Infrastructure for Bangla Information Retrieval in the Context of ICT for Development

Nafid Haque, M. Hammad Ali, Matin Saad Abdullah, Mumit Khan
BRAC University, Bangladesh
nafid99@yahoo.com, hammad2099@yahoo.com, mabdullah@bracuniversity.net, mumit@bracuniversity.ac.bd

Abstract

In this paper, we talk about developing a search engine and information retrieval system for Bangla. Current work done in this area assumes the use of a particular type of encoding or the availability of particular facilities for the user. We wanted to come up with an implementation that did not require any special features or optimizations in the user end, and would perform just as well in all situations. For this purpose, we picked two case studies to work on in our effort to finding a suitable solution to the problem. While working on these cases, we encountered several problems and had to find our way around these problems. We had to pick and choose from a set of software packages for the one that would best serve our needs. We also had to take into consideration user convenience in using our system, for which we had to keep in mind the diverse demographics of people that might have need for such a system. Finally, we came up with the system, with all the desired features. Some possible future developments also came into mind in the course of our work, which are also mentioned in this paper.

1. Introduction

Over the last decade or so, there has been a huge increase in the use of computers for storing information and ensuring access to this information. Computers have become our preferred method of interaction, work and most importantly, storage of data. The small lifetime of paper documents, and the difficulty of extracting important information from such documents has made it necessary for important information to be stored in digital form. As the amount of data stored electronically increases day by day, so does the need for efficiently searching through this vast collection of data. A huge body of data will serve us no purpose if at times of need we cannot find the desired information without having to sequentially browse through all of it. Not so long ago, almost all computer data was exclusively in English. In recent years, though, there has been a rise in the use of other languages with computers. Since the learning of the English language proved to be a bottleneck in the process of familiarizing people with a computerized system, there has been a greater frequency of using one's native language with computers. Bangla is one such language. However, the plethora of text corpus in Bangla is not represented accurately if we only take into consideration the number of websites being presented in Bangla. Nevertheless, newspapers and magazines are being increasingly attentive towards creating and maintaining their own websites. From that there is an automatic rise in the need for a search engine that can take a Bangla query string, consult the indexed database and produce results. This is particularly true for newspaper websites, which have extremely dynamic content but at the same time also need to ensure easy access to their archives. Apart from the Bangla documents available over the Internet there are many other documents in Bangla that are stored locally for certain purposes and are used to get the information required at the time of need. The most common example of a considerable amount of Bangla data residing in local machines would be the data repository that certain organizations like D.Net [8] want to make available to people in the rural areas of the country. Ensuring this access to information is absolutely integral to the development of a country like Bangladesh. People in the rural areas are often in need of information regarding issues such as the legal system of Bangladesh, or the symptoms of certain diseases, or the best practices regarding the harvest of crops. All of these data can be made available in portable media that can be sent out to information kiosks in remote parts of the country. With this comes the need for efficiently searching through all of this data, based on a set of relevant keywords. All of this data would not be of any use if certain important information could only be retrieved through systematically going through the entire *corpus*. The time and manpower required for such an operation makes the solution prohibitively expensive and ineffective. Towards this end, what we are trying is not to coin a revolutionary new technology, something that the world has never seen before. We are more interested in *harnessing available technology for the sake of national development.* A lot has been done to ensure human rights in a diverse number of sectors, ranging from education and medical facilities to the rights of practicing one's own religion and philosophy in life. However, little has been done to ensure the rights of access to information for people from all walks of life. This is the era of information and communications technology. Just like before this there have been upsurges in the fields of industries, electricity or even the advent of computers as a tool, this is the era of the upsurge in Information and Communications Technology for Development. In today's world, the most influential men are those who have the most convenient access to the largest body of information. As stated in the WSIS Geneva 2003 Declaration, "our common desire and commitment to build a people-centered, inclusive and development-oriented Information Society, where everyone can create, access, utilize and share information and knowledge, enabling individuals, communities and peoples to achieve their full potential in promoting their sustainable development and improving their quality of life, premised on the purposes and principles of the Charter of the United Nations and respecting fully and upholding the Universal Declaration of Human Rights" [15].

