International Journal of Reviews in Computing

IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

TEXT MINING: ADVANCEMENTS, CHALLENGES AND FUTURE DIRECTIONS

[1]MAHESH T R, [2]SURESH M B, [3]M VINAYABABU

[1]Asst Prof., Department ISE, SBMJCE, Bangalore, India

[2]Asst Prof., Department ISE, EWIT, Bangalore, India

[3]Assoc. Prof., Department of MCA, IARE, Hyderabad, India

E-mail: sureshmb@gmail.com , maheshtr.khushi@gmail.com, mvinayababu@gmail.com

ABSTRACT

Text mining, also known as text data mining or knowledge discovery from textual databases, refers to the process of extracting interesting and non-trivial patterns or knowledge from text documents. Regarded by many as the next wave of knowledge discovery, text mining has very high commercial values. Last count reveals that there are more than ten high-tech companies offering products for text mining. Has text mining evolved so rapidly to become a mature field? This article attempts to shed some lights to the question. We first present a text mining framework consisting of two components: *Text refining* that transforms unstructured text documents into an *intermediate form;* and *knowledge distillation* that deduces patterns or knowledge from the *intermediate form*. We then survey the state-of-the-art text mining products/applications and align them based on the text refining and knowledge distillation functions as well as the intermediate form that they adopt. In conclusion, we highlight the upcoming challenges of text mining and the opportunities it offers.

Keywords: *Text mining, data mining, knowledge discovery*

1. INTRODUCTION

Text mining, also known as text data mining [3] or knowledge discovery from textual databases [2], refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents. It can be viewed as an extension of data mining or knowledge discovery from (structured) databases [1,4]. As the most natural form of storing information is *text*, text mining is believed to have a commercial potential higher than that of data mining. In fact, a recent study indicated that 80% of a company's information is contained in text documents. Text mining, however, is also a much more complex task (than data mining) as it involves dealing with text data that are inherently unstructured and fuzzy. Text mining is a multidisciplinary field, involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining.

The amount of textual-based information stored electronically, whether on our own computers or on the Web, is rapidly accumulating. Any desktop or laptop computer can accommodate huge amounts of data due to the advances in hardware storage devices. Accumulating information is easy, finding relevant information on demand can be difficult. Constructing data structures (indices) to facilitate the retrieval of relevant information becomes problematic as the size of collections continue to escalate. Equally important is the ability to extract specific patterns or features to meet particular information needs. In this chapter we discuss novel developments in the design of software for large-scale index creation and algorithms for feature extraction from textual media.

This article presents a general framework for text mining consisting of two components: Text refining that transforms free-form text documents into an intermediate form; and knowledge distillation that deduces patterns or knowledge from the intermediate form. We then use the proposed framework to study and align the state-of-the-art

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

text mining products and applications based on the text refining and knowledge distillation functions as well as the intermediate form that they adopt.

2. A FRAMEWORK OF TEXT MINING

Text mining can be visualized as consisting of two phases: *Text refining* that transforms free-form text documents into a chosen *intermediate form,* and *knowledge distillation* that deduces patterns or knowledge from the intermediate form. Intermediate form (IF) can be *semi-structured* such as the conceptual graph representation, or *structured* such as the relational data representation. Intermediate form can be *document-based* wherein each entity represents a document, or *concept based* wherein each entity represents an object or concept of interests in a specific domain.

Mining a document-based IF deduces patterns and relationship across documents. Document clustering/visualization and categorization are examples of mining from a document-based IF. Mining a concept-based IF derives pattern and relationship across objects or concepts. Data mining operations, such as predictive modeling and associative discovery, fall into this category. A document-based IF can be transformed into a concept-based IF by realigning or extracting the relevant information according to the objects of interests in a specific domain. It follows that document-based IF is usually domain-independent and concept-based IF is domain-dependent.

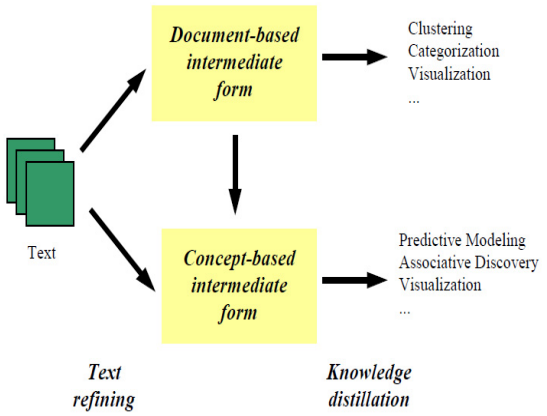

Figure 1. A text mining framework

For example, given a set of news articles, text refining first converts each document into a document-based IF. One can then perform knowledge distillation on the document-based IF for the purpose of organizing the articles, according to their content, for visualization and navigation purposes. For knowledge discovery in a specific domain, the document-based IF of the news articles can be projected onto a concept-based IF depending on the task requirement. For example, one can extract information related to "*company*" from the document-based IF and form a company database. Knowledge distillation can then be performed on the company database (company-based IF) to derive company-related knowledge.

