Open Source Autonomous Bengali Corpus

Summit Haque
Department of CSE
Shahjalal University of Science & Technology, Sylhet

Md. Abu Shahriar Ratul
IICT Cell, Shahjalal University of Science & Technology, Sylhet

Md. Yousuf Ali Khan
Department of CSE
Shahjalal University of Science & Technology, Sylhet

ABSTRACT

Through Sentiment Analysis System it is possible to know what kind of information is there in a text. For example, one can identify is the text about a particular product, political view, sport, entertainment, education, politics, etc. or not. It is also possible to further categorize text in positive, negative or neutral. So, through proper Sentiment Analysis, the current technology would go to another step. There are so many works on Sentiment Analysis that have been done already in different languages. But due to lack of data, the work on Sentiment Analysis on Bangla Text is very limited. Because word categorization accuracy depends heavily on the size of the text corpus used to derive the inter-word statistics. So, it was planned to develop an automated corpus generation system that traverses the Web collecting text and stores them under the defined category. This flexible scheme can produce very large general-purpose corpora or particular samples of domain-specific text.

General Terms

Sentiment Analysis, Bengali Corpus.

Keywords

Corpus, Autonomous Corpus, Autonomous Bengali Corpus, ZIPF Law.

1. INTRODUCTION

A few years ago corpora with a size of 100 million words considered large enough but with the recent Advancement of computer science and technology corpora contains 1 billion words considered as medium size corpus. Surprisingly it is found out that the biggest Bengali corpus build so far contains nearly 30 million words which are good enough but not nearly as big as British National Corpus or many other foreign corpora out there. So it was decided to build an autonomous Bengali corpus that will surf the Web and collect Bangla words and store them under the defined category. But building something with that magnitude and functionality needs a trained crawler. And for training a crawler like that a categorized Bengali Corpus is needed.

Text categorization (also known as text classification or topic spotting) is the task of automatically sorting a set of documents into categories from a predefined set. This task has several applications, including automated indexing of scientific articles according to predefined thesauri of technical terms, filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogs of Web resources, spam filtering, identification of document genre, authorship attribution, survey coding, and even automated essay grading. Automated text classification is attractive because it frees organizations from the need of manually organizing document bases, which can be too expensive, or simply not feasible given the time constraints of the application or the number of documents involved [1].

So we have to backtrack and started to build a normal categorized Bengali corpus which will help us to build an open-source autonomous Bengali Corpus System.

Corpus is the most essential part of Natural Language Processing (NLP) research as well as a wide range of linguistic studies. Prerequisite of any research related to Language Engineering is a well build Corpora. Corpus can be defined as a collection of machine-readable authentic texts (including transcripts of spoken data) that is sampled to be representative of a particular natural language or language variety though “representativeness” is a fluid concept [2] A well build corpus is vital for working with linguistic phenomena such as lexicography, language variations, historical linguistic, spelling variation, morphological structure and work sense analysis Autonomous Corpus is an automated corpus generation system that traverses the Web collecting text that satisfies pre-specified criteria to gather a huge amount of data to build a large corpus. In 1991 The project for the first Bangla corpus building was initiated and closed in 1995 by the department of electronics (DOE), Govt. of India and was created by the Central Institute of Indian Languages (CIIL) and by then the first electronic corpus. Since then this corpus of three million words has been delivering much of the linguistic data required by the scholars working on Bangla. The book of “Corpus linguistics and Language Technology” by N. S. Dash is a warehouse for corpus related studies with special attention to Bangla, where he has discussed almost every linguistic feature of this language and the study is supported by data from the CIIL corpus. Bharati, Sangal, and Bendre (1998) analyzed frequency distribution, common word comparison between Bangla and other seven Indian languages [3]. 'Prothom-Alo' news corpus has been developed by collecting data from a Bangladeshi daily newspaper, the 'Prothom-Alo', for the year 2005. Although the corpus contains a moderate size of more than 18 million words, the corpus is not representative of the Bengali language. As they cited, Prothom-Alo being a news corpus is biased to some particular editing style while flexible in terms of new word type usage. This corpus may also not be a good source to create a language model. Moreover, the corpus is not available for the research community [4]. SUMono corpus, currently the best and the largest Bengali Corpus having a large-scale collection of representative Bengali texts. The corpus consists of 27,118,025 words and 571,572 unique words in Bengali [4].