Computers today have become a common household tool in developed countries. However, the technology made possible by the use of computers is not equally accessible in all areas of a developing country like Bangladesh. Digital divide has become a matter of great concern. In many countries, the situation is such that in the cities people are enjoying the facilities provided by the latest and the best in communications infrastructure, whereas in other parts of the country people do not even have telephone facilities, let alone access to the internet. For instance, in a developing country like Bangladesh, digital divide is a very serious problem. The Internet is a novel concept in most parts of Bangladesh. Ensuring Internet access all over the country is a faraway dream, something that might yet take decades. Even in the areas that do have internet facilities, the quality of service varies widely. What we want is an alternative, one that can be achieved in a much shorter time but perform just as well. So *in this paper we propose an infrastructure to extract information from a collection of Bangla text with all the modern features of an online search engine*. The proposed solution will be able to extract information from any online sites as well as data that are stored locally on any portable media, and will not have any dependencies that would make it impractical for use in remote areas of a developing country like Bangladesh.

2. Background study

Extracting information from a collection of data is not a new concept. Many proprietary and open-source search engines are capable of this task. The search engines that support Unicode can easily be used for searching for a query string in any language supported by Unicode. One of the leading online search engines today is Google which has recently come up with an application called Google Desktop [1]. Google Desktop indexes the data stored on a local machine and allows an interface to search through the data with all the features of its online search engine. There are many online search engines other than Google, like Yahoo [2] and MSN [3]. However, since these leading search engines are all proprietary products of their respective developers, they cannot be customized to meet specific user needs. Apart from these proprietary search engines there are many open-source engines freely available that can be customized to meet specific user needs. Lucene [4] is one of the most popular and mature open-source search engines. Nutch [5] is a useful environment built upon Lucene that gives the user the full advantage of a search engine. Vicaya [6] is a search engine built on Nutch and Lucene. Vicaya can itself reside on any portable media like a CD and search through its contents without any prior installation on the machine. Our paper proposes a system similar to Vicaya but with additional features to make searching in Bangla easier for the user.

3. Methodology

The reason behind focusing not just on search engines for the web but also on an Information Retrieval system for the local machine stemmed from the fact that there is a greater demand for searching Bangla text from a local machine or portable media than from the Internet. This is due to the lack of availability of Internet facilities in remote areas of the country. *In remote areas of the country people are in need of specific information. Keeping this in mind many Government and Non-government organizations have taken initiatives to setup information centers where a person can come for their required information.* The organizations working with this concept wish to setup these information centers as close as possible to those who need the information. However, Internet facilities cannot be provided to these centers. Thus arises the need for providing all the information in some portable media like CD/DVD ROM. Obviously, then comes the need to search through the corpus at regular intervals. At present, all the available information retrieval/search facilities from a local media are based on the concept of raw string matching – probably the simplest and most inefficient method of searching. The problem with string matching is that it finds all and only the words that are spelled exactly like the query word. That is, it does not allow for the concept of fuzzy searches, words that are spelled very similarly, or words that sound phonetically similar and are very often confused for each other. Another limitation, although not very common, could be the fact that such searching techniques do not use the power of reverse indexing, something that has become very common in the arena of web search engines. This means that each time a query word is typed in, the entire corpus is searched all over again to find each and every instance of the word. As can be readily seen, this can be a great obstacle towards efficient retrieval of data.

So, we needed a state-of-the-art implementation that would work for the Internet as well as for the local machine. Crawling through the data at regular intervals would be a must for websites with contents that change on a regular basis. Thus the index file should be updated along with the website so that the new data can be searched too. For data repositories in a local machine, crawling need only be a one-time thing. We can collect all the data, index it and provide the indexed database in the same media along with the actual collection of data. For these two purposes, we chose two entities as a case study: Prothom Alo as the website and projects managed by an organization called D.Net for the local machine information retrieval system.

