Towards XML Mining: The Role of Kernel Methods

Buhwan Jeong, Daewon Lee, Jaewook Lee, and Hyunbo Cho
Department of Industrial and Management Engineering
Pohang University of Science and Technology (POSTECH)
San 31, Hyoja, Pohang, 790-784, South Korea
{*bjeong, woosuhan, jaewookl, hcho*}*@postech.ac.kr*

Abstract

XML mining is a unique application of data mining, in that it deals with structured XML contents. The introductory paper provides a brief but comprehensive review of milestones towards XML mining. XML mining is not a one-day outcome by chance, but an accumulated inheritance of continuous evolution from data mining throughout text mining and web mining. Furthermore, the paper envisages the applications of kernel methods to XML mining. Preliminary results on schema-matching simulation reveal the kernel methods for structured data are an adequate tool for XML mining.
keyword: Kernel methods, schema matching, structured data, text mining, XML mining

1 Introduction

XML[1] mining, first named in [1], is a unique application of data mining to XML contents. Since its introduction, XML has gained much attention such in business applications collaborations (e.g., ebXML[2] and Web Service[3]) and recently web personalization (e.g., Web 2.0[4]). In spite of its frequent use, we have accomplished very little to reason about XML. XML mining uncovers the explicit and implicit semantics of XML contents. The potential use of XML mining is very wide from a more precise and personalized search to data-centric enterprises integration.

XML mining is a collective consequence of a variety of efforts including not only the XML formalism, but also data mining, text mining, and recent web mining. As so far we know, no comprehensive research, albeit the very beginning, exists on drawing up its genealogy, in relationships with other mining areas. More importantly, XML mining is not a rehash application one just applies old data/text mining techniques to XML contents. The originality of XML mining requires new wineskins for this new wine – the XML formalism.

The paper aims at first revealing the consequent course of XML mining research from the conventional data mining throughout text mining and web mining. Second, the paper presents

[1] eXtensible Markup Language, http://www.w3.org/XML
[2] http://www.ebxml.org
[3] http://www.w3.org/2002/ws
[4] http://en.wikipedia.org/wiki/Web_2.0

a research layer for XML mining. Third, we envision to use kernel methods for the purpose of XML mining. Preliminary results on simulation are also provided.

The remainder of the paper is organized as follow: Next, a comparative study of XML mining with data/text mining is addressed. Our layered view of XML mining is highlighted, and followed by general ideas about how to apply kernel methods to each layer of XML mining. Consequently, preliminary experiments are conducted and conclusion remarks are given.

2 Mining Something: Data, Text, Web, and XML

This section succinctly reviews and compares various mining activities towards XML mining. For convenience, we categorize them into data mining (in a narrow sense), text mining, web mining, and XML mining.

2.1 Overview

Mining is an activity to discover potentially useful information from large volumes of data through some processes of purification of relevant data from irrelevant ones, transformation to other forms, and sometimes association with other data and known patterns. Various computation applications commonly accommodate those mining activities, including signal processing and outlier detection, pattern recognition, clustering and classification, prediction and control, data dimension reduction, information retrieval and extraction, knowledge discovery, and so forth. Here we concentrate more on data mining applications that consume non-vectorial data such as texts. Particularly for this, we identify three mining applications – text mining, web mining, and XML mining. Table 1 summarizes their characteristics distinguishing from traditional data mining handling numerical data. As shown in the table below, the type and source of data are the main differentiating criterion. New emerging mining activities are capable of non-vectorial data such as textual data, multimedia data, and structured texts, while the traditional data mining deals with numerical (and categorical) data only. Each mining activity is detailed in following subsections.

