Connecting Text Mining and Natural Language Processing in a Humanistic Context

Xin XIANG
John UNSWORTH

Graduate School of Library and Information Science, University of Illinois, Urbana, Champaign1 Introduction

Recent integration of advanced information technology and humanistic research has seen many interesting results that are brand new to traditional humanistic research. In the NORA project this integration was largely exemplified. In an effort to produce software for discovering, visualizing and exploring significant patterns across large collections of full-text humanities resources in digital libraries and collections, NORA project features the powerful D2K data mining toolkit developed by NCSA at University of Illinois, and the creative Tamarind preprocessing package developed by University of Georgia. In NORA project, D2K and Tamarind need to talk to each other, among other relevant components. The idea behind this connection is as follows:

- Make use of existing efforts and try to avoid duplication. Natural language processing, such as part-of-speech tagging, word sense disambiguation, and bilingual dictionary creation, has long been recognized as an important technique for text data mining (Amstrong, 1994). D2K has proved to be an effective and comprehensive data mining toolkit, and Tamarind prepares data gleaned from large-scale full text archives. Getting them work together is an easy and time-saving way of achieving the goal of NORA.
- Separate different tasks according to institution makes the multi-institutional project easier. D2K has been developed and used in several institutions within University of Illinois, and Tamarind was developed in University of Georgia for simplifying primary text analysis tasks. This separation keeps each institution focusing on a relatively independent module that they have the most experience with.
- Prepare information about tokens once and for all. Natural language processing tasks prove time-consuming and computation-intensive. Separating these tasks from data mining part of this project obviates D2K toolkit from performing basic data analysis every time it runs, thus streamlining the whole process.

The problem, however, is that D2K and Tamarind are developed using different programming languages and have different communication mechanisms. To put them together requires reconcilement and restructuring at both sides. This has eventually been achieved in a prototype application, where a collection of Emily Dickinson's poems is classified as either "hot" (erotic) or "not hot" based on the language used (Kirschenbaum, 2006).

As the size of data increases, the problem of scalability emerges. The huge size of many humanistic collections will make unrealistic the solution of storing all the tables in the database. A perfect method to address this problem has not been found, and content presented here demonstrates how we approach the text mining problem in the prototype when the size of collections is not very large.

2 The D2K Toolkit

D2K - Data to Knowledge is a flexible data mining and machine learning system that integrates analytical data mining methods for prediction, discovery, and deviation detection, with information visualization tools (D2K). It provides a graphic-based environment where users with no knowledge in computation and programming can easily bring together software functional modules and make an itinerary, in which a unique data flow and a task are performed. These modules and the entire D2K environment are written in Java for maximum flexibility and portability.

The data mining and machine learning techniques that have been implemented in D2K include association rule, Bayes rule, support vector machine, decision tree, etc. These techniques provide many possibilities of classifying collections available to this project, like hundreds of Emily Dickinson's poems.

Although D2K has the ability of performing basic natural language processing tasks, it is still beneficial to delegate those tasks to a toolkit that is specifically designed to do this, i.e., Tamarind.

3 Gate and Tamarind

Both D2K and Tamarind use Gate as their fundamental natural language processing toolkit. Gate has been in development at the University of Sheffield since 1995 and has been used in a wide variety of research and development projects (Gate).

Tamarind is a text mining preprocessing toolkit built on Gate, analyzing XML-based text collections and putting the results into database tables (Downie 2005). It serves as a bridge between Gate and D2K, and connects them through the use of persistent database. It supports JDBC-based data retrieval, as well as SOAP-based language-independent APIs.

Table 1 shows a typical table in Tamarind database. The "xpath" field contains the location of a token in the TEI document in terms of XPath expression, "doc_id" is the unique ID of the TEI

document, while "t_type_id" is the part-of-speech tag. Based on this table, some statistical characteristics of tokens, like term occurrence (term frequency), co-occurrence and document frequency, could be generated, thereby obviating the data mining toolkit (D2K) from performing the data-preparing task.

xpath	doc_id	pos_id	t_type_id	token_id
/TEI.2[1]/teiHeader[1]/fileDesc[1]/titleStmt[1]/title[1]/Token[1]	1	1	1	1
/TEI.2[1]/teiHeader[1]/fileDesc[1]/titleStmt[1]/title[1]/Token[2]	1	2	2	2

Table 1: A Tamarind Table

After the whole collections is parsed and analyzed, the information related to the position, part of speech and type of each token is stored in a PostgreSQL database for future access. The Tamarind application exposes these information so that D2K as a client can connect and retrieve them through JDBC (Java Database Connectivity) or SOAP (Simple Object Access Protocol).

