cannot be grouped in the same web-based pages retrieved on giving the user query. A spell variation list of some local based (Assamese region) and other Named entity consisting of 3539 and 1631 entries respectively was manually compiled.

Multiword Expressions (MWEs) List: are the sequence of words separated by space or delimiter (denoted by -) which determines a unique meaning instead of words' individual meanings. A list comprising of 1627 Multi-word Expressions have been identified for the Assamese language. Some of the Assamese MWEs are “হাড় ছাল” (Haar Saal), “খটক খটক (Khotok Khotok)” etc. MWEs helps in retrieving important results on given a query string. As for example, on firing a query “হাড় ছাল(Haar Saal)” if this query is not considered as a single collocation than the IR system would retrieve us separate results on হাড়(Haar) and ছাল(Saal) which means “bones” and “skin” respectively. Representing relevant MWEs to the search engine would gain relevant searches from Assamese IR system.

Assamese Stop-word List: The most frequently occurring words in a text are stop-word. They do not play an important role in retrieving information and they should be removed before processing of the query string by an IR or during the indexing of the crawled data. These words have very low discrimination value and are sometimes referred to as noisy words. Assamese stop word list is created by Linguistic scholars which contain 264 words. Some examples are যেতিয়া, যেন, যেনিবা, যেনে, যোগে, লগ, লৈ (Jetia, Jen, Jeniba, Jene, Juge, Log, Loi) etc.

Assamese Dictionary: In the development phase of the Assamese monolingual system, a root word list consisting of 15,750 words was manually created. The root list will facilitate the tasks of Morphological analysis.

B. Developing Informative resources

Informative resources provide valuable and important information from them. Such resource refers to query database, URLs etc. Compiling such information from various sources is an important task assigned to us.

Assamese Query database: Queries provide a way to retrieve important documents or results from a web database through an IR system. Tourism based queries were manually created and a number of query based on Assam Region was 2037. Some examples of queries are লক্ষ্মীনাথ বেজবৰুৱা [Lakhminath Bezbaruah], চেৰা বিহু [Sera Bihu], চুটি গল্প [Suti Golpo] etc. Such queries were used for testing the Assamese IR system.

URL database: A list of Govt. and General Assamese URL's were identified by our linguistic team for crawling purpose. 147 numbers of Assamese seed URL and 129 number of blog URLs were identified. Total URLs=276.

C. Developing Processing Resources

a) Assamese Morphological Analyzer: Stemming is a process used in morphological analysis and IR to reduce the inflected words to their base or root form. Generally, stemming a word means to reduce the inflected term to a written word form not necessary that the stemmed word is identical with the morphological root of the word. A stemmer is also developed for the Assamese language which outputs with a good accuracy of about 85% shown in [32] and is embedded with the search engine. A rule-based method with dictionary look-up approach is used to develop the Assamese Stemmer.

b) Language Identifier: The technology of Language identification has become more important with the growth of WWW. Automatic language detection on written texts, also known as language identification is basically a categorization task. If a query is placed in one language say Assamese than the system has to recognize that language as a first preprocessing step and later understand and retrieve relevant documents in that particular language. Also for developing Multilingual IR system if a query is placed in Assamese language and the results are to be retrieved in some other language say English than it is necessary to correctly recognize the language, translate the query into the language of the respective order. An abbreviated Assamese list recognizes the language.

c) POS Tagger: POS tagger automatically labels POS to a word depending on the particular context. Assamese POS Tagger is implemented by stochastic tagger i.e., by using Conditional Random Field (CRF) and Transformation-Based Learning (TBL) for the Assamese language. We obtain 87.17% and 67.73 % tagging accuracy for TBL and CRF respectively which was trained through a manually tagged 140000 word corpus, mentioned in [33]. With the help of POS tagger, the IR accuracy is improved. Also when words are labeled with POS tasks like extracting MWEs, recognizing named-entities becomes easier.

d) Named Entity Recognizer: Named Entity is a text element indicating the name of a person, organization and location. Named Entity Recognition is a task of two stages - first to identify the proper nouns and then to classify the proper names into categories such as person name, organization names, location etc. The POS tagged words provide a powerful feature in recognizing Named Entities. Named-Entities also provide a useful indexing feature in IR tasks. As for example “গুৱাহাটি বিশ্ববিদ্যালয় [Gauhati Biswavidyalay]” when treated as a Named entity with Organization tag then web pages related to it is going to be retrieved by web users. Else if not considered as NE then গুৱাহাটি [Gauhati] as location name and বিশ্ববিদ্যালয় [Biswavidyalay] as university separate web pages will be retrieved by the search engine. But there are some issues to be handled in case of Named Entities like