There are thousands of corpora in the world, but most of them are created for specific research projects and are not publicly available. Since corpus creation is an activity that takes time and costs money, it is certainly desirable for readers to use such ready-made corpora to carry out their work. Unfortunately, however, this is not always feasible or possible. Now if we focus on Bengali Corpus, there is no open-source Bengali Corpus that matches the size or quality of Standard Corpora like English and many other languages. Though The SUMono corpus has been constructed systematically for the Bengali Language its size is not big compare to like English and many other European and Asian languages. To build a Corpus like BNC which contains around 100 Million Words [5], we have to develop an Autonomous Corpus System in which can use Bengali Web Sources. But currently, there is no Autonomous Corpus System available for the Bengali Language. A Team from Brac University, Bangladesh working on Automatic Bengali Corpus Creation but this is in its very early stage and not publicly available for building a perfect autonomous corpus system we need to train the system with a huge amount of data. So we started crawling and manually categorized them as a Sample Training Data and the amount of Data is around 10 Million words and 0.26 Million unique words [4].

2. BACKGROUND STUDY

2.1 Foreign Languages Corpora

We studied Foreign Language Corpora and based on that information we prepare a comparison table among them below.

Table 1: Foreign Language Corpora Comparison [5]

Corpus Name	Country	Corpus Size(in Words)
British National Corpus	United Kingdom	100 Million
The American National Corpus	United States	11.5 Million
The Polish National Corpus	Poland	30 Million
The Hungarian National Corpus	Hungary	40 Million
The Russian Reference Corpus	Russia	100 Million
The German National Corpus	Germany	100 Million

2.2 Bengali Language Corpora

We studied Bengali Language Corpora and based on that information we prepared a comparison table among them below.

Table 2: Bengali Language Corpora Comparison [5]

Corpus Name	Country	Corpus Size(in Words)
SUMono	Bangladesh	27,118,025
Prothom-Alo	Bangladesh	18,100,378
CIIL	India	3,004,573

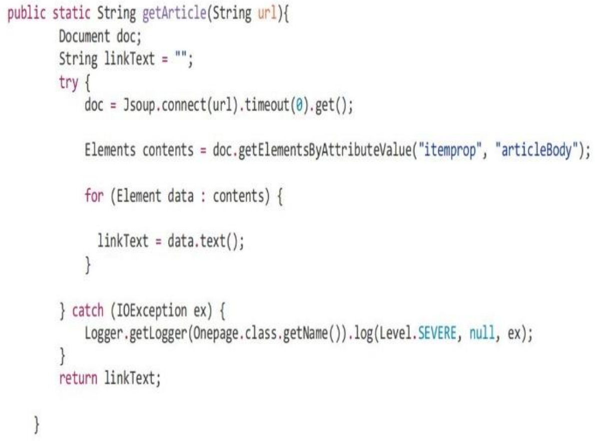
After studying the above Dataset Amount it is clearly understood that Bengali Corpora is in its early stage comparing to other foreign languages Different kinds of projects have been carried out in order to exploit the language data that populate the web. Some of them focused on the direct exploitation of the Internet through search engine techniques (e.g. WebCorp, Renouf et al., 2007). Others were interested in massive language collections (with an almost absence of further control and processing of data) for strict computational purposes (e.g.Clarke et al., 2002, 53 billion words) [6].

So we want to build a corpus with the absence of further control and processing of data to build a huge corpus because there is no such Bengali Corpus (Even without Processing, Corpus Size is So much smaller compared to other Foreign Languages)

3. METHODOLOGY

3.1 Link Collection

We developed a Java Program to crawl the ‘Prothom-Alo’ to build our Corpus. We divided our works into some modules to make the work organized. Our First Step is to Collect All the article links of ‘Prothom-Alo’ based on our selected categories. After crawling the links we stored these on Database.

