MRAR: Mining Ranked Association Rules Using Information Extraction

M. Kanaka Surya Padmaja
MTech,
CMRCET

S. Siva Skandha
Asst. Professor,
CMRCET

Ch. Raja Kishore
Asst. Professor,
CMRCET

Abstract: In this paper, we develop a text-mining system by integrating methods from Information Extraction (IE) and Data Mining (Knowledge Discovery from Databases or KDD). By utilizing existing IE and KDD techniques, text-mining systems can be developed relatively rapidly and evaluated on existing text corpora for testing IE systems. We present a general text-mining framework called MRAR which employs an IE module for transforming natural-language documents into structured data and a KDD module for discovering prediction rules from the extracted data. We present experimental results on inducing prediction and ranked association rules from natural-language texts demonstrating that MRAR learn more accurate rules than previous methods for these tasks. We also present an approach to using rules mined from extracted data to improve the accuracy of information extraction. Experimental results demonstrate that such discovered patterns can be used to effectively improve the underlying IE method.

I. INTRODUCTION

The recent abundance of digital information available electronically has made the organization of textual information into an important task. Text mining is a burgeoning new technology for discovering knowledge from text data. With the fast growth of the number of pages on the World Wide Web, text mining plays a key role in managing information and knowledge, and is therefore attracting increasing attention (Berry, 2003b, 2003a; Feldman, 1999; Hearst, 2003, 1999; Grobelnik, 2001, 2003; Mladenic, 2000; Muslea, 2004).

Text Data Mining and Information Extraction

Data Mining (DM) or Knowledge Discovery in Databases (KDD) is the process of identifying novel and understandable patterns in data (Han & Kamber, 2000; Witten & Frank, 1999). Data mining seeks not only information or answers to the question which the user already knows to ask, but discovers deep knowledge embedded within the data. In order to do that, data mining applies computational techniques, usually in the form of a learning algorithm, to find potentially useful patterns in the data. Most existing data mining approaches look for patterns in a relational table of data (Agrawal, Imielinsky, & Swami, 1993).

Text mining or text data mining, the process of finding useful or interesting patterns, models, directions, trends, or rules from unstructured text, is used to describe the application of data mining techniques to automated discovery of knowledge from text (Chakrabarti, 2002; Han & Kamber, 2000). Generally text mining has been viewed as a natural extension of data mining (Hearst, 2003, 1999). This reflects the fact that the advent of text mining relies on the burgeoning field of data mining to a great degree.

However, unlike data mining, which focuses on the well-structured collections that exist in either relational databases or data warehouses, text mining excavates data that is far less structured. Much of today's electronic data resides not in traditional relational databases, but “hidden” in the Web and natural-language documents. In this paper, we present a new framework for text mining based on the integration of traditional data mining and Information Extraction (IE).

The goal of an IE system is to find specific data in natural-language texts. The data to be extracted is typically given by a template which specifies a list of slots to be filled with substrings taken from the document. IE is useful for a variety of applications, particularly given the recent proliferation of Internet and web documents. Recent applications include course and research project homepages (Freitag, 1998a; Thompson, Smarr, Nguyen, & Manning, 2003), seminar announcements (Freitag, 1998b), apartment rental ads (Soderland, 1999), job announcements (Califf & Mooney, 1999), geographic web documents (Etzioni, Cafarella, Downey, Kok, Popescu, Shaked, Soderland, Weld, & Yates, 2004), government reports (Pinto, McCallum, Wei, & Croft, 2003), and medical abstracts (Bunescu, Ge, Kate, Marcotte, Mooney, Ramani, & Wong, 2004).

Traditional data mining assumes that the information to be “mined” is already in the form of a relational database. Unfortunately, for many applications, electronic information is only available in the form of unstructured natural-language documents rather than structured databases. IE addresses the problem of transforming a corpus of textual documents into a more structured database, thereby suggesting an obvious role that can be played in text mining when combined with standard KDD methods. In this paper, we suggest using an IE module to locate specific pieces of data in raw text, and to provide the resulting database to the KDD module for rule mining.

