The Language Observatory Project (LOP)

Yoshiki Mikami, Pavol Zavarsky
Mohd Zaidi Abd Rozan, Izumi Suzuki
Masayuki Takahashi, Tomohide Maki, Irwan Nizan Ayob
Nagaoka University of Technology
Nagaoka, Niigata, Japan
+81-258-46-6000
mikami@kjs.nagaokaut.ac.jp

Paolo Boldi, Massimo Santini,
Sebastiano Vigna
Università degli Studi di Milano
Dip.to di Scienze dell'Informazione
Via Comelico 39/41, Milano, Italy
+39-0250316305
boldi@acm.org

ABSTRACT
The first part of the paper provides a brief description of the Language Observatory Project (LOP) and highlights the major technical difficulties to be challenged. The latter part gives how we responded to these difficulties by adopting UbiCrawler as a data collecting engine for the project. An interactive collaboration between the two groups is producing quite satisfactory results.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing-indexing methods.

General Terms
Languages, Measurement, Performance, Standardization.

Keywords
Language Digital Divide, Language, Scripts, Character Sets, Web Crawler, Language Identification

1. INTRODUCTION

1.1 Background: Language Digital Divide
The background of the project is a thorough recognition of the unbalanced usage of languages in cyberspace. There are more than 6,000 languages currently spoken around the globe. However, a considerable number of these languages pose serious problems when they are used on the Internet. For example if we visit the site of the Office of the Higher Commissioner for Human Rights of the United Nations, we find more than three hundred different language versions - from Abkhaz to Zulu - of the "Universal Declaration of Human Rights (UDHR)" [1]. But we also find many language translations, especially non-Latin script based languages, are just posted as "gif" files, not in encoded texts. This situation can be described as "digital divide among languages" or just "language digital divide". As a UNESCO resolution mentions [2], "the promotion and use of multilingualism and universal access to cyberspace" is an urgent item in the agenda of global information society.

1.2 Objectives of the Language Observatory
Recognizing such an urgent challenge, the Language Observatory project [3] was planned primarily to provide means for assessing the usage level of each language in cyberspace. More specifically, the project is expected to produce a periodic statistical profile of language / scripts / character code usage in the cyberspace. Once the observatory fully functions, the following questions are to be answered: How many different languages are found in the virtual universe? Which languages are missing from the virtual universe? How many web pages are written in any given language, say Pashto? How many web pages are written using the Tamil script? What kind of character encoding scheme (CES) is employed to encode a given language, say Berber? How quickly is Unicode replacing conventional locally developed encoding schemes on the net? Another goal of the project team is to develop a proposal for technical improvements and policies in order to expand access to the web for a wider variety of languages and scripts.

WWW 2005, May 10-14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

1.3 Technical Challenges
In pursuing the above objectives, technically we have to acquire two major components.

(1) Powerful Web Crawler: LOP needs a crawler which can collect billions of pages at least once a year by using limited network and storage resources. Assuming total pages of web space at 10 billion and average page size at 10KB (text only), then simple calculation shows that we need storage space of 100TB and 25Mbps bandwidth is needed to download the whole web space once every year.

100TBytes / (365*24*3600) sec = 3.2MB/sec = 25Mbps

The maximum bandwidth usage, however, can only be achieved through full parallelism of crawler architecture.

Another technical requirement to crawler is scalability. Whereas our current storage/computing resources are limited, crawling capability should be easily expandable depending upon available resources.

(2) Reliable and Robust Language Identification Algorithm: LOP needs a language identification algorithm capable of identifying language properties of page [language, script and encoding scheme] with high reliability and with wide enough coverage of target languages. Currently available commercial and non-commercial language guessers/identifiers can identify selected, widely used languages only, mostly European languages plus limited Asian languages like Chinese, Korean and Japanese. LOP's target is far beyond that: we hope to identify at least the more than 300 languages of UDHR translation. Our preliminary experiment shows that META-tag information on language/charset cannot always give us a reliable clue. In addition, when taking into account the difficulty of preparing a full set of dictionaries of so many languages, a dictionary-based approach should be avoided. Finally many languages, especially those "falling through the net" are encoded by varieties of *exotic* CESs not found in any of the internationally recognized registries such as ISO-IR or IANA.

Surprisingly, we found 24 different CESs for Hindi and 15 CESs for Tamil being used for web pages in these languages.

2. OUR APPROACH & PRELIMINARY RESULTS

2.1 Selection of crawler: UbiCrawler

After a few months of investigation into various crawlers, we decided to adopt UbiCrawler [4] for the LOP's main page collecting engine. UbiCrawler is a scalable, fully distributed web crawler developed by the joint efforts of the Dipartimento di Scienze dell'Informazione of the Università degli Studi di Milano and the Istituto di Informatica e Telematica of the Italian National Council of Research. The crawler is composed of several agents that autonomously coordinate their behavior in such a way that each of them scans its share of the web. An agent performs its task by running several threads, each dedicated to the visit of a single host at a time (no host is ever visited by two threads at the same time). Assignments of hosts to agents takes into account the mass storage resources and bandwidth available to each agent.