Case 1: We chose Prothom Alo [7] since it is the largest circulating Bangla newspaper at present and regularly manages a website. Though being the largest online daily, Prothom Alo was found to be using a proprietary encoding format rather than Unicode. Thus there is a need for a system that would identify the type of encoding being used and if the encoding is not Unicode, then the system will convert it to Unicode (text only). The crawler can then create the index database of the available content. Also we found that the main subscribers/users of the online version of the newspaper are the Non-Resident Bangladeshi (NRB) citizens. These people mostly prefer writing Bangla using the English alphabets. Thus a mechanism allowing users to write their Bangla search query in English would be an added advantage in a search engine package.

Case 2: For the part concerning information retrieval from a local machine, we chose the D.Net [8] projects Pallitathya [9] and Abolombon [10]. The objective of these projects is to ensure access to information to people in remote areas of the country, where Internet is not an option. Details about these projects are given below:

Abolombon is a project geared towards creating awareness among rural people about governance and human rights issues. The project has several dimensions: lack of awareness about their rights, lack of awareness related to the role and obligations of government institutions, lack of availability of information related to legal support, inadequate legal references for legal aid, among others [10]. The project has developed substantive digital legal content in simple Bangla. This information has also been made available through a web portal for open access to target people [10]. The legal contents are also available off line through CDs in project areas. D.Net delivers the legal contents through rural information centers in remote areas. A D.Net employee trained with the use of computers acts as the intermediary and searches for precise data from these CDs for people to come to these centers looking for help and advice. This intermediary would be the end user of our system, the person typing in the query words, obtaining the results and making it useful for the person in need of it.

Thus, with the vision of building a fully-functional Bangla text search engine with all modern features, we have chosen Lucene and Nutch to be the base of our work [4, 5]. Our aim is to customize the open-source search engine Nutch so that it can search for Bangla text from anywhere, be it portable media or the Internet. The implementation would also be able to do this regardless of the encoding method used. The system we are proposing will have three different parts integrated into a common package. The three parts can be referred to as content extraction, indexing and user-friendly input methods. The following sections will highlight each individual part of the package.

3.1. Content extraction

From case 1 we have found that even the largest daily newspaper available online is not using Unicode as their encoding format. As the search engine only supports Unicode, there is a need to convert any other encoding format to Unicode. We have used a simple Java program that crawls through a website and tries to identify the encoding being used by analyzing the Meta Tags, Font types etc. After identifying the type of encoding, the extractor program then converts only the available text into its corresponding Unicode format leaving out any graphics. Thus, given an URL, it would try to find out the encoding used for the site. If it is using Unicode then a crawl can be performed directly. However, if the site is using some other encoding format, then it tries to find out more about the type of encoding used through a set of heuristics developed from stochastic modeling. Once the format is known, it can then use this knowledge to extract all the contents and convert them to its equivalent Unicode encoding. Nutch can then simply crawl through this set of files and create the index. Searching can then proceed on the indexed Unicode database.

3.2. Indexing

Initially, we focused on the search engine API Lucene. Lucene is an open source project that provides the basic functionalities of a search engine without providing the user interface [4]. It is written in Java and suitable for full-text search, especially in cross-platform scenarios. However, since Lucene is just an API it does not provide the framework for user input. So we moved onto Nutch, a package built upon Lucene. In addition to all the functionalities of Lucene, Nutch also provides an easy-to-use interface for user interaction [5]. Since it is written in Java, we would also have Unicode support. Specifically designed to crawl through Unicode text data over the Internet, Nutch was our best choice for our case study on the Prothom Alo website. Of course, to exactly meet our needs we had to make certain changes to the default configurations for Nutch. Following we give a quick overview of what changes we had to make based on our needs.

First of all, we had to create a file containing the set of URLs that we wanted Nutch to crawl. In our case, this was the URL to the Prothom Alo site. We kept the set of websites to be crawled limited to just the one due to bandwidth limitations, and also because we were still at a prototype stage and testing on one domain would be enough to gauge how our solution is performing. Following this step, we had to modify some of the parameters in the Nutch configuration files according to the site that we wanted to crawl, the settings of the Internet connection being used and other issues such as the depth to which we wanted the crawl to run. Once all this has been done, we can just run the crawl and the database would be created. Then this database could be used to search through the contents of the site. More details on how these configurations are to be tweaked can be found from the Nutch tutorial [12] and the Nutch Wiki [11]. For the searching, we would also need to have a Tomcat server running on the machine [13]. Some changes had to be made to ensure that the Tomcat server could support UTF-8 characters [14], which it does not by default. With all of these ready, we could crawl any site, create the database and then search through that database. So we essentially had an information retrieval system for the web, for the Bangla language. If a Bangla site uses Unicode, we could just crawl through it and get the database ready, limited only by the available bandwidth. If the site does not use Unicode, then we would have the added step of using the content extractor to grab the text content, converting it to Unicode and then crawling through that corpus. Nevertheless, the process of crawling and then searching through the database remains the same.