Table 1: A comparison table among data mining, text mining, web mining, and XML mining

Mining type	Data Mining	Text Mining	Web Mining	XML Mining
Data type	Vectorial data	Non-vectorial data		
	Numerical/categorical data	Textual data	Textual & multimedia data	Structured text
Problem type & issues	Clustering, classification, regression, prediction, optimization and control	Vector space model (VSM) LSA/PCA, NLP (word sense)	Web content mining, web structure mining, web usage mining	Knowledge formalism, structural similarity
Applications	Bio/chem-informatics, pattern recognition, fraud detection, robotics, market analysis, credit scoring, etc	Text clustering, categorization, and summarization; authorship identification; etc	Web search and retrieval, web topology analysis, usage pattern analysis, etc	Schema matching, personalized web, XML message mapping, ontology alignment, etc
Kernel methods	Support vector machine (SVM)	VSM, string kernel		String & tree kernel

2.1.1 Data Mining

In a narrow sense, data mining is the search process of implicit, previously unknown, but potentially useful information from voluminous vectorial data. In general, its tasks can be categorized into exploratory data analysis, descriptive modeling, predictive modeling, discovery patterns and rules, retrieval by content. To accomplish these tasks, data mining process consists of several steps: data cleansing, feature construction/extraction, algorithm and parameter selection, and interpretation and validation. Most research works currently focus either on development of new algorithms or on improvement of existing algorithms in terms of computation time and accuracy. The data mining algorithms have four basic components [2]:

- **Model and Pattern Structure** that determines the underlying structure or functional forms that we train from data.
- **Score function**, by which qualify of the constructed model are measured.
- **Optimization and Search Method** optimizing the score function and searching over different model and pattern structures.
- **Data Management Strategy** that handles data access efficiently during optimization.

Recently, kernel techniques play an important role in all components above. For example, support vector machines (SVMs) well define a classifier based on structured risk minimization and its score function is a quadratic form so we can avoid local minimum problems and easily optimize it. In addition, by using kernel trick, we are (implicitly) able to deal with the data set in the high-dimensional space efficiently. Kernel methods are also used as distance or similarity measures in a variety of data mining tasks such as clustering, classification/regression, dimensionality reduction, density estimation.

2.1.2 Text Mining

Compared with the traditional data mining, text mining is loosely characterized as the process of analyzing texts to extract information that is useful for particular purpose, however explicitly stated in the texts [3]. Since the textural data are unorganized, amorphous, and difficult to deal with algorithmically, the main research stream in text mining is how to transform the textual data into numerical one, while preserving the original meanings of texts, to be consumable by machine learning and statistical methods.

A state-of-the-art tool to handle textual data is a vector space model (VSM), a.k.a., bag-of-words, since its introduction in 1971 [4] [5]. A VSM is a term-by-document matrix, in which each element is an indicator whether or not the corresponding document contains a certain term (or phrase). However, the VSM ignores the order of terms in documents. To overcome this drawback, kernel methods for structured data (e.g., string kernel, tree kernel) [6] have recently gained a great attention in text mining. Such kernel methods are uniquely capable of handling the data while preserving as much their structural information as possible. Detailed descriptions about the VSM and kernel methods are given in Section 4.1.

Another issue to improve the performance of text mining is to properly interpret the meanings of a term by means of semantic similarity between terms. The semantic similarity is one of the hottest topics in NLP (natural language processing). Most of semantic similarity measures incorporate the synonym (and hyponym) relation among words/terms [7] [8] [9]. Since the fundamental idea behind most of text mining activities is to quantify the similarity among texts, the similarity measure plays a crucial role for that purpose, especially when the same concepts in documents are explained in different words.

2.1.3 Web Mining

Web mining is a specific application of text mining to web contents, structures, and usage [10], on which web mining falls into content mining, structure mining, and usage mining, respectively [11] [12]. First, web content mining analyzes the contents of web resources, recognized as a form of text mining in general even though recent advances in data mining are capable of manipulating other multimedia data including image, sound, video, etc. For textual web contents, the same mining techniques used in text mining are used. In addition, web content mining can take advantage of the (semi-)structured nature of web page texts. With the appearance of XML, the logical structure within a page has gained more attentions (see next subsection).

Second, web structure mining exploits the hyperlink structure/topology among web pages; thus, it focuses on sets of web pages, which are related to each other by hyperlinks. Through structure mining, one may identify the relative relevance of different pages that appear equally pertinent when analyzed with respect to their contents in isolation. One of commercially successful algorithms for web structure mining is PageRank [13] used in Google.com, which computes the relevance by counting the incoming hyperlinks from other pages.