4 NORA Architecture

Several issues were raised as to how to effectively and efficiently connect physically and institutionally distributed components in the NORA project. For example, should Tamarind expose its data to client through Java API (as a Java JAR file) or SOAP API (through Web service)? Should Tamarind just provide raw data like that in the previous table or something more advanced and complicated like the frequently used TF-IDF value? Is D2K responsible for converting the database table to a data structure more convenient for D2K to handle, like D2K table? How can the user requests be conveyed to D2K in a user-friendly and compact fashion?

Experiments and discussion eventually led to the adoption of JDBC-based data retrieval and SOAP-based Web service for user request delivery. Although SOAP-based Web service providing more advanced and platform-independent API interface is a good choice for delivering Web-based requests, it seems inefficient to transmit large amount of data, like the occurrences of all tokens in the whole collection, through HTTP protocol, especially when the data store and the text mining application do not reside on the same host. This, however, does not exclude the possibility of implementing some not-so-data-intensive APIs, like metadata retrieval, through SOAP in the future.

Table 2 gives sample data pairs pulled out of Tamarind database. It is a list of which token occurs in which document and is generated by a join of several tables in the Tamarind database.

document	token
Seaf709v1-tbh.xml	with
Seaf709v1-tbh.xml	that
Seaf709v1-tbh.xml	of
Seaf709v1-tbh.xml	joseph
Seaf709v1-tbh.xml	in
Seaf709v1-tbh.xml	egypt

Table2 : Data (document-token pairs) from Tamarind Database

After data about tokens is pulled out of the Tamarind database, it is converted to a structure called "D2K table" which is convenient for the D2K toolkit to handle. Actually the D2K table is the restructuring of the token-document pairs taken from the database as a matrix containing the occurrences of each token in each document. Table 3 gives an example. Depending on the collection, it could contain hundreds of rows and thousands of columns.

	with	that	of	joseph	in
Seaf709v1-tbh.xml	0	0	1	1	0
Seaf709v2-tbh.xml	1	0	1	1	0
Seaf709v3-tbh.xml	0	2	1	1	1
Seaf709v4-tbh.xml	0	0	0	2	0
Seaf709v5-tbh.xml	1	0	0	0	1

Table 3: A D2K Table

This D2K table is ready for evaluation by several common machine learning techniques, like naive Bayes and support vector machine. For the Dickinson prototype, naive Bayes algorithm is used and an overall classification accuracy of over 70% is achieved. In the prototype, the D2K toolkit is launched by Infovis, an information visualization toolkit, through Web service.

Figure 1 depicts the control flow of the whole prototype system.

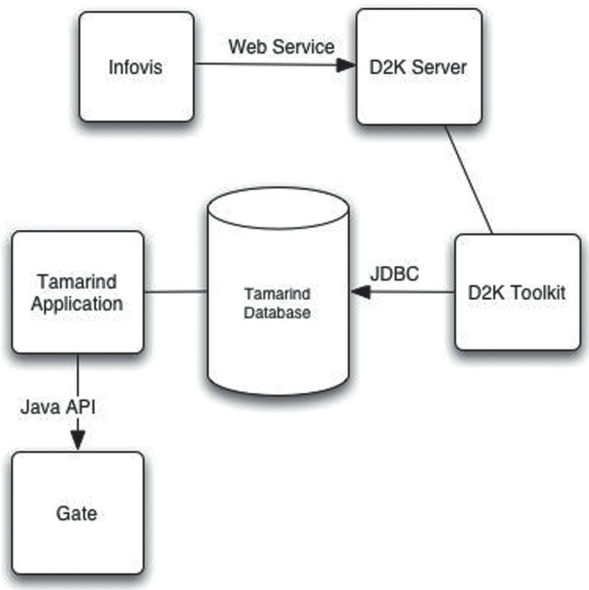

Figure 1. NORA Architecture

References

Armstrong, S. (1994). *Using Large Corpora*. MIT Press.

Kirschenbaum, M. Plaisant, C. Smith, M. Auvil, L. Rose, J. Yu, B. and Clement, T. (2006) *Undiscovered public knowledge: Mining for patterns of erotic language in Emily Dickinson's correspondence with Susan (Gilbert) Dickinson*. ACH/ALLC 2006.

http://alg.ncsa.uiuc.edu/do/tools/d2k.

http://gate.ac.uk.

Downie, S. Unsworth, J. Yu, B. Tcheng, D. Rockwell, G. and Ramsay S. (2005) *A revolutionary approach to humanities computing?: Tools development and the D2K data-mining framework*. ACH/ALLC 2005.