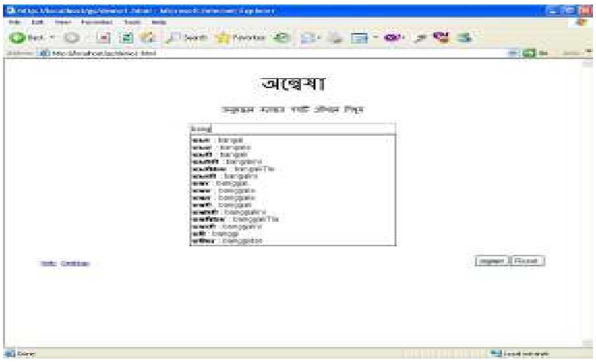

Figure 1: Prototype interface for our system

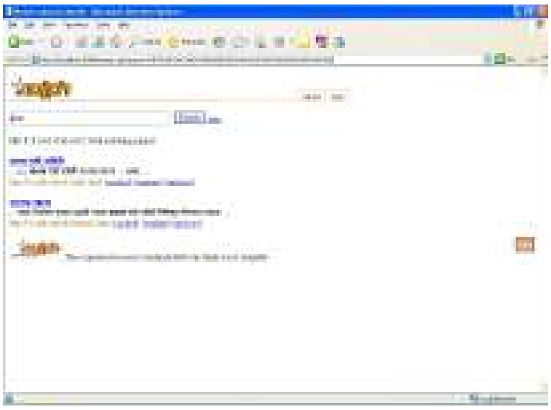

Figure 2: Search results being displayed

3.3. User-friendly input methods

Taking the search query as an input needs a good consideration to make the entire project a success. Though it has been quiet a long time that Bangla is being used in daily computing, yet there is lack of a national standard keyboard layout. A wide range of Bangla keyboard layouts are available and thus the users of Bangla keyboards are not focused to a single keyboard layout. Also from recent trends it can be seen that people prefer to write Bangla phonetically using the English alphabets. Keeping all this in mind we have designed our system so that it can be of help for users regardless of the keyboard layout they use. As the whole system is based on Unicode, users can use any Unicode supported Bangla keyboard to enter their search string in to the system. In case users do not have a Unicode supported Bangla keyboard, the system will provide the user an option to enter their search strings in a transliterated form (writing Bangla using English alphabets.) A JavaScript code has been developed and integrated with the system so that the user can enter their Bangla text in a transliterated form. Here the user enters the Bangla query string in English alphabets and the JavaScript code converts that to its Bangla equivalent. The script code is also capable of suggesting the user with near matches of their input words. It can even suggest words from the index database created by Nutch so that the user can further refine his query string for the nearest match. This will be a great help in a situation where the user does not know the exact spelling of the word, a common scenario in the case of searching for names of individuals, among other things.

With the three above stated segments for the proposed system, the entire system can be a very strong tool for information retrieval from a corpus of any size. All the contents need not be on the Internet but can reside on any portable storage medium (e.g. CD/DVD ROM). In case of searching data from a portable media, the contents need to be indexed first and then along with a live version of the system can be put on the storage medium preferred. Then, the storage medium (e.g. CD/DVD ROM) can be carried to any place and the data in it can be retrieved with a query string. In case the system is to be used for searching data from the Internet, the system can be fed with a list of URLs at the beginning. Then the system will crawl all the contents of the given URLs and create its index file. Later the system can be used to search for information within this set of URLs. As the contents to be grabbed from the Internet may change periodically, thus a need for re-indexing may arise. In that case the system needs to be configured so that the given URLs can be crawled at regular intervals.

