Text mining in a digital library

Ian H. Witten, Katherine J. Don, Michael Dewsnip
Computer Science, University of Waikato, Hamilton, New Zealand
Email {ihw, kjdon, mdewsnip}@cs.waikato.ac.nz

Valentin Tablan
Computer Science, University of Sheffield, Sheffield, UK
Email V.Tablan@sheffield.ac.uk

1 Introduction

Digital librarians strive to add value to the collections they create and maintain. One way is through selectivity: a carefully chosen set of authoritative documents in a particular topic area is far more useful to those working in the area than a huge, unfocused collection (like the web). Another is by augmenting the collection with high-quality metadata, which supports activities of searching and browsing in a uniform and useful way. A third way, and our topic here, is to enrich the documents by examining their content, extracting information, and using it to enhance the ways they can be located and presented.

Text mining is a burgeoning new field that attempts to glean meaningful information from natural language text. It may be loosely characterized as the process of analyzing text to extract information that is useful for particular purposes. It most commonly targets text whose function is the communication of factual information or opinions, and the motivation for trying to extract information from such text automatically is compelling—even if success is only partial. "Text mining" (sometimes called "text data mining"; Hearst, 1999) defies tight definition but encompasses a wide range of activities: text summarization; document retrieval; document clustering; text categorization; language identification; authorship ascription; identifying phrases, phrase structures, and key-phrases; extracting "entities" such as names, dates, and abbreviations; locating acronyms and their definitions; filling predefined templates with extracted information; and even learning rules from such templates (Witten, in press).

Techniques of text mining have much to offer digital libraries and their users. Here we describe the marriage of a widely used digital library system (Greenstone) with a development environment for text mining (GATE) to enrich the library reader's experience. The work is in progress: one level of integration has been demonstrated and another is planned. The project has been greatly facilitated by the fact that both systems are publicly available under the GNU public license—and, in addition, this means that the benefits gained by leveraging text mining techniques will accrue to all Greenstone users.

2 The Greenstone digital library system

Developed over the last six years, the Greenstone open source digital library software from the New Zealand Digital Library project[1] enjoys considerable success and is widely used (Witten and Bainbridge, 2003). It provides a new way of organizing information and making it available over the Internet. A *collection* of information is typically comprised of several thousand or several million *documents*, and a uniform interface is provided to all documents in a collection. A library may include many different collections, each organized differently—though there is a strong family resemblance in how they are presented. Greenstone's strengths include international language support, multilingual interfaces, and a flexible document importing process that handles different formats—HTML, Word, PDF, PostScript, and E-mail messages, to name but a few. Images, video and audio require accompanying textual metadata.

[1] Available at http://greenstone.org

In Greenstone, the structure, organization, and presentation of any particular collection are determined when the collection is set up. This includes such things as the format or formats of the documents, how they should be displayed on the screen, the sources of metadata, what browsing facilities are to be provided, what full-text search indexes are required, and how the search results should be displayed. Once a collection has been established it is easy to add new documents to it—so long as they have the same format as the existing documents, and the same metadata is provided, in the same way. The structure, organization, and presentation of the collection are recorded as a textual prescription called the "collection configuration file."

Source material is imported into the system by "plug-ins" that cater for different document and metadata formats. Any given collection may have source documents in many different forms. There are plug-ins for plain text files, HTML web pages, Microsoft Office files, PDF and PostScript documents, E-mail, some proprietary formats, and for generic tasks such as recursively traversing directory structures containing such documents. There are metadata plug-ins for XML, MARC records, LaTeX, Refer, and various proprietary formats. New plug-ins can be written to accommodate new file formats. Greenstone builds browsing indexes from metadata using "classifiers," analogous to plug-ins, which create browsable structures of various kinds—alphabetically tabbed lists (of titles, for example), date-selected lists, and hierarchical browsing structures. Like plug-ins, new classifiers can be written for special-purpose browsing structures.

Greenstone already contains some text mining subsystems. One uses simple heuristics to extract acronyms and their definitions from the full text of a collection, and adds this information as metadata (optionally marking up each occurrence of the acronyms too). This is implemented as a plug-in which can be included in any collection simply by including its name in the collection's configuration file; a corresponding classifier is used to display an alphabetic index of acronyms. Another text mining plug-in extracts key-phrases from the documents and adds them as metadata. A third computes a hierarchy of all phrases contained in the text of the documents and allows the user to browse it, optionally in conjunction with a standard thesaurus (Paynter and Witten, 2001). These three examples are implemented independently in a somewhat *ad hoc* manner. The present note describes how Greenstone is being augmented with a general text mining subsystem that offers far greater flexibility.