- Ambiguous word "কবিতা" [Kobita]. It means a name of a person or a poem.
- Secondly, Spelling variation of শ্ৰী শ্ৰীশান্ত [Sri Srisanta]. Whether শ্ৰী [Sri] in শ্ৰীশান্ত [Srisanta] is a pre-nominal word or a named entity? NER using HMM is applied and implemented for the Assamese language.

e) MWEs identifier: Statistical measure- PMI (Pointwise Mutual Information) & Chi-Square and Language specific rules helped Assamese computational scholars to automatically extract MWEs from the raw corpus. From Assamese WordNet number of possible Assamese MWEs extracted was 4891. Automatic Identification of Assamese and Bodo MWEs through statistical approaches & linguistic rules is being developed for the Assamese language [34]. Also, WordNet based IR system is developed for Assamese Language by [35].

D. Assamese IR System Development Module

a) Query Processing Module: Query module is the first processing stage of an IR system. Various forms of query like short query, long query, and narrative query can be inputted in the search module. This module at first analyse the query fired by the user by applying some language processing applications and later provides them as an input to the search subsystem the next important phase of the IR system. The language processing applications comprise of the following tasks shown in Figure. 3

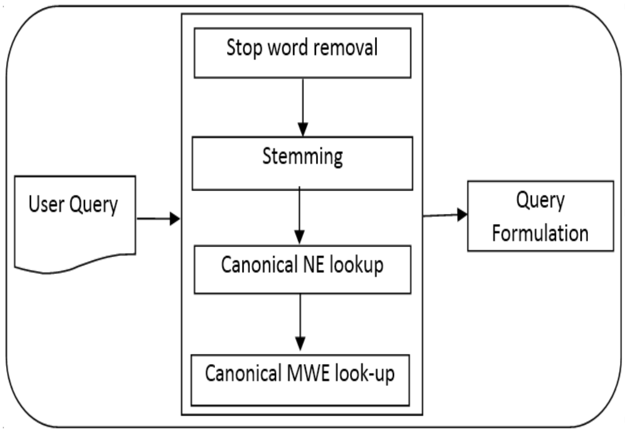


Fig. 3. Query Processing Module (QPM)

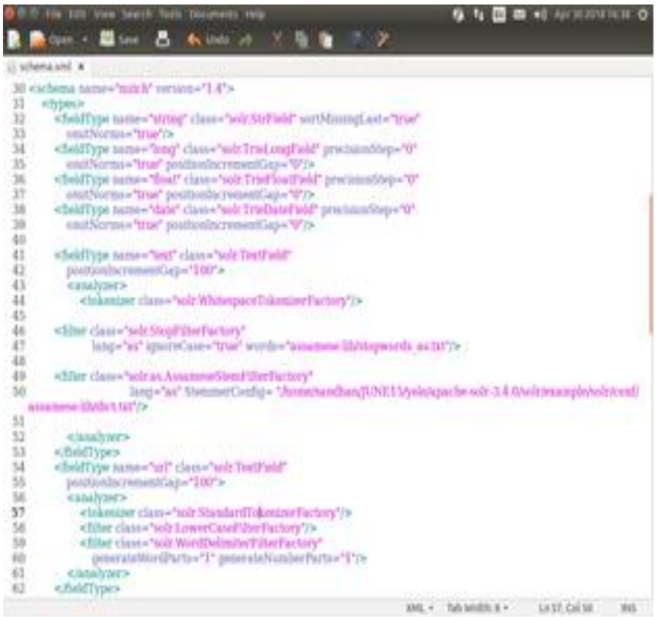
For our implementation, Solr-the open source search platform is used. Solr provides a user-friendly search interface by enabling an easily customizable environment through the solrconfig.xml file. To process the user inputted query the extended Dismax (eDisMax) parser is used. Insolrconfig.xml parsing mode is specified through setting defType parameter as- <str name=”defType”>edismax</str>. This parser process starts with the simple phrases fired by the user as a query. Then it parses the query based on the various predefine components or fields and prepares the search terms considering the various boost scores for different fields. A simplified version of syntax from the Lucene query parser issued by the DisMaxparser. The eDisMax support the AND, OR, NOT, + and – clauses alongwith the basic functionalities of DisMaxparser. The mm parameter representing the minimum match available in solrconfig.xml indicate the minimum number of query terms should match with the document. By default, the mm=1 means at least one query term should appear on each of the retrieved document. Considering a snap of parser query edismax (title:অসম 3̂.0|content:অসম 1̂0.0), here the edismax function takes অসম (Asom) as a query word and it will search the relevant results for the query by considering the fields- title and content. The score will be calculated as the maximum score of these two rather than the sum of the score for these two. The boost value for various field of the query terms used by the edismax parser is specified in the solrconfig.xml file. The boost values are as follows: <str name=”qf”>url3̂.0 content1̂0.0 title3̂.0host2̂.0</str>. It indicates the boost score 3.0, 10.0, 3.0, 2.0 for the fields url, content, title, host respectively. The MWE and NE found in the query are boost by values 5.0 which is specified in the parser program.