4. Future work

At present, the content extractor that we are using only works for text content. In future, we would like to hone this implementation further so that it can work for other, more diverse types of content. This is important, because more often than not websites carry a lot of non-text information that could be just as important to the user. We want the user to be able to see all of this content while using our implementation and not just stay confined to seeing text. So we plan on working further with the content extractor in relation to our system. In addition, we want to enhance the system further, and go from an *information retrieval* system to an *information extraction* system. Such a system would have uses in the field of text summarization, text categorization using the presence of keywords or information extraction from a newspaper article or any such information portal.

5. Conclusion

In this paper, we have tried to suggest a system that would make it possible to search through a large collection of data in Bangla within a feasible amount of time and expense, regardless of the type of encoding being used and the availability of other facilities. Instead of focusing exclusively on searching for the web, we paid equal attention to information retrieval from portable media. We did not try to devise something completely original, something that no one has ever thought of before. *On the contrary, we were trying to suggest ways in which we could capitalize on open-source technology to ensure elimination of digital divide in different parts of Bangladesh. We wanted to suggest an infrastructure that would make sure that access to important information within limited time becomes a right and not a privilege.* The main incentive behind our work was not the intention to come up with a system that no one has thought of or worked on before. Rather, it was trying to contribute something to the mission planned by the United Nations and the International Telecommunications Union in their endeavor to ensure fair access to information for everyone regardless of their geographical location or other background information, as expressed in the Geneva Declaration of 2003 [15]. This is the biggest reason why we did not focus solely on web search, and just try to hone it further. In a country like Bangladesh, the majority does not have access to the Internet. Furthermore, there does not seem to be any probability of ensuring Internet access in all parts of the country even within the next decade or so. So we tried to find a way to do the best we can *to harness the power of Information and Communications Technology for development*, even within the limitations that are imposed on us in the context of this country. We wanted to make sure that what we implemented would be of use not only to the city-dwellers wanting to browse through today's newspaper, but also the peasant in a remote village wanting to know about the best practices regarding his livelihood. We are still far away from the day when people from such diverse backgrounds will achieve equal rights in every walk of life, but we can always hope that our work will ensure equal rights to them at least in their pursuit of information. *Our work may not be groundbreaking in terms of innovation or pioneering in a field, but we trust that it will prove to be very important for the sake of national development. We believe it was our responsibility to use our abilities, and available technology, to do something for the betterment of the nation*. We sincerely hope that the infrastructure suggested in this paper can go far towards this end.

6. Acknowledgement

This work has been supported in part by the PAN Localization Project (www.PANL10n.net) grant from the International Development Research Center, Ottawa, Canada.

7. References

[1] Google Desktop, available online at http://desktop.google.com/about.html

[2] Yahoo! Search, available online at http://search.yahoo.com

[3] MSN Search, available online at http://search.msn.com

[4] E. Hatcher and O. Gospodnetic, "Lucene in Action", April 2006.

[5] The Official Nutch Website – http://www.lucene.apache.org/nutch

[6] Vicaya, available online at http://vicaya.sourceforge.net

[7] Prothom Alo, the largest online daily newspaper in Bangla, available online at www.prothom-alo.net

[8] D.Net – Development Research Network, www.dnet-bangladesh.org

[9] Pallitathya, a research program of D.Net on understanding information needs from a village perceptive, http://www.pallitathya.org/

[10] Abolombon, a program of D.Net designed to improve access to legal information on governance and human rights issues, http://www.abolombon.org/

[11] The Nutch wiki, available online at http://wiki.apache.org/nutch/

[12] The Nutch tutorial for Version 0.7.x, available online at http://www.lucene.apache.org/nutch/tutorial.html

[13] A step by step guideline on how to configure and use Tomcat, available online at http://www.coreservlets.com/Apache-Tomcat-Tutorial/

[14] Weblog on enabling Tomcat to support UTF-8 Encoding - http://rollerweblogger.org/page/roller/20040415

[15] FAQ on the World Summit Information Society at http://www.itu.int/wsis/