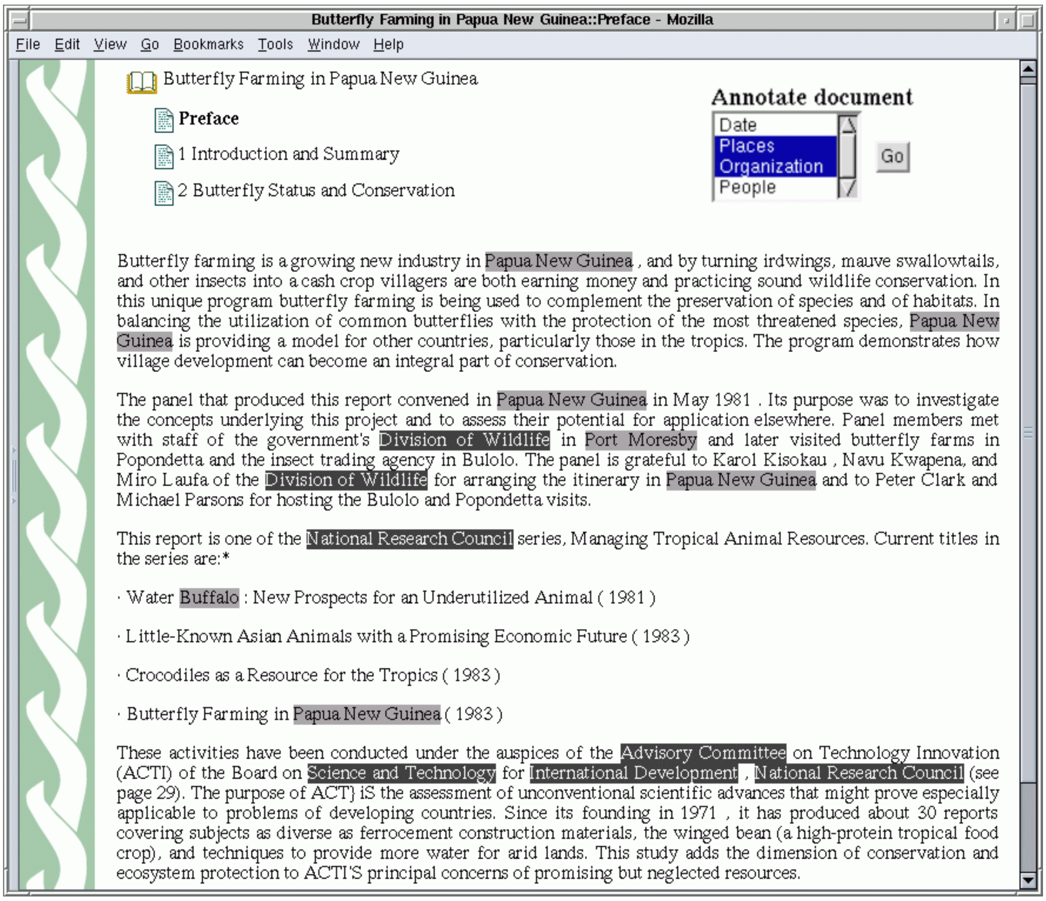
3 The GATE text mining environment

One particular framework and development environment for text mining, called General Architecture for Text Engineering or GATE[2] (Cunningham, 2002), aims to help users develop, evaluate and deploy text mining systems. It provides support not just for standard text mining applications such as information extraction, but also for tasks such as building and annotating corpora, and evaluating the applications. It is currently being used, among other things, for the creation and annotation of a number of corpora in many languages, e.g., the American National Corpus, and a 63 million word corpus of Indic languages (Baker *et al.*, 2002).

GATE includes a wealth of tools for text processing tasks such as tokenization, sentence splitting, part-of-speech tagging, shallow parsing, gazetteer list lookup, information retrieval, and named entity recognition. It also provides access to several types of linguistic resources such as lexicons and ontologies. GATE, like Greenstone, uses Unicode throughout (Tablan *et al.*, 2002) to facilitate application to non-English languages, and it been used to develop applications and corpora in a variety of Slavic, Germanic, Romance, and Indic languages.

GATE is distributed with a lightweight information extraction system, named ANNIE, which detects person and organization names, geographical locations, dates, times and money amounts. It employs a gazetteer with lists of names such as cities, countries or organizations and cue-words such as days of the week. The bulk of the work is performed by a semantic tagger that applies hand-crafted rules written in a language in which patterns can be described and annotations created

[2] Available at http://gate.ac.uk


Butterfly Farming in Papua New Guinea::Preface - Mozilla
File Edit View Go Bookmarks Tools Window Help
Annotate document
Date
Places
Organization
People
Go

Butterfly Farming in Papua New Guinea

Preface

1 Introduction and Summary

2 Butterfly Status and Conservation

Butterfly farming is a growing new industry in Papua New Guinea , and by turning irdwings, mauve swallowtails, and other insects into a cash crop villagers are both earning money and practicing sound wildlife conservation. In this unique program butterfly farming is being used to complement the preservation of species and of habitats. In balancing the utilization of common butterflies with the protection of the most threatened species, Papua New Guinea is providing a model for other countries, particularly those in the tropics. The program demonstrates how village development can become an integral part of conservation.