Third, web usage mining takes care of transaction records for a certain web site, often log files in a web server, instead of its web pages. Therefore, it analyzes the behavior patterns of visitors and guesses their preferences. Based on the patterns, web usage mining enables personalized webs. Association rules and sequence mining are typical ones that guide visitors into next destinations (either associate products or other web pages) the visitors are possibly interested in. To sum up, a right direction is to systemically combine web content mining, structure mining, and usage mining together. For example, one may crawl and gather web pages using keyword-based content mining, rank their relevance using hyperlink scores and consequently recommend other associate pages when a user chooses a particular page. Today, this strategy is well suited for commercialization such as AdSense[5] of Google.com, AdCenter[6] of Microsoft, and SearchMarketing[7] of Yahoo.com.

2.1.4 XML Mining

XML mining is a special type of web content mining, but also unique in that contents in an XML document are modular and well-structured, while a web content is more likely a plain text document. That is, XML mining must be capable of manipulating the structure of contents as

[5] https://www.google.com/adsense
[6] https://adcenter.microsoft.com
[7] http://searchmarketing.yahoo.com

well as the contents themselves. Main research issues are how to construct content models from XML documents (i.e., knowledge representation) and how to compute their structural similarity. Since a tree representation is the native formalism for XML, many research works incorporate it including DOM (Document Object Model) [14], DFT (Discrete Fourier Transform) [15], SLVM (Structured Link Vector Model) [16], and EEV (Extended-Element Vector) and NFA-XML (Non-deterministic Finite Automata-XML) [1]. Depending on the representation, various algorithms to compute the structural similarity are adopted such as TED (Tree Edit Distance), (Weighted-) Tag Similarity, Kernel Matrix, etc [16] [17] [18].

The applications of XML mining are very diverse including personalized web, schema matching, XML message mapping, ontology alignment, data warehouse management, e-catalog mapping, web service discovery and composition, agent communication, etc [19] [20]. It makes more impacts on business-to-business (B2B) and government-to-business (G2B) applications than on business-to-customer (B2C) applications.

3 XML Mining Framework

We define XML mining as "Given a query XML document d_q, find the most similar, but not identical[8], (fragments of) XML document(s) d from a collection of documents D". XML mining is certainly distinguished from text and web mining, in that it deals with modularly structured contents while text and web mining handle un-/semi-structured ones, e.g., HTML documents. A novel approach capable of exploiting the structural information in an XML document as well as its contents must be envisaged. Our view of XML mining is depicted in Figure 1, in which an XML document is treated in three layers – external schema, context, and (textual) content. XML mining must sequentially and/or synthetically accommodate every layer as well as differently approach to each layer, i.e., schema matching, context mapping, and content mining[9]. Note that the first schema matching is schema level comparison, while the others are instance level comparison.

Schema Matching. A schema (e.g., XML Schema, DTD, XLANG NG, etc) is another XML document that governs its instance documents' grammatical and structural form; therefore, schema matching should be performed before comparing contents (i.e., XML instance documents). Schema matching compares two schemas to check whether they define a concept in an equivalent way (i.e., structural equivalence). In other words, matched schemas mean they are equivalently replaceable with only few modifications, if necessary. This schema matching ensures the minimal requirements that two XML documents having the same/similar structure are possibly used to capture concepts/information in a similar implementation context.

Context Mapping. The structured nature of XML documents gives us an advantage to precisely restrict the meaning of their contents. That is, ancestor elements to an element play its

[8]The exclusion of 'identical documents' is to avoid duplicate documents returned [21].

[9]Note that the terms 'matching', 'mapping', and 'mining' can be often interchangeably used to implicitly indicate 'association of similar objects'. However, in the paper, we intend to separate them to have more precise customary meanings.