b) Search Subsystem: After firing query in the search box and pressing the search button by the user, at first the query processing module gets activated. The QPM refines the search terms by filtering the stopwords, removing the inflections, looking-up in the MWE and NE lists and later passes it on to the next subsystem of the IR-the search/retrieval module. The search module can be considered as the heart of an Information Retrieval system. The main activities performed by this module are coined below:

- The web is crawled for some seed URLs up to a certain depth (the level up to which the web contents will be crawled, may be 2, 3etc. depending on the required application) and later the files are downloaded. The contents of the files downloaded are for a specific language and domain. For Assamese IR system, we have tried to perform crawling on the web URLs whose contents are available in the Assamese language.
- The contents/texts are extracted from the files downloaded and some pre-processing is performed on them. Later, the extracted texts are converted to indices.
- The results of a searched query are extracted by a look-up approach. The searched keywords are looked-up in the indices built for efficient and fast retrieval.
- The pages returned by the search system are further sorted by applying some ranking approaches and most relevant results are retrieved to the web user by the search subsystem.

Each of the activities performed by the search module internally involves lots of computational processing. That computational processing tasks involved in each phase are explained in the next subsystem-Information processing module elaborately.

c) *Information Processing module:* The Assamese search engine is comprised of two parts: SOLR- the search engine interface to the Apache Lucene search library and the NUTCH which is the open source software web crawler used to index web content. First of all both Nutch1.4 and Solr3.4 was installed to a LINUX programming environment along with software like Apache2, OpenJDK-1.7

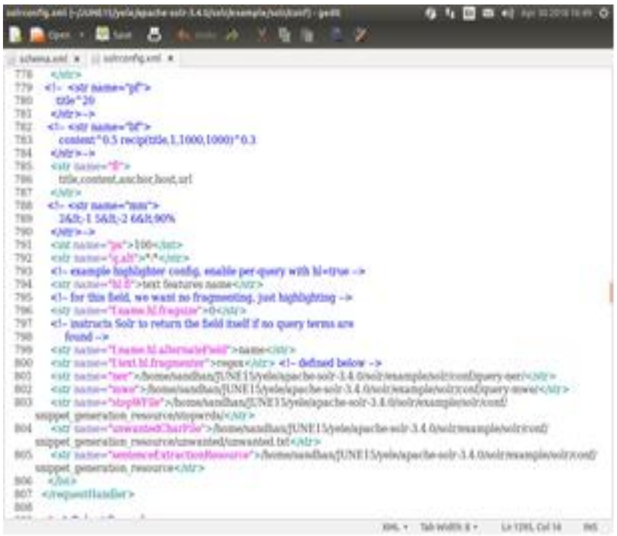


```
<schema name="nutch" version="1.4">
  <types>
    <fieldType name="string" class="solr.StrField" sortMissingLast="true"
      omitNorms="true"/>
    <fieldType name="long" class="solr.TrieLongField" precisionStep="0"
      omitNorms="true" positionIncrementGap="0"/>
    <fieldType name="float" class="solr.TrieFloatField" precisionStep="0"
      omitNorms="true" positionIncrementGap="0"/>
    <fieldType name="date" class="solr.TrieDateField" precisionStep="0"
      omitNorms="true" positionIncrementGap="0"/>

    <fieldType name="text" class="solr.TextField"
      positionIncrementGap="100">
      <analyzer>
        <tokenizer class="solr.WhitespaceTokenizerFactory"/>

    <filter class="solr.StopFilterFactory"
          lang="as" ignoreCase="true" words="assamese-lib/stopwords_as.txt"/>

    <filter class="solr.as.AssameseStemFilterFactory"
                lang="as" StemmerConfig= "/home/nandhan/JUNE15/yela/apache-solr-3.4.0/solr/example/solr/conf/
assamese-lib/dict.txt"/>

      </analyzer>
    </fieldType>
    <fieldType name="url" class="solr.TextField"
      positionIncrementGap="100">
      <analyzer>
        <tokenizer class="solr.StandardTokenizerFactory"/>
        <filter class="solr.LowerCaseFilterFactory"/>
        <filter class="solr.WordDelimiterFilterFactory"
          generateWordParts="1" generateNumberParts="1"/>
      </analyzer>
    </fieldType>
```