The panel that produced this report convened in Papua New Guinea in May 1981 . Its purpose was to investigate the concepts underlying this project and to assess their potential for application elsewhere. Panel members met with staff of the government's Division of Wildlife in Port Moresby and later visited butterfly farms in Popondetta and the insect trading agency in Bulolo. The panel is grateful to Karol Kisokau , Navu Kwapena, and Miro Laufa of the Division of Wildlife for arranging the itinerary in Papua New Guinea and to Peter Clark and Michael Parsons for hosting the Bulolo and Popondetta visits.

This report is one of the National Research Council series, Managing Tropical Animal Resources. Current titles in the series are:*

· Water Buffalo : New Prospects for an Underutilized Animal (1981)

· Little-Known Asian Animals with a Promising Economic Future (1983)

· Crocodiles as a Resource for the Tropics (1983)

· Butterfly Farming in Papua New Guinea (1983)

These activities have been conducted under the auspices of the Advisory Committee on Technology Innovation (ACTI) of the Board on Science and Technology for International Development , National Research Council (see page 29). The purpose of ACT} iS the assessment of unconventional scientific advances that might prove especially applicable to problems of developing countries. Since its founding in 1971 , it has produced about 30 reports covering subjects as diverse as ferrocement construction materials, the winged bean (a high-protein tropical food crop), and techniques to provide more water for arid lands. This study adds the dimension of conservation and ecosystem protection to ACTI'S principal concerns of promising but neglected resources.

Figure 1. A Greenstone document with some entities highlighted by ANNIE

as a result. Patterns can be specified by giving a particular text string, or annotations that have previously been created by modules such as the tokenizer, gazetteer, or document format analysis. Also included are modules that recognize relations between entities and detect co-reference.

One application of GATE is a system for extraction of entity names that is capable of processing texts from widely different domains and genres. This has been used to perform recognition and tracking tasks of named, nominal and pronominal entities in several types of text. GATE has also been used to produce formal annotations about important events in a text commentary that accompanies football video program material (Declerck *et al*., 2001).

4 Marrying Greenstone and GATE

Digital libraries present two rather different opportunities for applying text mining. One is at "display time," when particular documents are located and served to the user. The other is at what we call "build time," which is when the digital library collection is created.

Display-time text mining. Digital library documents, once located, can be mined "on the fly" before presenting them to the user. Figure 1 shows a section of a book on *Butterfly Farming in Papua New Guinea* (from the *Humanity Development Library* at http://nzdl.org) that the user is reading. The "Annotate document" menu at the top right calls the display-time text mining feature. When items on this menu are selected, text of the selected type is highlighted in the document display. In this case the user has selected *Places* and *Organizations*, and these are highlighted wherever they occur, in different colors (simulated by white-on-grey and black-on grey in the illustration). For example, “Papua New Guinea” is highlighted as a place, while “National Research Council” is highlighted as an institution. “Buffalo” in “Water Buffalo” has also, erroneously, been highlighted as a place name. Other possible annotation types include dates and personal names. The idea is that highlighting selected items makes it easier for the user to scan the document for particular pieces of information.

The selectable items in Figure 1 were identified and extracted by ANNIE, GATE’s information extraction system. Whenever an annotation type is selected from the menu Greenstone calls the information extraction module dynamically to identify items of that type. This incurs a delay that depends on the document’s size—but the delay is usually short: the system processes text at the average rate of 15 Kb/sec. In order to achieve this the software is loaded when the Greenstone server is executed and remains resident in memory thereafter, to avoid a start-up delay (of about 10 seconds) in which all the text processors are initialized and prepared for use.

For this work we used an experimental version of Greenstone that we are developing, rather than the standard version. In this new architecture a digital library comprises a set of information collections along with a group of independent modular services, such as a querying service (which takes a query and returns a list of document identifiers), a document retrieval service (which takes a document identifier and returns the document text), and a browsing service (which, for example, takes a classification code and returns all the documents with that classification). GATE is encapsulated in a module as just another service. It takes messages with two parameters, the type of annotation and a document, and returns the document with relevant items marked up with XML tags. A style sheet in the Greenstone server module chooses the colors in which the items are displayed.

All Greenstone modules answer “describe-yourself” requests that give details about what parameters they take. In order to generate the menu in Figure 1, this message is sent to the GATE module, which returns a list of the annotations that are available. Apart from this one module, the rest of Greenstone knows nothing of GATE. However, it does know about a general class of services, *enrich*, that take a document and return the same document with some elements marked up. Such services are used for other functions (such as acronym markup in document text, mentioned above).