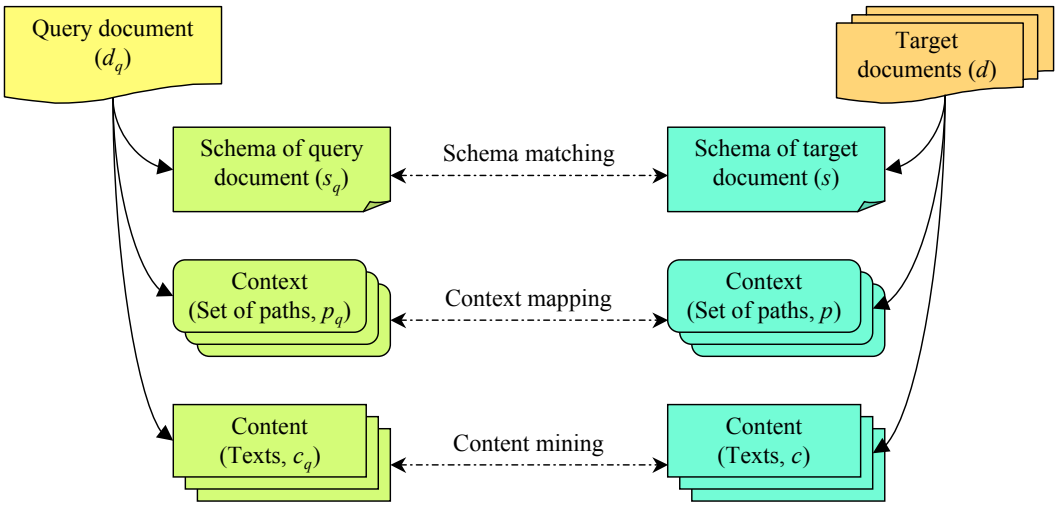

Figure 1: The proposed layered view of XML mining: schema matching, context mapping, and content mining

context role, thereby determining more specific meaning. For example, *Title* under *Person* means one's position, while that under *Book* implies the name of a book. The context of a content is defined as a path from the root element to the very element containing the content (i.e., a sequence of ancestor element names). The context mapping is a subsidiary activity to determine the candidate fragments to be mined from a target document, given a fragment of the query document. The idea behind this context mapping is to guide a mining tool to the right text source to mine.

Content Mining. Finally, the contents under mapped (or similar) contexts are compared. This content mining is wholly dependent on the results of context mapping. One may apply any traditional text mining techniques to the contents.

4 Kernel Methods and XML Mining

The section briefly introduces the kernel methods for structured data and then shows how to apply them to XML mining.

4.1 Kernel Methods for Structured Data

Kernel methods (support vector machines in particular) easily converts a linear algorithm into a non-linear one by mapping the original samples into a higher-dimensional non-linear Hilbert space $\mathcal{H}$ so that a linear model in the new space $\mathcal{H}$ is equivalent to a non-linear model in the original space $\mathcal{X}$. However, this transformation often results in catastrophic computation complexity [22]. Fortunately, so-called *kernel trick* resolves it by getting the scalar product implicitly computed in $\mathcal{H}$ when an algorithm solely depends on the inner product between vectors. Fur-

thermore, recently invented kernel methods can effectively incorporate other types of data (i.e., structured data) than numerical data.

4.1.1 Vector Space Model (VSM)

The VSM is a special type of kernel methods for textual data. It transforms a document into a numerical vector, each element of which is an indicator whether or not the document contains a certain word (or phrase). A VSM can be represented either in a boolean vector model, which has only zeros or ones indicating respectively words' absence or presence, or in a term weighting model, which has scalar values that take into account the appearance frequency of a term in a set of documents. A widely used weighting method is 'Term Frequency-Inverse Document Frequency (TF-IDF)' [5], in which the weight of the i^{th} term in the j^{th} document, denoted as w_{ij}, is defined by $w_{ij} = TF_{ij} \times IDF_i = TF_{ij} \times log(n/DF_i)$, where TF_{ij} is the number of occurrences of the i^{th} term within the j^{th} document, and DF_i is the number of documents (out of n) in which the term appears. For two documents v_1 and v_2, their semantic distance is computed as the cosine of their angle θ, i.e., $d(v_1, v_2) = \cos\theta = \mathbf{V}_{v_1}^{\mathrm{T}}\mathbf{V}_{v_2}/\|\mathbf{V}_{v_1}\|_2\|\mathbf{V}_{v_2}\|_2$. Since this interpretation requires too much computation for massive real-world documents, alternatively it is often to use LSA (or LSI, Latent Semantic Analysis/Indexing)[10] [23] and PCA (Principal Components Analysis) [5] of the original VSM for computation efficiency.