Fig. 4. Snapshot of schema.xml file

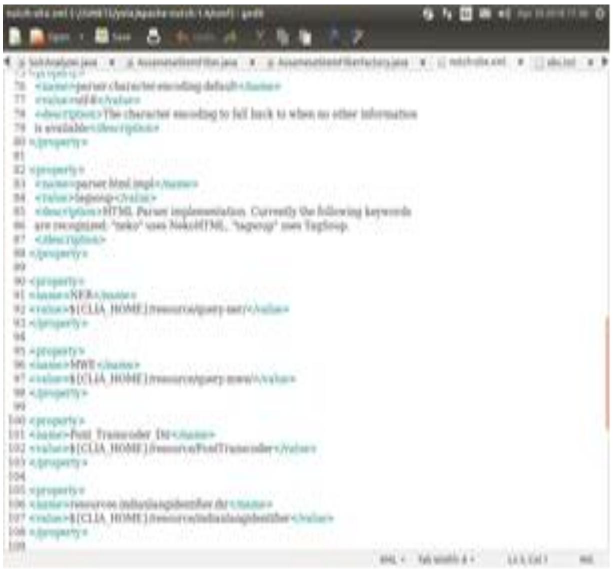
The basic configuration of Solr is made in both solr-config.xml and schema.xml file. Snapshots are shown in Figure 4 and 5. Solrconfig.xml is used to configure the Solr instances. The Data Directory specification to hold the indexes in Solr and the cache configuration is done here. Schema.xml lists down the filters which is to be called upon a token stream is a list of tokens of the document that is needed to be indexed. The Solr architecture applies the filters mentioned in schema.xml on the token stream given.

Later navigating to solr/example we need to start the command in the Linux terminal: java -jar start.jar Before crawling any data we need to set some properties on the file nutch-site.xml of the /conf directory. A snap is shown in Figure6.



```
  </str>
<!-- <str name="pf">
   title^20
  </str>-->
 <!-- <str name="bf">
   content^0.5 recip(title,1,1000,1000)^0.3
  </str>-->
  <str name="fl">
   title,content,anchor,host,url
  </str>
 <!-- <str name="mm">
   2&lt;-1 5&lt;-2 6&lt;90%
  </str>-->
  <int name="ps">100</int>
  <str name="q.alt">*:*</str>
  <!-- example highlighter config, enable per-query with hl=true -->
  <str name="hl.fl">text features name</str>
  <!-- for this field, we want no fragmenting, just highlighting -->
  <str name="f.name.hl.fragsize">0</str>
  <!-- instructs Solr to return the field itself if no query terms are
     found -->
  <str name="f.name.hl.alternateField">name</str>
  <str name="f.text.hl.fragmenter">regex</str> <!-- defined below -->
  <str name="ner">/home/sandhan/JUNE15/yele/apache-solr-3.4.0/solr/example/solr/conf/query-ner/</str>
  <str name="mwe">/home/sandhan/JUNE15/yele/apache-solr-3.4.0/solr/example/solr/conf/query-mwe/</str>
  <str name="stopWFile">/home/sandhan/JUNE15/yele/apache-solr-3.4.0/solr/example/solr/conf/
snippet_generation_resource/stopwords/</str>
  <str name="unwantedCharFile">/home/sandhan/JUNE15/yele/apache-solr-3.4.0/solr/example/solr/conf/
snippet_generation_resource/unwanted/unwanted.txt</str>
  <str name="sentenceExtractionResource">/home/sandhan/JUNE15/yele/apache-solr-3.4.0/solr/example/solr/conf/
snippet_generation_resource</str>
 </lst>
</requestHandler>
```