Build-time text mining. All collections in Greenstone go through a process of “building” in which documents are imported into the system, converted from their original representation into Greenstone’s XML-based native format, and the necessary full-text indexes and databases are constructed. This is often a lengthy process: it can take from a few moments to several hours depending on the size of the collection (Mbytes to Gbytes) and the number of searchable indexes to be constructed. For dynamic collections it will be incremental: a partial building process may occur whenever new documents are added to the collection. (Currently, Greenstone collections are basically static, but incremental operation can be achieved by seamlessly integrating a large main collection with a small auxiliary one which is rapidly re-built whenever documents are added.)

Text mining can be applied at this stage to extract metadata from the documents and enrich documents by marking up appropriate items in the text. This off-line operation has the advantage that information access structures can be built with the extracted metadata. For example, items such as personal, place, and company names could be extracted from the documents and built into browsing structures, and also built into separate searchable indexes—so that users can search for a particular name which is also a common word and have only the pertinent documents returned. It has the drawback that all documents must be mined even if they are destined never to be read, and that storage is required for all extracted metadata and markup—which may be considerable.

In Greenstone, build-time text mining would be implemented as a plug-in. GATE has not yet been integrated in this way because this part of the new Greenstone is still under development. However, we are planning to use precisely the same GATE module as is used at display time—another advantage of the service-oriented, modular architecture.

Comparison. Display-time text mining takes place entirely within the interactive digital library server and applies immediately to all existing collections without the need to re-build them. This makes it ideal for experimentation, to get a quick feel for how it works on existing collections. It also has the potential advantage (although not realized in this application) that its operation can be made dependent on the query and the result set. For example, the result set could be clustered to allow an informative graphical display of the returned documents. The disadvantage of display-time operation is that while the information gleaned can be used to enhance the presentation of documents, it cannot assist in locating them.

Because of the way the GATE module was designed, the ANNIE information extraction system could be replaced by an alternative that was tailored to the particular needs of the collection being processed. This could lead to higher accuracy in the name recognition. Highly customized systems could be used at build time, when execution time is not a concern, while ANNIE—or a streamlined version—could be used at display time in order to improve responsiveness.

5 Conclusions

Text mining can be used to add value to documents in digital library collections; Greenstone already incorporates a few examples. What we have described is how a general text mining environment can be included within a digital library system. This provides a more satisfactory basis for putting future advances in text mining to work immediately for the benefit of digital library users. Our current implementation processes documents at display time only: in future we plan to incorporate text mining into the collection-building process so that the data produced can be used for searching and browsing as well as for document display. It will also be necessary to design ways in which desired text mining operations can be specified in a collection's configuration file.

The linkage of a general digital library system with a general text mining system presents many other possibilities. Tracking entities across documents leads to automatic hyperlinking of co-references. Semantic indexing could be accomplished by annotating texts with ontology classes and allowing semantic searches rather than simple textual queries, providing some of the advantages of the semantic web. Document summaries could be generated automatically to serve as the "snippets" that are presented in lists of search results. Having GATE embedded in a digital library system will also benefit its own users by allowing them to experiment on real text collections in a digital library, rather than with specially-constructed text corpora. This spells good news for digital library users.

References

Baker, P., Hardie, A., McEnery, A., Cunningham, H. and Gaizauskas R. (2002) "EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation." *Proc Language Resources and Evaluation Conference*, pp. 819-825. Gran Canaria, Spain.

Cunningham, H. (2002) "GATE, a General Architecture for Text Engineering." *Computers and the Humanities*, Vol. 36, pp. 223-254.

Declerck, T., Wittenberg, P. and Cunningham, H. (2001) "The automatic generation of formal annotations in a multimedia indexing and searching environment." *Proc of the ACL/EACL Workshop on Human Language Technology and Knowledge Management*, Toulouse, France, pp. 129-136.

Hearst, M.A. (1999) "Untangling text mining." *Proc Annual Meeting of the Association for Computational Linguistics*. University of Maryland, June.

Paynter, G.W. and Witten, I.H. (2001) "A combined phrase and thesaurus browser for large document collections." *Proc European Conference on Digital Libraries*, Darmstadt, Germany, pp. 25-36.

Tablan, V., Ursu, C., Bontcheva, K., Cunningham, H., Maynard, D., Hamza, O., McEnery, A., Baker, P. and Leisher M. (2002) "A Unicode-based environment for creation and use of language resources." *Proc Language Resources and Evaluation Conference*, pp. 66-71. Gran Canaria, Spain.

Witten, I.H. and Bainbridge, D. (2003) *How to build a digital library*. Morgan Kaufmann, San Francisco.

Witten, I.H. (in press) "Text mining." In *Practical Handbook of Internet Computing*, edited by M.P. Singh. CRC Press, Boca Raton, Florida.