4.1.2 String and Tree Kernel

The VSM suffers from the fact that it retains the occurrence frequency of words only, but ignores their order in a document. Alternatively, recent kernel methods, i.e., string kernels, accommodate the order of words, based on the similarity of two strings on the number of common subsequences. These subsequences need not be contiguous in the strings but the more gaps in the occurrence of the subsequence, the less weight is given to it in the kernel function. For example, take two strings 'cat' and 'cart'. Clearly the common subsequences are 'c', 'a', 't', 'ca', 'at', 'ct', and 'cat' and they are, respectively, represented in penalties of 'c':$(\lambda^1\lambda^1)$, 'a':$(\lambda^1\lambda^1)$, 't':$(\lambda^1\lambda^1)$, 'ca':$(\lambda^2\lambda^2)$, 'at':$(\lambda^2\lambda^2)$, 'ca':$(\lambda^3\lambda^4)$, and 'cat':$(\lambda^3\lambda^4)$ after applying penalties based on the gaps in the occurrence of a subsequence (i.e., the total length of a subsequence in the two strings) with a decay factor λ. The kernel function is then simply the sum over these penalties, i.e., $k(\text{cat}, \text{cart}) = 2\lambda^7 + \lambda^5 + \lambda^4 + 3\lambda^2$ [24]. It is worthy noting that by superseding the alphabets (characters) by words (or syllables in some cases) the same idea and computation behind the string kernels are directly applied to documents manipulation. More theoretical aspects and variants of string subsequence kernels are found in [6] [24] [25] [26].

Another popular structured data are formed in a (labeled ordered) tree. The key idea to capture such tree structural information in a kernel function is to consider all sub-trees occurring in a parse tree, i.e., parse tree kernel [27]. In addition, a recent tree kernel is an extension to string kernels, in which a sequence of node labels in the order of a depth-first traversal is constructed and each node label is identified as a symbol [28].

[10] LSA uses singular value decomposition (SVD) for data dimension reduction.

4.2 Kernel-based XML Mining

The kernel methods for structured data are applicable to XML mining in various ways. The string kernels with minor modifications are used to not only identify and extract useful information from textual contents as used in [6], but also to compare two pieces of context information (i.e., successive element labels in a path). Undoubtedly, the VSM also provides the same state-of-the-art performance[11] in XML content mining as in traditional text mining. The tree kernels are used to measure the structural similarity between XML schema documents and between XML instance documents. By transforming the tree structure into a string (e.g., by a depth-first traversal), the string kernels and the VSM are also used to compute the structural similarity. Followings succinctly describe how to transform an XML document into formats digestible by corresponding kernel methods.

Schema Matching. Schema matching is to choose the most similar pair of schemas in terms of structural similarity. Schema matching has two folds – comparison between XML instance documents and comparison between XML schema documents. Computing the structural similarity between a pair of XML instance documents is relatively easy because we have only to apply the tree kernels to their DOM trees or reduced DOM trees. However, checking the structural equivalence between schemas is not arbitrary because the schema structures of interest are hidden behind their DOM trees. To cope with this problem, we design a Core Tree representation that captures the intrinsic structure of a schema. The core tree actually exploits the most collective and expressive structure among diverse instance documents derived from a schema. Simply stated here, the root node of a core tree is set by the label of a schema's root element (i.e., the value of *name* attribute) and its child nodes are determined as the element names specified in its type definition (i.e., *complexType*). Recursively, child nodes are appended. In this case, to make the core tree ordered, the left-to-right order of child nodes should be preserved as specified in constructors of *sequence*, *union*, *choice*, etc.

Context Mapping. Context mapping is to make candidate pairs of contents to be compared in content mining in order to improve the mining accuracy. Since the meaning of an element (and its content) is restricted by it ancestor elements, the context is represented in a path from the root element to the very element. When XML instances conform to different schemas, their corresponding schemas may also differently define the same concept, that is, a concept can have different names (e.g., *ConferencePaperTitle* vs. *ConferenceProceedings/FullPaper/Title*)[12] in different schemas. To minimize effects from such differences in element names, each name should be normalized through a recursive series of tokenization (i.e., separation by tokens such as punctuation, cases, blank characters, digit, etc), lemmatization (i.e., morphological analysis to find the basic form), and elimination (i.e., discard of unnecessary or less informative words such as articles, prepositions, conjunction, an so on) [19] [20].

[11] As shown in Section 5, string kernels give a better performance than the VSM does.

[12] An element/attribute name is often defined by a compound word concatenated with several words. In addition, different schemas are often accordance with different NDR's (Naming and Design Rule).