Fig. 5. Snapshot of solr-config.xml



```
 <name>parser.character.encoding.default</name>
 <value>utf-8</value>
 <description>The character encoding to fall back to when no other information
 is available</description>
</property>

<property>
 <name>parser.html.impl</name>
 <value>tagsoup</value>
 <description>HTML Parser implementation. Currently the following keywords
 are recognized: "neko" uses NekoHTML, "tagsoup" uses TagSoup.
 </description>
</property>

<property>
<name>NER</name>
<value>${CLIA_HOME}/resource/query-ner/</value>
</property>

<property>
<name>MWE</name>
<value>${CLIA_HOME}/resource/query-mwe/</value>
</property>

<property>
<name>Font Transcoder Dir</name>
<value>${CLIA_HOME}/resource/FontTranscoder</value>
</property>

<property>
<name>resource.indianlangidentifier.dir</name>
<value>${CLIA_HOME}/resource/indianlangidentifier</value>
</property>
```

Fig. 6. Customized nutch-site.xml

This basically sets the userAgent property used in the HTTP request headers when Nutch hits a site. Then moving back to /nutch directory we need to run the following command : cd/runtime/local/bin/**.** But there reports an IOException error which overcomes by adding the command: export ANT_OPTS="-Dhttp.proxyHost=10.10.127.3-Dhttp.proxyPort = 3128". Once, the error encountered is removed we can start crawling our data. Before we can do that, we need to tell Nutch the file to crawl and it is done by creating a file of the URLS we wish to spider or crawl. We need to create a new directory in our nutch folder called /url and then create a text file within it called url_10.txt. An example of Assamese URL is: http://xondhan.com/. Now we can start crawling and is done by firing the following command:

bin/nutch crawl url -dir crawl -depth 3 -topN 50

Executing this command shows us an error- Error: JAVA_HOME is not set**.** So, we need to set the path: export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-i386 before executing the crawl command. Now Nutch will spider or crawl each URL in the file in /url and build a crawl database. The CML parser parses the available raw content from the web. From the parsed content the available outlinks, scores, signatures, metadata are kept and these are essential to update crawldb. The outlinks play a vital role in crawling by indicating the crawler where to go next. A database consisting of /crawldb, /linkdb and /segments is formed after crawling. Once the crawling is done Indexing is required to be made. The index database is also facilitated with the NLP tools and technologies.

With Solr running in the back-end, Nutch data can be also put by using the command: bin/nutch solrindex http://172.16.3.73: 8983/solr /crawl/crawldb /crawl/linkdb /crawl/segments. After successful execution of this command, index file is generated at solr /data directory . Solr comes with a default web interface (Figure8) which allows us to search. We can access it at http://localhost:8983/solr/admin/. On entering some text into the Query-String box and hitting "Search" if our query matches any results we can visualize the output in a file of the XML format.

d) Output generation subsystem

Output generation system gives the user the final output of the IR system. This section discusses two important modules of the output generation system- snippet generation module and a summary generation module. Both the modules are individually important for displaying the output of an IR system. Snippet generation module is the major output module of Assamese IR system which generates and later displays the snippet of individual retrieved documents. Snippet provides salient information about a document. Also, summary generation module generates a summary of the documents which may be query-biased or general.