II. TEXT MINING AND INFORMATION EXTRACTION

“Text mining” is used to describe the application of data mining techniques to automated discovery of useful or interesting knowledge from unstructured text. Several techniques have been proposed for text mining including conceptual structure, association rule mining, episode rule mining, decision trees, and rule induction methods. In addition, Information Retrieval (IR) techniques have widely used the "bag-of-words" model for tasks such as document matching, ranking, and clustering.

The related task of information extraction aims to find specific data in natural-language text. DARPA's Message Understanding Conferences (MUC) have concentrated on IE by evaluating the performance of participating IE systems based on blind test sets of text documents. The data to be extracted is typically given by a template which specifies a list of slots to be filled with substrings taken from the document. This template includes slots that are filled by strings taken directly from the document. Several slots may have multiple fillers for the job-posting domain as in programming languages, platforms, applications, and areas.

We have developed machine learning techniques to automatically construct information ex-tractors for job postings, such as those listed in the USENET newsgroup misc.jobs.offered. By extracting information from a corpus of such textual job postings, a structured, searchable database of jobs can be automatically constructed; thus making the data in online text more easily accessible. IE has been shown to be useful in a variety of other applications, e.g. seminar announcements, restaurant guides, university web pages, apartment rental ads, and news articles on corporate acquisitions.

III. INTEGRATING DATA MINING AND INFORMATION EXTRACTION

In this section, we discuss the details of our proposed text mining framework, DISCOTEX (Discovery from Text Extraction). We consider the task of first constructing a database by applying a learned information-extraction system to a corpus of natural-language documents. Then, we apply standard data-mining techniques to the extracted data, discovering knowledge that can be used for many tasks, including improving the accuracy of information extraction.

The DISCOTEX System

In the proposed framework for text mining, IE plays an important role by preprocessing a corpus of text documents in order to pass extracted items to the data mining module. In our implementations, we used two state-of-the-art systems for learning information extractors, RAPIER (Robust Automated Production of Information Extraction Rules) and BWI (Boosted Wrapper Induction). By training on a corpus of documents annotated with their filled templates, they acquire a knowledge base of extraction rules that can be tested on novel documents. RAPIER and BWI

Document

Title: Web Development Engineer Location: Beaverton, Oregon

This individual is responsible for design and implementation of the web-interfacing components of the AccessBase server, and general back-end development duties.

A successful candidate should have experience that includes:

One or more of: Solaris, Linux, IBM AIX, plus Windows/NT Programming in C/C++, Java

Database access and integration: Oracle, ODBC CGI and scripting: one or more of Javascript,

VBScript, Perl, PHP, ASP

Exposure to the following is a plus: JDBC, Flash/Shockwave, FrontPage and/or Cold Fusion.

A BSCS and 2+ years experience (or equivalent) is required.

Filled Template

- title: 'Web Development Engineer"
- location: 'Beaverton, Oregon"
- languages: 'C/C++", 'Java", 'Javascript", 'VBScript", 'Perl", 'PHP", 'ASP"
- platforms: 'Solaris", 'Linux", 'IBM AIX" 'Windows/NT"
- applications: 'Oracle", 'ODBC", 'JDBC", 'Flash/Shockwave", 'FrontPage", 'Cold Fusion"
- areas: 'Database", 'CGI", "scripting"
- degree required: 'BSCS"
- years of experience: '2+ years"

Figure: Sample text and filled template for a job posting

IV. USING MINED RULES TO IMPROVE IE

After mining knowledge from extracted data, DISCOTEX can predict information missed by the previous extraction using discovered rules. In this section, we discuss how to use mined knowledge from extracted data to aid information extraction itself.

The Algorithm

Tests of IE systems usually consider two performance measures, precision and recall defined as:

$$F := \phi$$

Precision = $D \in \Upsilon$

$$R \in RB$$

Recall= $\dfrac{\text{Number of correct fillers extracted}}{\text{Number of fillers in correct templates}}$

Many extraction systems provide relatively high precision, but recall is typically much lower. Previous experiments in the job postings domain showed RAPIER'S precision (e.g. low 90%'s) is singificantly higher than its recall (e.g. mid 60%'s). Currently, RAPIER'S search focuses on finding high-precision rules and does not include a method for trading-off precision and recall. Although several methods have been developed for allowing a rule learner to trade-off precision and recall, this typically leaves the overall F-measure unchanged.