Content Mining. Content mining is the same process as traditional text mining, in that it transforms textual contents into a VSM or directly applies string kernels to the contents. To improve mining accuracy, the contents also need some normalization processes such as elimination and stemming (e.g., Porter stemming algorithm [29]).

For more practical mining, it is important to properly determine the decay factor (i.e., $\lambda \leq 1$), regardless the use of the exponential function. In addition, one way to incorporate prior knowledge into the kernel methods is to use variant decay factors to each word (or node). One is word soft matching that takes the synonymous relation among words into consideration [30], and another is a node-depth dependent decay factor, i.e., $\lambda_n = \frac{\lambda_0}{depth(n)^r}$, where $depth(n)$ is the depth of the node n ($depth(root) = 1$) and $r \geq 1$ is a relevant factor.

5 Preliminary Experiment

To evaluate kernel's applicability to XML mining, we conduct a preliminary experiment that measures various structural similarities between 200 pairs of randomly selected CC (Core Components) schemas[13]. For this experiment, we transform each schema (i.e., core tree) into a normalized string by a depth-first traversal. As shown in Figure 2, the kernel methods (including VSM's) outperform TED[14] in terms of correlation with human judgment[15]. Moreover, the string kernels (i.e., KN.1 and KN.2) give much better performance than VSM's do.

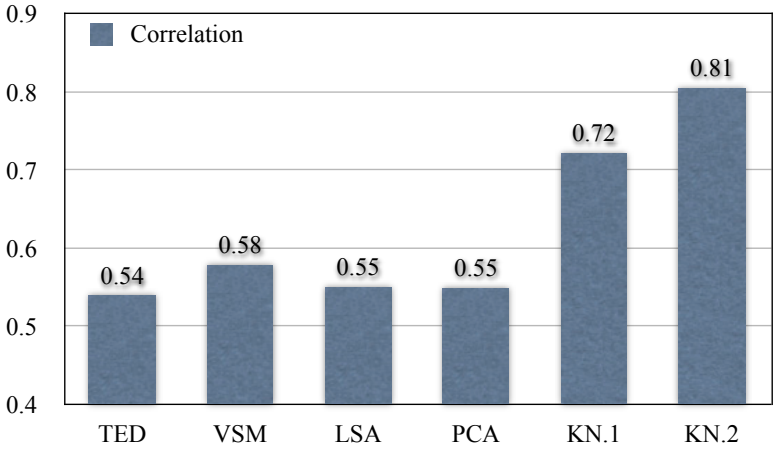

Figure 2: Correlation between human judgment and various structural similarity measures – TED, VSM with cosine of the angle, VSM with LSA, VSM with PCA, and two Kernel-based measures (i.e., KN.1 and KN.2 with different decay factors λ).

[13]The schemas come from OAGIS BOD at http://www.openapplications.org.

[14]TED (Tree Edit Distance) is a state-of-the-art structural similarity for tree structures.

[15]Four human experts score the similarity of every schema pair.

6 Conclusion

XML mining is a very promising area to data mining as the explosive use of XML. XML mining is a unique application distinguishing from the traditional text mining, in that it deals with structured contents. Therefore, the research issues in XML mining is how to incorporate XML's structured nature. To this end, we envision a layered XML mining consisting of schema matching (or structural equivalence), context mapping, and content mining. Instead of transforming textual data into numerical ones (as conventional text mining approaches have done), it is more desirable to manipulate the textual data as-is. For this, recently developed kernel methods for structured data are very useful. As SVMs have done, we expect recent kernels for structured data will boost up not only XML mining, but also text mining. Nonetheless, since such kernels are mainly developed for bio/chem-informatics, new variant kernels for texts (and XML contents) are required.

References

[1] J. Lee, K. Lee, and W. Kim, "Preparations for semantics-based XML mining," in *Proceedings of IEEE International Conference on Data Mining (ICDM2001)*, Nov. 2001, pp. 345–352.

[2] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Cambridge, MA: The MIT Press, 2001.

[3] I. Witten, *Text Mining*. Boca Raton, FL: Chapman & Hall/CRC Press, 2005, pp. 14.1–14.22.