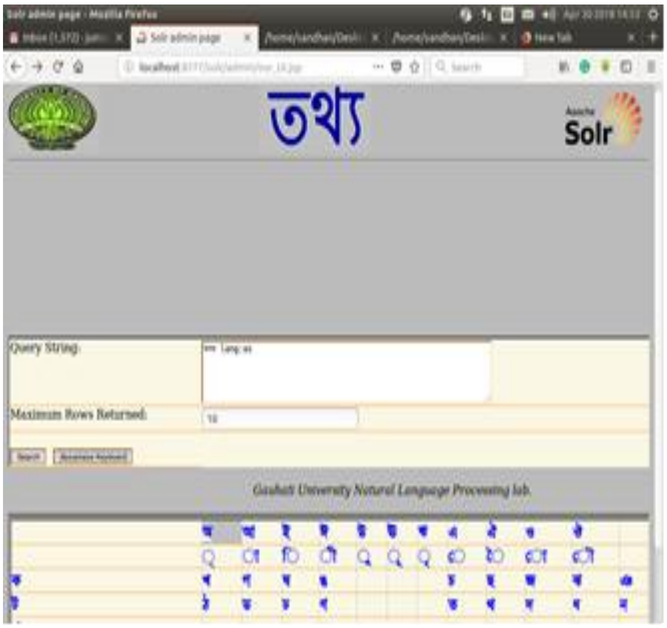


Fig. 7. Customized search interface

Summarization can be made in two ways-extractive and abstractive summarization. Both these two modules-Snippet generation module and summary generation module go through three common phases- 1) Keyword extraction, 2) Sentence extraction, 3) Top sentences identification. After processing phase 3 two individual phases are implemented A) summary generation module B) Snippet identification and generation module. A customized search interface of Solr is shown in Figure 7 where how the query can be fired is also shown. An output representation is shown in Figure 8 and Figure 9.

Fig. 8. Search results of Assamese IR

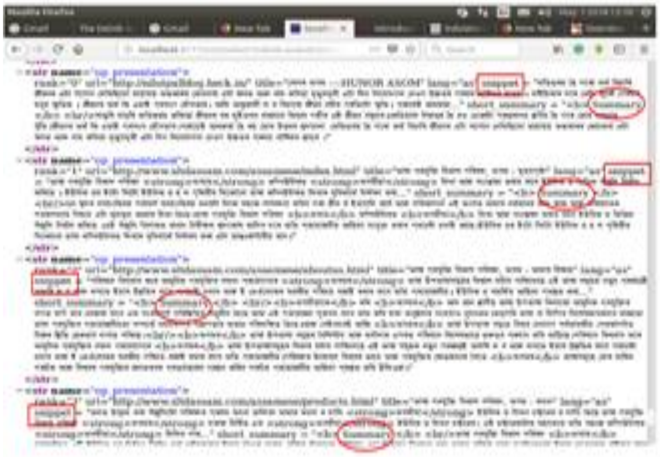


Fig. 9. Snippet and Summary results

V. TESTING & EVALUATION

We evaluated query processing module, retrieval performance of the Assamese IR system.

A. Query Processing evaluation

The query performance was evaluated by analyzing each module like- if Stemmer does stemming properly, if MWE, NE is correctly detected, stop word is removed or not? It is found that the query performance of the Assamese IR system is reasonably good as MWE, NE's are correctly detected and stemming is perfumed accurately to the inflected terms. Graphical results are shown in Figure10.

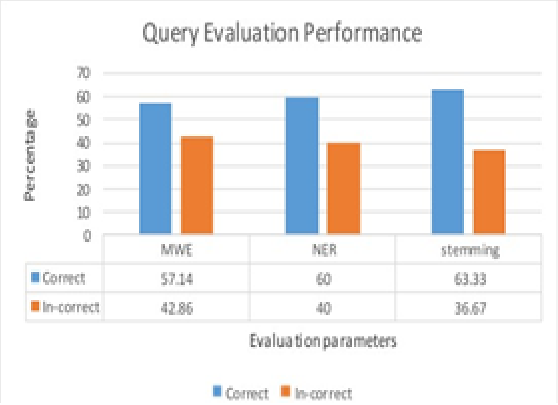


Fig. 10. Evaluation of Query Processing Module

B. Retrieval Performance

Retrieval performance of the IR system was based on General Queries and Regional queries. It was measured using p@k metric and later MAP (Mean Average Precision) score gives us the result. k indicate here the number of documents. The values 0-irrelevant; 0.25-low relevant; .5-partial relevant; 1- total relevant are considered for evaluation. Figure 11 and Figure12 shows the result of the 20 general queries and 20 regional query.