By using additional knowledge in the form of prediction rules mined from a larger set of data automatically extracted from additional unannotated text, it may be possible to improve recall without unduly sacrificing precision. For example, suppose we discover the rule "VoiceXMLE *language*" "Mobile E *area*". If the IE system extracted "VoiceXML E *language*" but failed to extract "Mobile", we may want to assume there was an extraction error and add "Mobile"to the area slot,

potentially improving recall. Therefore, after applying extraction rules to a document, DISCOTEX applies its mined rules to the resulting initial data to predict additional potential extractions.
Input: RB is the set of prediction rules.
That make any incorrect predictions on either the training or validation extracted templates are discarded. Since association rules are not intended to be used together as a set as classification rules are, we focus on mining prediction rules for this task.
The extraction algorithm which attempts to improve recall by using the mined rules is summarized. Note that the final decision whether or not to extract a predicted filler is based on whether the filler (or any of its synonyms) occurs in the document as a substring. If the filler is found in the text, the extractor considers its prediction confirmed and extracts the filler.

V. APPROACH OF INFORMATION EXTRACTION

Extracting skill types and skill values: The algorithm to extract skill types and skill values is divided into two parts. In the first step, we apply the preprocessing and in the second step we apply an algorithm described in Table 3 to identify the skill type and skill value features. The preprocessing steps are as follows:

(i) Entire input text is converted to lower case and special characters are removed.

(ii) Stop words occurring in general purpose stop words list are removed.

(iii) The skills section in resume is identified with the keyword ‘Skills’ in the heading irrespective of the position of the Skills section in the resume.

(iv) The skill type and its skill value(s) are identified and separately stored using a delimiter (: in our case).

(v) Skill value(s) corresponding to each skilltype are sorted lexically and separated by a comma (,). For a skill value having more than one word, the words are concatenated. For example, the skill values for skill feature ‘database technologies: ms sql, postgres sql, mysql’ would be changed to skill value string: ‘mssql, mysql,postgressql’.

(vi) To resolve human errors like spelling mistakes, typo errors etc., we define a data structure called ‘skill values list’ with ‘skill type’ as a hash key and its possible ‘skill values’ as its values. Each skill value is checked in the skill values list. In case of many partial matches, the skill value is replaced by the skill value from the list with which it has the longest match. In case of no match, the list is manually updated with the skill value after verification. The possible skill values are extracted from the resume dataset.

(vii) A skill value can have more than one name referring to it. For example mssql and microsoft sql refers to same skill. To resolve such ambiguity we identify the various possible ways of redundant occurrences through data analysis and prepare a hash table with the canonical names as the hash key and various possible names as a list of hash values corresponding to the canonical name. All these different names should be replaced by a common name or canonical name to resolve this issue.

5.1 Finding Frequent Item sets

- Build a compact Tree structure using 2 passes over the data-set.
- Extracts frequent item sets directly from that tree
- Scan data and find support for each item. –Scan data and find support for each item.
- Discard infrequent items.
- Sort frequent items in decreasing order based on their support.
- Use this order when building the Tree, so common prefixes can be shared.
- Extract frequent item sets from the tree as follows.
- Divide and conquer: first look for frequent item sets ending in e, then de, etc. . . then d, then cd, etc. . .
- First, extract prefix path sub-trees ending in an item (set). (hint: use the linked lists)
- Each prefix path sub-tree is processed recursively to extract the frequent item sets. Solutions are then merged.

5.2 Finding the set of all valid association rules

- An association rule X-->Y is a relationship between two itemsets X and Y such that X and Y are disjoint and are not empty. A valid rule is a rule having a support higher or equals to minsup and a confidence higher or equal to minconf. The support is defined as sup(x-->Y) = sup (X U Y) / (number of transactions). The confidence is defined as sup(x-->Y) = sup (X U Y) / sup (X).
- Now the relationship between itemset and association rule mining is that it is very efficient to use the frequent itemset to generate rules (see the paper by Agrawal 1993) for more details about this idea. So association rule mining will be broken down into two steps: - mining frequent itemsets - generating all valid association rules by using the frequent itemsets.