[4] M. Berry, Z. Drmac, and E. Jessup, "Matrices, vector spaces, and information retrieval," *SIAM Review*, vol. 42, no. 2, pp. 335–362, 1999.

[5] M. Kobayashi and M. Aono, *Vector Space Models for Search and Cluster Mining*. Springer-Verlag New York, Inc., 2003, pp. 103–122.

[6] H. Lodhi, J. Shawe-Tayler, N. Christianini, and C. Watkins, "Text classification using string kernels," in *Advances in Neural Information Processing Systems*, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press, 2001, vol. 13.

[7] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, Aug. 1995, pp. 448–453.

[8] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet::Similarity – Measuring the relatedness of concepts," in *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI'04)*, July 2004.

[9] M. Jarmasz and S. Szpakowicz, "Roget's thesaurus and semantic similarity," in *Proceedings of Conference on Recent Advances in Natural Language Processing (RANLP)*, Sept. 2003, pp. 212–219.

[10] G. Stumme, A. Hotho, and B. Berendt, "Semantic web mining: State of the art and future directions," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, no. 2, pp. 124–143, June 2006.

[11] R. Kosala and H. Blockeel, "Web mining research: A survey," *ACM SIGKDD EXPLORATIONS*, vol. 2, no. 1, July 2000.

[12] R. Cooley, J. Srivastava, and B. Mobasher, "Web mining: Information and pattern discovery on the world wide web," in *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, Nov. 1997.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," in *Proceedings of the 7th International World Wide Web Conference*, 1998, pp. 161–172.

[14] A. Nierman and H. Jagadish, "Evaluating structural similarity in XML documents," in *Proceedings of the 5th International Workshop on the Web and Database (WebDB2002)*, June 2002.

[15] S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese, "Fast detection of xml structural similarity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 2, Feb. 2005.

[16] J. Yang, W. Cheung, and X. Chen, "Learning the kernel matrix for XML document clustering," in *Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*. Washington, DC.: IEEE Computer Society, 2005, pp. 353–358.

[17] B. Jeong, B. Kulvatunyou, N. Ivezic, H. Cho, and A. Jones, "Enhance reuse of standard e-business XML schema documents," in *Proceedings of International Workshop on Contexts and Ontology: Theory, Practice and Application (C&O'05) in the 20th National Conference on Artificial Intelligence (AAAI'05)*, July 2005.

[18] D. Buttler, "A short survey of document structure similarity algorithms," in *Proceedings of the 5th International Conference on Internet Computing (IC2004)*, June 2004.

[19] P. Shvaiko and J. Euzenat, "A survey of scham-based matching," *Journal of Data Semantics IV*, vol. 3730, pp. 14–171, July 2005.

[20] B. Jeong, "Machine learning-based semantic similarity measures to assist discovery and reuse of data exchange XML schemas," Ph.D. dissertation, Department of Industrial and Management Engineering, Pohang University of Science and Technology, June 2006.

[21] S. Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan-Kauffman Series in Data Management Systems, 2002.

[22] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.

[23] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, pp. 259–284, 1998.

[24] T. Gärtner, "A survey of kernels for structured data," *ACM SIGKDD EXPLORATIONS Newsletter*, vol. 5, no. 1, pp. 49–58, 2003.

[25] C. Leslie, E. Eskin, and W. Noble, "The spectrum kernels: A string kernel for SVM protein classification," in *Proceedings of the Pacific Symposium on Biocomputating*, 2002, pp. 564–575.

[26] G. Paass, E. Leopold, M. Larson, J. Kindermann, and S. Eickeler, "SVM classification using sequences of phonemes and syllables," in *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*. Springer-Verlag, 2002, pp. 373–384.

[27] M. Collins and N. Duffy, "Convolution kernels for natural language," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Cambridge, MA: MIT Press, 2002.

[28] S. Vishwanathan and A. Smola, "Fast kernels for string and tree matching," in *Advances in Neural Information Processing Systems*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, vol. 15.

[29] P. Willett, "The porter stemming algorithm: Then and now," *Electronic Library and Information Systems*, vol. 40, no. 3, pp. 219–223, 2006.

[30] N. Cancedda, E. Gaussier, C. Goutte, and J. Renders, "Word-sequence kernels," *Journal of Machine Learning Research*, vol. 3, pp. 1059–1082, 2003.