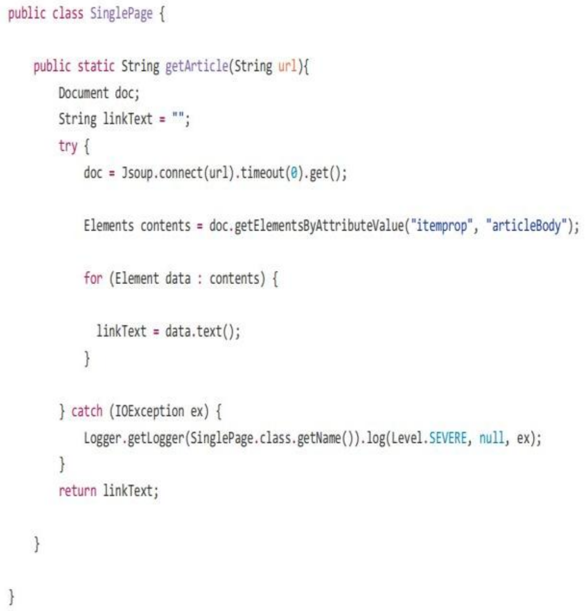


```
public static String getArticle(String url){
        Document doc;
        String linkText = "";
        try {
            doc = Jsoup.connect(url).timeout(0).get();

            Elements contents = doc.getElementsByAttributeValue("itemprop", "articleBody");

            for (Element data : contents) {

              linkText = data.text();
            }

        } catch (IOException ex) {
            Logger.getLogger(Onepage.class.getName()).log(Level.SEVERE, null, ex);
        }
        return linkText;

    }
```

Fig 1: Link Collection Code Snippet

3.2 Text Collection

In the second module, we retrieve the links from the database and crawl the whole article and stored them as text files for further processing.

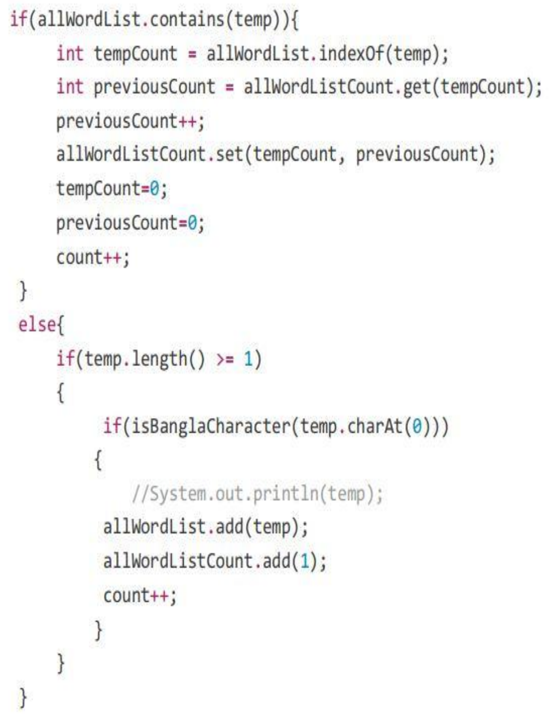


```
public class SinglePage {

    public static String getArticle(String url){
        Document doc;
        String linkText = "";
        try {
            doc = Jsoup.connect(url).timeout(0).get();

            Elements contents = doc.getElementsByAttributeValue("itemprop", "articleBody");

            for (Element data : contents) {

              linkText = data.text();
            }

        } catch (IOException ex) {
            Logger.getLogger(SinglePage.class.getName()).log(Level.SEVERE, null, ex);
        }
        return linkText;

    }

}
```

Fig 2: Text Collection Code Snippet

3.3 Word Detection and Count

In the third module, we retrieve the whole text from text files and then filter the text for English words, numbers, and symbols. After that, we tokenize the dataset using delimiters and stored each work along with frequency in the hash map.

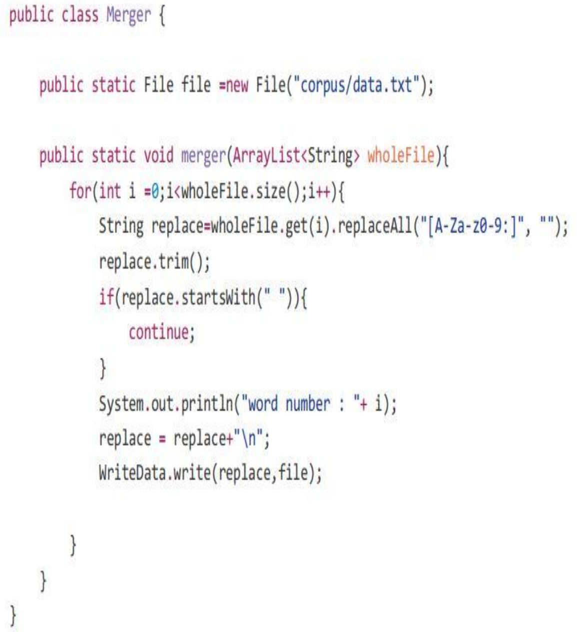


```
if(allWordList.contains(temp)){
     int tempCount = allWordList.indexOf(temp);
     int previousCount = allWordListCount.get(tempCount);
     previousCount++;
     allWordListCount.set(tempCount, previousCount);
     tempCount=0;
     previousCount=0;
     count++;
 }
 else{
     if(temp.length() >= 1)
     {
          if(isBanglaCharacter(temp.charAt(0)))
         {
            //System.out.println(temp);
          allWordList.add(temp);
          allWordListCount.add(1);
          count++;
         }
     }
 }
```