5.3 Ranking the Association Rules by their weights:

Class Descriptor: The domain expert classifies the attributes involved in transactions will be classified in to different categories. The process of classification as follow

1. Initially classes will be derived based on the properties; hence each class contains set of properties. These classes can be recursive i.e., a class may refer one or more other classes as sub classes.
2. Based on attribute properties, attributes will be categorized into a class.
   Ex: if most of the attribute ‘a’ properties matched to class ‘c’ then $a \in c$
3. The domain expert also initiates to derive the relation between classes. The relation can be between any two classes, such as

Relation between class and sub-class of other class
Relation between two direct classes
Relation between two sub classes
Note: All related classes of a sub class also related to it’s parent class

VI. Experiments and Results exploration

We used correctness of the associability as the evaluation metric. Adding to this metric, precision, recall, and F-measure also being measured and compared with DISCTEX[16] that is used earlier to extract association rules by Information extraction. These supplementary procedures are definite using fallowing equations.

$$pr = \frac{t_+}{t_+ + f_+}$$

Here in this equation the pr signifies the accuracy, t_+ signifies the true positives and f_+ signifies the false positive

$$rc = \frac{t_+}{t_+ + f_-}$$

Here in the Eq-4.3.2, the 'rc' signifies the recollect, 'f_-' signifies the false negative.

$$F = \frac{2 * pr * rc}{pr + rc}$$

Here in this equation, 'F' signifies the F-measure.

Table 1: Accuracy, recall and F-measure values for feature selection methods (filter, wrapper, GFO) got from experiments accomplished by SVM

	RAPIER	DISCOTEX	MRAR
Precision	0.9852	0.9876	0.9952
Recall	0.9783	0.9743	0.9923
F-Mesure	0.981738	0.980905	0.993748

Fig2: A assessment account of accuracy, recollect and f-measure obtained for filter, wrapper and projected GFO

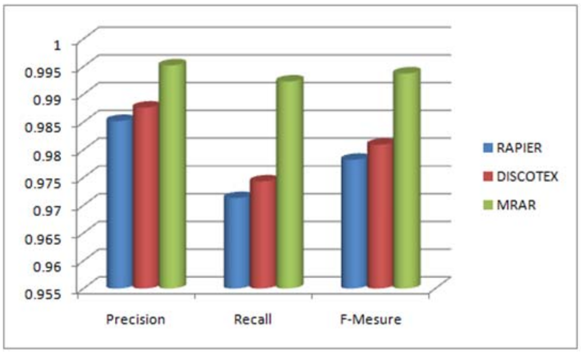

Table2: The explanations linked to association rule mining from text data making use of Information Extraction.

VII. Conclusion

The goal of text mining is to discover knowledge in unstructured text. The related task of IE concerns transforming unstructured text into a structured database by locating desired pieces of information. Although handmade IE systems have existed for a while, automatic construction of information extraction systems using machine learning is more recent. MRAR combines IE and standard data mining methods to perform text mining as well as improve the performance of the underlying IE system. It discovers prediction rules from natural-language corpora, and these rules are used to predict additional information to extract from future documents, thereby improving the recall of IE.

Existing methods for mining rules from text use hard, logical criteria for matching rules. However, for most text processing problems, a form of soft matching that utilizes word-frequency information typically gives superior results. Therefore, the induction of soft-matching rules from text is an important, under-studied problem. The standard rule mining algorithms have problems when the same extracted entity or feature is represented by similar but not identical strings in different documents. Consequently, we developed an alternate rule induction system called MRAR that allows for partial matching of textual features.

We presented experimental results applying to documents retrieved from the USENET. The empirical results obtained shows that MRAR focuses on inducing accurate rules by gradually ranking textual instances in contrast to DISCOTEX and RAPIER.