2.2 UbiCrawler's download speed & coverage

Country domains of a group of 57 Islamic countries, members of the Organization of the Islamic Conference (OIC), were chosen for the Observatory's first-round experimental data collection using Ubicrawler. 10 agents ran in parallel and an agent performed its task by running 100 threads each. The crawler downloaded more than 42 million web pages stored in more than 80,000 web servers including more than 6,000 sub-domains. Downloading speed is more than twenty million pages per day. The amount of compressed data obtained from these domains is 348GB. This means that the average bandwidth used for net downloading is well over 16Mbps (actual bandwidth usage is much bigger than this because of various factors). This high performance was achieved by our small computing resource comprising of just ten servers.

Although further assessment is needed, the number of downloaded pages is larger than that indexed by Google or Yahoo for most ccTLDs. UbiCrawler proved the extensiveness of downloading is at a satisfactory level.

2.3 N-gram based Language Identification

As stated above, META-tag information is not always reliable, and we should develop our own language identification engine based on a non-dictionary approach. Our basic response to this is N-gram statistics based identification coupled with use of META-tag information as an optional and secondary clue for identification. Further details of our approach are given in [5].

In this context, LOP's crawler should be easily tuned to adjust to the requirements for which part of web page information should be collected and stored. Here, UbiCrawler's design principle again fits our purpose. One of the main objectives in designing and implementing UbiCrawler was to allow users to configure the crawler behavior to fit their needs. In particular, the store format is highly modular: every page is stored as a sequence of *chunks*, and the user can decide which chunks should be stored (e.g., full-text, compressed text, http headers, etc.). Note that this choice imposes a trade-off between disk space and efficiency; indeed, when a page is downloaded, some form of parsing is necessary, if anything for extracting hyperlinks; so the question is how much information must be extracted and stored for future use. Of course, to keep storage occupancy to its minimum, one might decide to store only the full-text of the page and headers (possibly in compressed form) and then reparse pages each time some information is needed. At the other extreme, one may decide to perform full parsing, and save all information in preprocessed form.

2.4 Tailoring of UbiCrawler for LOP

One point was requested in tailoring UbiCrawler for LOP. This is regarding character sets and encoding schemes. When an HTTP request is issued, the server usually declares the page charset in the *charset* header. Alternatively, the page charset might be found in a META element. If no explicit character set is declared, ISO-8859-1 is assumed (as specified by the HTTP protocol RFC). Before its usage in the Language Observatory Project, UbiCrawler relied on the Java Unicode internal representation for characters in all text processing: the byte stream returned by the server was converted to Unicode characters (using the character encoding information provided), and then processed. Of course, an *ad hoc* implementation of a quick search algorithm for the META tag was necessary, as when searching for META information the character encoding is not known yet, so the search must be performed at byte level.

The needs of the Language Observatory Project, however, made it necessary to change the policy followed by UbiCrawler: the typical pages of interest are often encoded in exotic schemes which are not even supported by the Java standard libraries, or may be even byte-level encoded with a custom font which, in practice, hides completely the actual encoding. For processing such pages correctly, human intervention is sometimes required and, in any case, an exact preservation of the byte stream returned by the server is necessary. As a result, UbiCrawler can now be configured to carefully preserve the exact data received from the server.

2.5 Conclusion: Next Agenda

Now we are working on development of N-gram statistics based language identification engine and are preparing teacher data for 300 languages. Taking into account the chaotic situation of exotic CES usage in several languages, we might have to prepare thousands of teacher data sets corresponding to every combination of language, script and CES.

3. ACKNOWLEDGEMENT

The study was made possible by the financial support of the Japan Science and Technology Agency (JST) under the RISTEX program. We also thank UNESCO for giving official support to the project since its inception.

4. REFERENCES

[1] http://www.unhchr.ch/udhr/navigate/alpha.htm

[2] Recommendation concerning the Promotion and Use of Multilingualism and Universal Access to Cyberspace, 2003.

[3] http://www.language-observatory.org

[4] Paolo Boldi, Bruno Codenotti, Massimo Santini and Sebastiano Vigna. UbiCrawler: A scalable fully distributed web crawler. *Software: Practice & Experience*, 34(8):711-726, 2004.

[5] Izumi Suzuki, Yoshiki Mikami, Ario Ohsato, Yoshihide Chubachi, A language and character set determination method based on N-gram statistics, *ACM Transactions on Asian Language Information Processing*, 1(3): 270-279, 2002