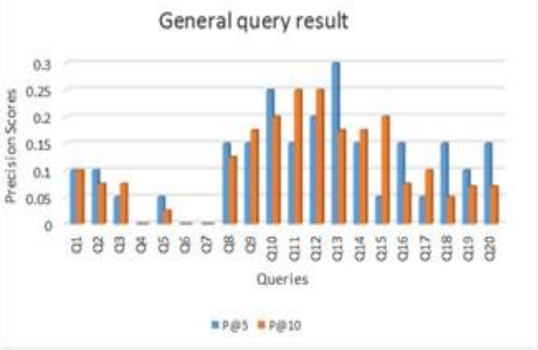


Fig. 11. The performance of IR system-20 general queries

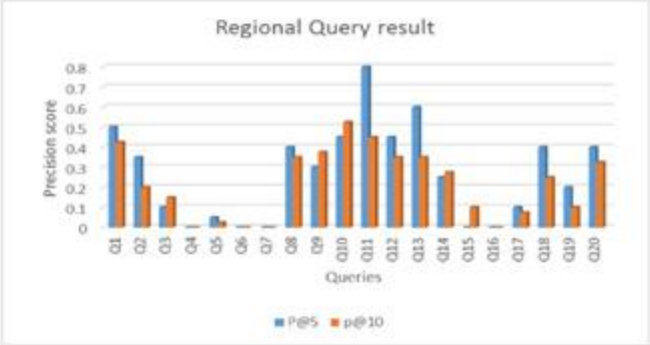


Fig. 12. The performance of IR system-20 regional queries

While evaluating the performance of the IR system, we derived that the IR system performs well for regional query (MAP score: 0.35) than the general query. It is such that few URLs contains contents in the web confined to general category in Assamese language compared to the regional ones.

VI. CONCLUSION AND FURTHER WORK

This paper gives us an overview of the work done by the Assamese NLP team to develop IR system. No such work has been reported to date. Various problems, challenges, resources compiled and developed is mentioned in this paper. Plug-ins like Word Sense disambiguation module can be embedded to the IR system to retrieve efficient results. Assigning correct sense tag to the ambiguous term will retrieve us correct results. It will be also helpful in case of query translation or in document translation.

In this work, we discuss the importance of Information Retrieval system concentrating on the development of a linguistically motivated IR system for the specific language-Assamese. Developing the IR system facilitates the people to know and understand the information s/he wants in their familiar language.

Over this period of research for developing the Assamese Information Retrieval system, a number of avenues for further research arise. Those are: Developing a Multilingual Information Retrieval system (E.g.: Hindi-Assamese, English-Assamese), Developing Medical information retrieval system, Developing an IR system related to any tasks of e-governance.

As the day goes by we will continue to witness more qualitative and quantitative web content and developing a more sophisticated automatic IR systems will enable Assamese community people to upgrade their knowledge without any hassle.

ACKNOWLEDGMENT

We would like to thank Deity, TDIL & MHRD for sponsoring the CLIA project-Phase II and allowing Assamese language to be a part of it. Moreover, we would like to thank CLIA consortium members, PIs, research scholars, research assistants for their valuable guidance, support and collaboration

REFERENCES

[1] T. Brants and Google Inc, “Natural language processing in information retrieval,” in Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands, pp. 1–13, 2004.

[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. Modern Information Retrieval, ACM Press, 1999.

[3] H. Joho and M. Sanderson, “Document frequency and term specificity”, In the Recherche d’ Information Assiste par Ordinateur Conference (RIAO), 2007.

[4] Roman W. Swiniarski Lukasz A. Kurgan Krzysztof J. Cios, Witold Pedrycz, Data Mining: A Knowledge Discovery Approach, Springer, 2007.

[5] Howard Robert Turtle, “Inference networks for document retrieval” Doctoral Dissertations Available from Proquest. AAI9120950, 1991.

[6] D Metzler and W.B. Croft, “Linear feature based models for information retrieval”, Inf. Retr. 16, pp. 1-23, 2007.

[7] https://www.kth.se/social/upload/507d1d3af276540519000002/Moore%27s%20law. pdf

[8] P. Switzer, “Vector Images in Document Retrieval”, Harvard University, ISR - 4, 1963.

[9] G. Salton, Automatic Information Organization and Retrieval, McGraw Hill Text, 1968.

[10] J. J. Rocchio, “Relevance Feedback in Information Retrieval”, Harvard University, ISR -9, 1965.

[11] S. Bjørner and S. C. Ardito, “Online Before the Internet, Part 1: Early Pioneers Tell Their Stories”, Searcher: The Magazine for Database Professionals, vol. 11, no.6, 2003.