In conclusion, we have presented a general framework for text mining by combining existing IE and KDD technologies, which was shown to give better efficiency in mining soft patterns. Instead of considering the documents as a simple bag of words or a string, we used a flexible method of plugging in a similarity metric for each field. Both rule- learning systems for automated discovery of knowledge from unstructured text were demonstrated to perform better than previous methods in several domains.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedingsof the 20th International Conference on V ery Large Databases (VLDB-94), pages 487–499,Santiago, Chile, Sept. 1994.

[2] R. Baeza-Y ates and B. Ribeiro-Neto. Modern Information Retrieval. ACM Press, New Y ork,1999.

[3] S. Basu, R. J. Mooney , K. V . Pasupuleti, and J. Ghosh. Evaluating the novelty of text-minedrules using lexical knowledge. In Proceedings of the Seventh ACM SIGKDD InternationalConference on Knowledge Discovery and Data Mining (KDD-2001), pages 233–239, SanFrancisco, CA, 2001.

[4] M. W . Berry, editor . Proceedings of the Third SIAM International Conference on DataMining(SDM-2003) W orkshop on T ext Mining, San Francisco, CA, May 2003.

[5] M. E. Califf, editor . P apers fr om the Sixteenth National Conference on Artificial Intelligence(AAAI-99) W orkshop on Machine Learning for Information Extraction, Orlando, FL, 1999.AAAI Press.

[6] M. E. Califf and R. J. Mooney . Relational learning of pattern-match rules for informationextraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence(AAAI-99), pages 328–334, Orlando, FL, July 1999.

[7] C. Cardie. Empirical methods in information extraction. AI Magazine, 18(4):65–79, 1997.

[8] C. Cardie and R. J. Mooney . Machine learning and natural language (Introduction to specialissue on natural language learning). Machine Learning, 34:5–9, 1999.

[9] F . Ciravegna and N. Kushmerick, editors. P apers fr om the 14th European Conference on Machine Learning(ECML-2003) and the 7th European Conference on Principles and Practiceof Knowledge Discovery in Databases(PKDD-2003) W orkshop on Adaptive T ext Extractionand Mining, Cavtat-Dubrovnik, Croatia, Sept. 2003.14

[10] W . W . Cohen. Fast effective rule induction. In Proceedings of the T welfth InternationalConference on Machine Learning (ICML-95), pages 115–123, San Francisco, CA, 1995.

[11] W . W . Cohen. Learning to classify English text with ILP methods. In L. De Raedt, editor,Advances in Inductive Logic Programming, pages 124–143. IOS Press, Amsterdam, 1996.

[12] W . W . Cohen. Improving a page classifier with anchor extraction and link analysis. InS. Becker, S. Thrun, and K. Obermayer, editors, Advances in Neural Information ProcessingSystems 15, pages 1481–1488, Cambridge, MA, 2003. MIT Press.

[13] DARP A, editor . Proceedings of the Seventh Message Understanding Evaluation and Conference (MUC-98), Fairfax, V A, Apr. 1998. Morgan Kaufmann.

[14] R. Feldman, M. Fresko, H. Hirsh, Y . Aumann, O. Liphstat, Y . Schler, and M. Rajman. Knowledge management: A text mining approach. In U. Reimer, editor, Proceedings of Second International Conference on Practical Aspects of Knowledge Management (P AKM-98), pages9.1–9.10, Basel, Switzerland, Oct. 1998.

[15] D. Freitag and N. Kushmerick. Boosted wrapper induction. In Proceedings of the SeventeenthNational Conference on Artificial Intelligence (AAAI-2000), pages 577–583, Austin, TX, July2000. AAAI Press / The MIT Press.

[16] Raymond J. Mooney and Un Y ong Nahm; T ext Mining with Information Extraction; Multilingualism and Electronic Language Management: Proceedings of the 4th International MIDP Colloquium, September 2003, Bloemfontein, South Africa, Daelemans, W., du Plessis, T., Snyman, C. and Teck, L. (Eds.), pp.141-160, Van Schaik Pub., South Africa, 2005