Language and encoding scheme identification of extremely large sets of multilingual text documents

Pavol ZAVARSKY, Yoshiki MIKAMI, Shota WADA
Department of Management and Information Sciences
Nagaoka University of Technology
1603-1 Kamitomioka, 940-2188 Nagaoka, Japan
zavarsky@vos.nagaokaut.ac.jp

Abstract

In the paper we present an outline of our approach to identify languages and encoding schemes in extremely large sets of multi-lingual documents. The large sets we are analyzing in our Language Observatory project [1] are formed by dozens of millions of text documents. In the paper we present an approach which allows us to analyze about 250 documents every second (about 20 million documents/day) on a single Linux machine. Using a multithread processing on a cluster of Linux servers we are able to analyze easily more than 100 million documents/day.

1 Introduction

Identification of written natural languages and character encoding schemes of text documents is not considered to be a difficult problem. It is true if a document is not written in many languages, is long enough, and the number of documents to be analyzed is not extremely large so that the identification of all documents can be finished within an acceptable period of time.

There are two major approaches in written language identification: N-gram and word based approach, see e.g. [3]-[7]. Almost all the existing approaches to language and character encoding scheme identification are *language-neutral*, in the sense that *they can identify any languages that they have been trained on.* Both N-gram and word based tools can be trained with any languages the user likes. When the user knows what languages he wants to distinguish between in his application, he gathers up training material in each of these, trains the tool, and uses the tool. Most of the tools are trained on European and a few Asian languages, because those are the most prevalent and useful in on-line documents, but the tools can be successfully used with many other languages. The important notion to understand is the distinction between the algorithm of the identification process itself, which is usually a kind of N-gram or word based classifier, an implementation of the algorithm, and the byte streams of training data. In the following, we present an outline of our implementation of quad-gram vector distance based language and character encoding identification which allows us to analyze more than 1500 documents every second.

2 Language and character encoding identification in Language Observatory project

Language Observatory project [1] aims to provide, among others, such information like:

- How many written languages are found in the cyberspace?
- How many web pages are written in a given language, e.g. Tamil?
- What kinds of character encoding schemes are employed to encode a given language, e.g. Khmer?

To achieve its goals, the Language Observatory project has to collect and analyze about 10 billion web pages every year. In other words, about 27 million web pages must be collected, parsed, and analyzed every day. We are recently able to collect information we are interested in from about 30 million web documents every day, see also [2]. The languages and character encoding schemes of the web pages form a part of the information we are extracting from the web pages. We have already collected, parsed and analyzed several hundred million of documents and more than 1.5 billion of URL links found on documents on web servers in countries of Organization of Islamic Conference and countries of Asia.

2.1 Efficient access to collected documents

Identification of languages and character encoding schemes on more than 20 million web documents every day requires both an efficient storage of downloaded and parsed web documents and an efficient implementation of the language and encoding scheme identification. In the Language Observatory project we store the snapshots of portions of the web in special store files. A typical size of a store file is 20GB ~ 100GB, depending on the size of portion of the web we are interested in. The special store file contains meta-information, such as HTTP headers, and compressed page content of about 2 million ~ 10 million web pages. The store file is a sequence of byte blocks of page records. Every page record starts with a header, which contains a magic cookie used for synchronization purposes. The original page content, without any character encoding conversion, is present in the valid page record of the store file in a compressed form using a Java deflater. The special file format to store web documents allows a very efficient storage and a fast access to all the stored documents from content analysis application programs, which are not limited to the language and character encoding identification.

2.2 Efficient implementation of language and character encoding scheme identification

We use language and character encoding scheme identification based on quad-gram profiles of languages and encoding schemes and Java packages, classes and methods provided to us by Basis Technology [8]. We are using efficiently object-oriented approach in the language and character identification. Our identification scenario employs reusable language and encoding scheme objects that can be called successively to perform detection on all documents stored in the store files outlined in the previous section. During the identification, a quad-gram profile is build for each valid web page in the store file and a vector distance measure between the input profile and the built-in profile is calculated. The best match has the shortest distance. Multi-profile hash containing all quad-grams of all built-in profiles is constructed at the initialization time. This approach allows adding, modifying and removing built-in profiles of languages and encoding schemes. We were able to verify that the runtime performance is linearly affected by the size of the store file, e.g. by the number of documents and the number of unique quad-grams in the documents. We were also able to verify that the runtime performance is linearly affected by the size of the built-in profiles of supported languages and encoding schemes. We have also tested and safely used language and encoding scheme identification in multi-threaded application run in parallel on ten machines of the Language Observatory Linux cluster. In the multi-threaded language and encoding identification we are able to analyze more than 1500 web pages every second.

3 Conclusion

An efficient storage format that allows a fast access to the stored text documents plays a crucial role in applications, such as our Language Observatory project, requiring millions of documents to be parsed and analyzed every day. The efficient storage format, briefly described in this paper, allows us an efficient implementation of the language and encoding identification based on concepts of object reusability and multi-threaded programming.

In conclusion, Language Observatory project has the aim to help to bridge Digital Divide and welcomes participation and contributions from all interested researchers all around the world.

4 Acknowledgements

The authors wish to thank Steve Cohen and Andrew Paulsen from Basis Technology, USA, and Dr. Massimo Santini, Dr. Paolo Boldi, and Dr. Sebastiano Vigna from the University of Milan, Italy, for their help and suggestions.

The study was made possible by the financial support of the Japan Science and Technology Agency (JST) under the RISTEX program.

References

[1] Language Observatory Project (2004-2006) http://www.language-observatory.org

[2] P.Boldi, S.Vigna, M.Santini: The UbiCrawler Project http://ubi.iit.cnr.it/projects/ubicrawler/

[2] K. R. Beesley. 1988 Language identifier: A computer program for automatic natural-language identification on on-line text. In *Proceedings 29th Annual Conference of the American Translators Association*, pages 47-54.

[3] J. C. Schmitt. 1991 Trigram-based method of language identification, *U.S. Patent* number: 5062143.

[4] J.M. Prager 1999 Linguini: Language identification for multilingual documents. In *32nd Hawaii International Conference on System Sciences*, Hawaii, USA

[5] W.B. Canvar and J.M. Trenkle 1994 *N-gram based text categorization.* In *Symposium on Document Analysis and Information Retrieval*, pages 161-176, University of Nevada, Las Vegas.

[6] G. Grefenstette 1995 Comparing Two Language Identification Schemes. In *3rd International Conference on Statistical Analysis of Textual Data (JADT 95)*, Rome, Italy.

[7] H. El-Shishiny, A.Troussov, DJ McCloskey, M. Takeuchi, A. Nevidomsky, P. Volkov 2004 "Word Fragments Based Arabic Language Identification. In *NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

[8] Rosette Language Identifier, 2004, Basis Technology, http://www.basistech.com.