3. ADVANCEMENTS IN TEXT MINING

Software companies develop products that may require megabytes of hard drive space. Without upgrading computers every few years, one cannot download favorite music, movies or play the most recent (popular) computer games. Researchers and scientists involved in data mining and information retrieval are facing the same reality – an enormous amount of storage may be needed to run simulations and store their outputs. In creating the General Text Parser (GTP) with network storage capability, we are trying to address the needs of experts in information retrieval and modeling who deal with large text corpora on a daily basis but are subject to limited storage capabilities. This software allows a user to parse a large collection of documents and create a vector space information retrieval model for subsequent concept-based query processing. GTP utilizes latent semantic indexing (LSI) for its information retrieval (IR) modeling [BB90, BDO95, BDJ99]. The user has the option of storing the model outputs on one of the available Internet Backplane Protocol (IBP) servers or depots [BBF+02, PBB+01] so that disk or memory space is shared over the network.

The underlying vector-space model exploited by the GTP is Latent Semantic Indexing (LSI). LSI is an efficient IR technique that uses statistically derived conceptual indices rather than individual words to encode documents. Specifically, LSI uses the singular value decomposition (SVD) or semi-discrete decomposition (SDD) of the large sparse term-by-document matrix mentioned above to build a conceptual vector space [BB90, BDO95]. A lower-rank approximation to the original term-by-document matrix is used to derive vector encodings for both terms and documents in the same dimensional subspace. The clustering of term or document vectors in this subspace suggests an underlying (latent) semantic structure in the usage of terms within the documents.

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

3.1 Network Storage Stack

The Network Storage Stack has been developed by the Logistical Computing and Internetworking Lab (LoCI) at the University of Tennessee [CwLL03]. As discussed in [BBF+02], the Network Storage Stack is modeled after the Internet Protocol (IP) Stack, and is designed to add storage resources to the Internet in a sharable, scalable manner. Figure below shows the organization of the Network Storage Stack.

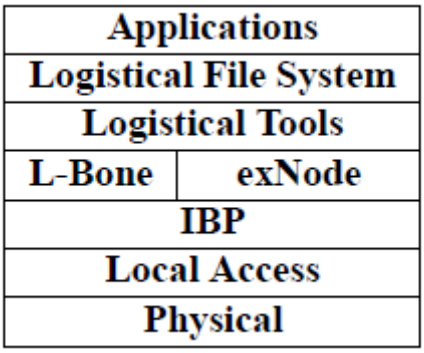

Figure 2. network storage stack

3.1.1. IBP

The Internet Backplane Protocol (IBP) is an essential part of the Network Storage Stack. IBP's purpose is to allow users to share storage resources across networks. Its design echoes the major advantages of Internet Protocol (IP): abstraction of the datagram delivery process, scalability, simple fault detection (faulty datagrams are dropped), and ease of access. These factors allow any participant in an IBP network to use any local storage resource available regardless of who owns it [BBF+02]. Using IP networking to access IBP storage creates a global storage service.There are some limitations that arise from two underlying network problems. The first problem concerns a vulnerability of IP networks to Denial of Use (DoU). The free sharing of communication within a routed IP network leaves every local network open to being overwhelmed by traffic from the wide area network. A second concern lies in that the storage service is based on processor-attached storage, which implies strong semantics: near-perfect reliability and availability. IBP is almost impossible to implement on the scale of the wide area networks [BBF+02]. These issues are resolved as follows:

- IBP storage is time limited. When the time expires the resources can be reused.An IBP allocation can also be refused by a storage facility (depot) if the user's request demands more resources than available.
- IBP is a best effort storage service [BBF+02]. The semantics of IBP storage are weaker than the typical storage service. Since there are so many unpredictable and uncontrollable factors involved, network access to storage may become permanently unavailable (if a depot decides to withdraw from the pool, for example).

IBP storage is managed by depots or servers used by a client to perform storage operations such as Allocate, Load, Store, and Copy. See [Mir03] for details on these and other storage operations.

3.1.2 ExNode

The management of several IBP capabilities can be complicated. The exNode library was created to help the user in this task and to automate most of the work. The exNode data structure is somewhat similar to the Unix inode, but at the same time it is fundamentally different.The exNode makes it possible for the user to chain IBP allocations into a logical entity that resembles a network file [BBM01]. Current IBP allocations have a limit of 2 GB; the exNode though allows the user to chain 2 billion IBP allocations, which equals 4 Exabytes[BBM01].

The exNode consists of two major components: arbitrary metadata and mappings. The exNode library allows the user to create an exNode, attach a mapping to it, store IBP capabilities into the mapping and add metadata to the mapping. The exNode can also be serialized to XML, so that exNodes created on one platform can be recognized on other supported platforms. Each exNode can have multiple copies of the allocation, which provides better fault-tolerance. If a depot becomes unavailable for some reason, the user can still retrieve data from the copies stored on other depots.