[12] K. Spärck Jones, “A statistical interpretation of term specificity and its application in retrieval”, Journal of documentation, vol.28, no.1, pp.11-21, 1972.

[13] G. Salton and C. S. Yang, On the Specification of Term Values in Automatic Indexing, Cornell University, TR73-173, 1973.

[14] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing”, Communications of the ACM, vol.18, no.11, pp.613–620, 1975.

[15] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval”, Information processing &management, vol.24, no.5, pp.513-523, 1988.

[16] S.E. Robertson, “The probability ranking principle in IR”, Journal of documentation, vol.33, no.4, pp.294-304, 1977.

[17] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman,”Indexing by latent semantic analysis”, Journal of the American society for information science, vol.41, no.6, pp.391-407, 1990.

[18] G. Salton, The Smart Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[19] S.E. Robertson and K.S. Jones, “Relevance Weighting of Search Terms”, Journal of the American Society for Information Sciences, 27(3), pp. 129–146, 1976.

[20] N Sager, Natural Language Information Processing: A Computer Grammar of English and its Application. Addison-Wesley, 1981.

[21] G.F. DeJong, “Prediction and Sustantiation: A New Approach to Natural Language Processing”, Cognitive Sciences, 3, pp. 251–273, 1979.

[22] G.F. DeJong, “An Overview of the FRUMP System”, Strategies for Natural Language Processing, pp. 149–176, 1982.

[23] P.M. Andersen, P.J. Hayes, A.K. Heuttner, L.M Schmandt and I.B. Nirenberg, “Automatic Extraction”. In Proc. of the Conference of the Asssociation for Artificial Intelligence, pp. 1089–1093, 1993.

[24] Apache Lucene(2011) http://lucene.apache.org/

[25] Apache Solr (2011) http://lucene.apache.org/solr/

[26] Apache Nutch(2005)http://nutch.apache.org/

[27] Regain (2004) http://regain.sourceforge.net/

[28] Oxyus (2010) http://sourceforge.net/projects/oxyus/

[29] Swish-e (2007)http://swish-e.org/

[30] MG4J (2005) http://mg4j.di.unimi.it/9

[31] M Khabsa, S Carman, S.R. Choudhury and C.L Giles, 2012, “A Framework for Bridging the Gap Between Open Source Search Tools”, In proceedings of the SIGIR Workshop on Open Source Information Retrieval, 2012.

[32] A.K. Barman, J Sarmah and S.K. Sarma, “Development of Assamese Rule based Stemmer using WordNet”, In proceedings of the 10th Global WordNet Conference, pp. 135-139, 2019

[33] A.K. Barman, J Sarmah, S.K. Sarma, “POS Tagging of Assamese Language and performance Analysis CRF++ and fnTBL approaches”, In Proceedings of UKSim 15th International Conference on Computer Modelling and Simulation, pp. 476-479, 2013.

[34] A.K. Barman, J Sarmah, S.K. Sarma, “Automatic Identification of Assamese and Bodo Multiword expressions” In proceedings of ICACCI 2013, pp. 26-30, 2013.

[35] A.K. Barman, J Sarmah, S.K. Sarma, “WordNet based Information Retrieval System for Assamese” In Proceedings of UKSim 15th International Conference on Computer Modelling and Simulation, pp. 480-484, 2013.

AUTHOR PROFILE

Dr. Anup Kumar Barman is currently working as an Assistant Professor in Department of I.T., Central Institute of Technology, Kokrajhar. He has pursued M.Sc and M.Tech in Information Technology from Gauhati University. This year in the month of June he was awarded Ph.D. from G.U. and the title of the thesis was “Monolingual Information Retrieval System for Assamese and Its related Natural Language Processing”.

Dr. Jumi Sarmah is a Ph.D. in Information Technology from Gauhati University. She obtained Visvesvaraya PhD fellowship funded by Govt. of India during her research period. She was also a member of the CLIA project funded by Govt. of India. Her research interest lies on Natural Language Processing, Machine Learning and Word sense disambiguation.

Prof. Shikhar Kr Sarma is currently working in Dept of I.T., G.U. He has a wide range of publications and teaching experience from different organizations. Professor Sarma was the Principal Investigator of the CLIA project and IndoWordNet of the Assamese Wing. He has guided a number of research scholars and actively participates in different social activities. He leads the Assamese UNICODE chapter and has mostly worked for the development of Assamese Language Technology.