Fig 3: Word Detection and Count Code Snippet

3.4 Merging and Building the Corpus

In this module, we merge the clean dataset from different categories and write them to a single text file.

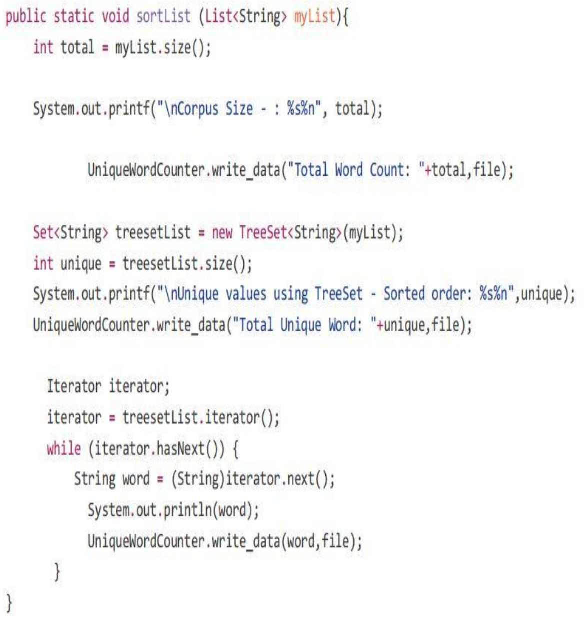


```
public class Merger {

    public static File file =new File("corpus/data.txt");

    public static void merger(ArrayList<String> wholeFile){
        for(int i =0;i<wholeFile.size();i++){
            String replace=wholeFile.get(i).replaceAll("[A-Za-z0-9:]", "");
            replace.trim();
            if(replace.startsWith(" ")){
                continue;
            }
            System.out.println("word number : "+ i);
            replace = replace+"\n";
            WriteData.write(replace,file);


        }
    }
}
```

Fig 4: Merging and Building the Corpus Code Snippet

3.5 Unique Words and Statistics

In this module, we count the unique words from the previous dataset and generate statistical reports.



```
public static void sortList (List<String> myList){
    int total = myList.size();

    System.out.printf("\nCorpus Size - : %s%n", total);

          UniqueWordCounter.write_data("Total Word Count: "+total,file);

    Set<String> treesetList = new TreeSet<String>(myList);
    int unique = treesetList.size();
    System.out.printf("\nUnique values using TreeSet - Sorted order: %s%n",unique);
    UniqueWordCounter.write_data("Total Unique Word: "+unique,file);

     Iterator iterator;
     iterator = treesetList.iterator();
     while (iterator.hasNext()) {
         String word = (String)iterator.next();
           System.out.println(word);
           UniqueWordCounter.write_data(word,file);
      }
}
```

Fig 5: Unique Words and Statistics Code Snippet

4. RESULT ANALYSIS AND DISCUSSION

After crawl the Newspaper 'Prothom-Alo', we prepared a dataset (link: https://scdnlab.com/corpus/), then we do some analysis on the data. The representation of Statistical Analysis given below.