3.1.3 L-Bone

The Logistical Backbone (L-Bone) is a resource discovery service that maintains a list of public depots and metadata about those depots [BBF+02,BMP02]. The L-Bone also uses the Network Weather Service (NWS) [WSH99] to monitor throughput between depots. As of March 2003, the L-Bone provides service of over 150 depots on five continents

3.1.4 LoRS

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

The next and final layer of the Network Storage Stack which we briefly mention is the Logistical Runtime System, or LoRS. The LoRS layer consists of a C API and a command line interface tool set that automate the finding of IBP depots via the L-Bone, creating and using IBP capabilities and creating and managing exNodes [CwLL03]. The LoRS library also provides flexible tools to deal with the lower levels of the Network Storage Stack. Sample network file-based functions [BBF+02] include: Upload, Download, Augment, Trim, Refresh, and List. LoRS supports checksums to ensure end-to-end correctness, multiple encryption algorithms since IBP depots are public, untrusted servers, and compression to reduce the amount of data transferred
and stored.

3.2 GTP with Network Storage

In the process of developing GTP, we realized that parsing large collections generates numerous large files that take up a lot of valuable disk space. The Logistical Networking Testbed developed at LoCI facilitated the temporary storage of these files on a remote network (Internet) along with immediate retrieval when needed. In the course of creating an index for a document collection, new documents may get added to the collection or some documents may be deleted. In any case, before the final collection is created, several revisions are usually done and the user may need to parse the collection multiple times. In some cases the collection is dynamic, as is the case with webpages (HTML), so that parsing is done on a regular basis in order to monitor updates.

If the user keeps all the files generated by GTP and GTPQUERY after each parsing, the subsequent output files will take up an excessive amount of local disk storage. Fortunately, the concept of network storage can alleviate this burden: the user can clean up his/her hard drive and store the information produced by the parser on a remote network. Since the storage provided by the Internet Backplane Protocol (IBP) is temporary, if the user is not satisfied with the results, he will not choose to extend the time on the files stored on the IBP and the storage will be automatically reused. If, on the other hand, the user wants to store the results of the parser permanently, he can either make sure that the time limits do not expire or he can download the files back to his personal machine and then write them to other media, e.g., a CD-ROM.

4. OPEN PROBLEMS AND FUTURE DIRECTIONS

4.1 Intermediate Form

Intermediate forms with varying degrees of complexity are suitable for different mining purposes. For a fine-grain domain-specific knowledge discovery task, it is necessary to perform semantic analysis to derive a sufficiently rich representation to capture the relationship between the objects or concepts described in the documents. However, semantic analysis methods are computationally expensive and often operate in the order of a few words per second. It remains a challenge to see how semantic analysis can be made much more efficient and scalable for very large text corpora.

4.2 Multilingual text refining

Whereas data mining is largely language independent, text mining involves a significant language component. It is essential to develop text refining algorithms, that process multilingual text documents and produce language-independent intermediate forms. While most text mining tools focus on processing English documents, mining from documents in other languages allows access to previously untapped information and offers a new host of opportunities.

4.3 Domain knowledge integration

Domain knowledge, not catered for by any current text mining tools, could play an important role in text mining. Specifically, domain knowledge can be used as early as in the text refining stage. It is interesting to explore how one can take advantage of domain information to improve parsing efficiency and derive a more compact intermediate form. Domain knowledge could also play a part in knowledge distillation. In a classification or predictive modeling task, domain knowledge helps to improve learning/mining efficiency as well as the quality of the learned model (or mined knowledge) [5]. It is also interesting to explore how a user's knowledge can be used to initialize a knowledge structure and make the discovered knowledge more interpretable.

4.4 Personalized autonomous mining

Current text mining products and applications are still tools designed for trained knowledge specialists. Future text mining tools, as part of the knowledge management systems, should be readily

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

usable by technical users as well as management executives. There have been some efforts in developing systems that interpret natural language queries and automatically perform the appropriate mining operations. Text mining tools could also appear in the form of intelligent personal assistants [6]. Under the *agent* paradigm, a personal miner would learn a user's profile, conduct text mining operations automatically, and forward information without requiring an explicit request from the user.

5. CONCLUSION

We have provided a very brief introduction to text data mining within the pages of this article. There are numerous challenges to the statistical community that reside within this discipline area. As in any data mining or exploratory data analysis effort, visualization of textual data is an essential part of the problem. The statistical community has a great deal to contribute to many of these problems. In this article we have presented the software advancements as well as some future directions in the area of text mining.

REFRENCES:

[1] Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., MIT Press, Cambridge, Mass., 1-36.

[2] Feldman, R. & Dagan, I. (1995) Knowledge discovery in textual databases (KDT). In proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, August 20-21, AAAI Press, 112-117.

[3] Hearst, M. A. (1997) Text data mining: Issues, techniques, and the relationship to information access. Presentation notes for UW/MS workshop on data mining, July 1997.

[4] Simoudis, E. (1996). Reality check for data mining. *IEEE Expert*, **11**(5).

[5] Tan, A.-H. (1997). Cascade ARTMAP: Integrating neural computation and symbolic knowledge processing. *IEEE Transactions on Neural Networks*, **8**(2), 237-250.

[6] Tan, A.-H. & Teo, C. (1998). Learning user profiles for personalized information dissemination. *In proceedings, International Joint Conference on Neural Networks (IJCNN'98)*, Alaska, 183-188.