Table 3: Summary of Dataset

Category	No. of Articles	Total Words		Number of Distinct Words
		Number	%	
Accident	1680	250490	2.5	19766
Art	981	531784	5.32	62258
Crime	11739	565358	5.65	32542
Economics	2801	714190	7.14	37192
Education	3761	928146	9.28	56106
Entertainment	3521	652038	6.52	49319
Environment	750	248380	2.48	26893
International	4001	698269	6.98	46354
Opinion	5531	3258664	32.61	123846
Politics	6434	1713761	17.15	57439
Science & Tech	1881	431002	4.31	33749
Sports	5843	471333	4.71	33803
Total	48923	9992082	100	263195

Table 4: The top 10 Frequent Words in Accident Category

Word	Frequency	%	Word	Frequency	%
ও	3373	1.34	দিকে	2015	0.8
থেকে	2399	0.95	উপজেলার	1786	0.71
একটি	2344	0.93	গতকাল	1755	0.7

এ	2316	0.92	আহত	1619	0.64
নিহত	2090	0.83	করে	1555	0.62

Table 5: The top 10 Frequent Words in Art Category

Word	Frequency	%	Word	Frequency	%
না	6512	1.22	আমার	3150	0.59
করে	4765	0.89	থেকে	3129	0.58
ও	3987	0.74	আমি	2965	0.55
আর	3473	0.65	তার	2877	0.54
এই	3413	0.64	এ	2673	0.5

Table 6: The top 10 Frequent Words in Crime Category

Word	Frequency	%	Word	Frequency	%
ও	8126	1.43	পুলিশ	4296	0.75
করে	5988	1.05	বলেন	3487	0.61
এ	5799	1.02	হয়	3342	0.59
থেকে	5545	0.98	একটি	3273	0.57
করা	5292	0.93	গতকাল	3131	0.55

Table 7: The top 10 Frequent Words in Economics Category

Word	Frequency	%	Word	Frequency	%
ও	9769	1.36	এই	4247	0.59
এ	7439	1.04	না	4044	0.56
থেকে	5454	0.76	হাজার	3983	0.55
করা	4508	0.63	টাকা	3938	0.55
করে	4248	0.59	হবে	3844	0.53

Table 8: The top 10 Frequent Words in Education Category

Word	Frequency	%	Word	Frequency	%
ও	16607	1.78	হয়	5146	0.55
উত্তর	15249	1.64	করা	4326	0.46
করে	7201	0.77	এ	4283	0.46
থেকে	5892	0.63	কী	4217	0.45
হবে	5530	0.59	প্রশ্ন	4066	0.43

Table 9: The top 10 Frequent Words in Entertainment Category

Word	Frequency	%	Word	Frequency	%
ও	6683	1.02	তিনি	3425	0.52
এ	4298	0.65	তার	3410	0.52
না	3911	0.59	সঙ্গে	3391	0.52
এই	3859	0.59	করে	3177	0.48
থেকে	3437	0.52	আর	3083	0.47

Table 10: The top 10 Frequent Words in Environment Category

Word	Frequency	%	Word	Frequency	%
ও	4333	1.74	না	1518	0.61
থেকে	2316	0.93	করা	1389	0.55
এ	1861	0.74	এই	1249	0.5
করে	1752	0.7	হয়েছে	1136	0.45
বলেন	1558	0.62	পানি	1058	0.42

Table 11: The top 10 Frequent Words in International Category

Word	Frequency	%	Word	Frequency	%
ও	7897	1.13	এই	4114	0.58
এ	5828	0.83	তিনি	3940	0.56
করে	5594	0.8	বলেন	3805	0.54
করা	4526	0.64	এক	3707	0.53
থেকে	4452	0.63	না	3610	0.51

Table 12: The top 10 Frequent Words in Opinion Category

Word	Frequency	%	Word	Frequency	%
ও	43157	1.32	এ	19991	0.61
না	38926	1.19	থেকে	18497	0.56
করে	26805	0.82	হবে	18406	0.56
এই	22077	0.67	করা	17805	0.54
যে	20166	0.61	এবং	15810	0.48

Table 13: The top 10 Frequent Words in Politics Category

Word	Frequency	%	Word	Frequency	%
ও	26526	1.54	করা	11603	0.67
বলেন	19620	1.14	তিনি	11013	0.64

এ	18038	1.05	হবে	10639	0.62
না	13599	0.79	থেকে	9827	0.57
করে	13500	0.78	করেন	7624	0.44

Table 14: The top 10 Frequent Words in Science & Tech Category

Word	Frequency	%	Word	Frequency	%
ও	5588	1.29	থেকে	2948	0.68
এ	4506	1.04	হবে	2304	0.53
করে	3955	0.91	এবং	2302	0.53
এই	3728	0.86	জন্য	2250	0.52
করা	3001	0.69	একটি	2062	0.47

Table 15: The top 10 Frequent Words in Sports Category

Word	Frequency	%	Word	Frequency	%
না	4481	0.95	রান	2202	0.46
এই	3151	0.66	সঙ্গে	2148	0.45
ও	2957	0.62	কিন্তু	2039	0.43
করে	2944	0.62	আর	2028	0.43
তবে	2227	0.47	থেকে	1951	0.41

4.1 Zipf's Distribution

Zipf's law is useful as a rough description of the frequency distribution of words in human languages [4].

Set against Zipf's law, frequency distribution in an actual dataset is also a reasonable way to gauge data sparseness and can provide evidence of imbalance in a sample. Zipf's law draws a relationship between the frequency of a word f and its position in the list, known as its rank r. The law states that: r.f = c, where r is the rank of a word, f is the frequency of occurrence of the word, and c is a constant that depends on the text being analyzed. [4]

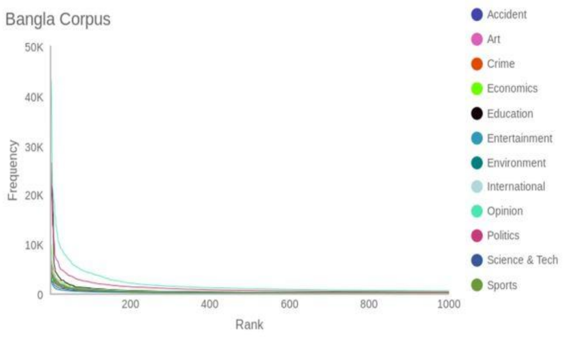


Fig 6: Zipf's Curve for All Twelve Categories

According to Zipf's law, graphs should be a straight line with slope but our graphs in not straight enough. So we can say our corpus is not balanced yet but we hope in the future it will be.

5. CONCLUSION

Sentiment Analysis is a trending topic of NLP. So, we hope our published dataset would help research enthusiasts to work on Bangla Sentiment Analysis. Currently, we are working on a crawler that will crawl the Main article from any given web page. After some research, we find an open-source project "Boilerpipe" [6] which can be very helpful in our corpus building. Categorized Corpus is just the beginning. It will help us to build a trained crawler that will surf the web and collect any Bengali article and create a Categorized Bangla Corpus with at least 1 billion words in it. We wanted to build an autonomous corpus but for the limited resource, we have changed our goal and started building an open-source categorized Bangla corpus. Currently, we successfully crawled nearly 10 million words and 0.2 million unique words from the web and run some analysis on it. As the quality of our corpus is still inferior comparing other Bengali corpora out there, the necessary steps to improve our dataset would be taken shortly.

6. REFERENCES

[1] Sebastiani, F. Text Categorization.

[2] Xiao, R. Corpus Creation.

[3] Majumder, K. M. Y. A., Islam, M. Z., Zaman, N. U., and Khan, M. Analysis of and Observations from a Bangla News Corpus.

[4] Mumin, M. A. A., Shoeb, A. A. M., Selim, M. R., and Iqbal, M. Z. SUMono: A Representative Modern Bengali Corpus

[5] McEnery, A., Xiao, R., Tono, Y. Corpora Survey.

[6] Panunzi, A., Fabbri, M, Moneglia, M., Gregori, L., and Paladini, S. RIDIRE-CPI: an Open Source Crawling and Processing Infrastructure for web Corpora Building.

[7] Sarkar, A. I., Pavel, D. S. H., and Khan, M. Automatic Bangla Corpus Creation.

[8] Lesher, G. W. A Web-Based System for Autonomous Text Corpus Generation.

[9] Oliver, A. Automaticcreation of WordNets from parallel corpora

[10] Pavel Kr´al, P., and Cerisara, C. Automatic Dialog Act Corpuscreation From Web Pages

[11] Jha, M., Andreas, J., Thadani, K., Rosenthal, S., and McKeown, K. Corpus Creation for New Genres: A Crowdsourced Approach to PP Attachment

[12] Maeda, K., Lee, H., Medero, S., Medero, J., Parker, R., and Strassel, S. Annotation Tool Development for Large-Scale Corpus Creation Projects at the Linguistic Data Consortium.

[13] Cieri, C., and Liberman, M. Issues in Corpus Creation and Distribution: The Evolution of the Linguistic Data Consortium

[14] Pavel, D. S. H., Sarkar, A. I., and Khan, M. A Proposed Automated Extraction Procedure Of Bangla Text For Corpus Creation In Unicode