ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

Data Mining: How to Mining Knowledge from Text

Ritesh Kumar
Assistant Professor, Department of Computer Science & Engineering, School of Engineering and Technology, Bahadurgarh, Haryana

ABSTRACT

An important approach to text mining involves the use of natural-language information extraction. Information extraction (IE) distills structured data or knowledge from un- structured text by identifying references to named entities as well as stated relationships between such entities. IE systems can be used to directly extricate abstract knowledge from a text corpus, or to extract concrete data from a set of documents which can then be further analyzed with traditional data-mining techniques to discover more general patterns. We discuss methods and implemented systems for both of these approaches and summarize results on mining real text corpora of biomedical abstracts, job announcements, and product descriptions.

Keywords: *Data mining .*

1. INTRODUCTION

Most data-mining research assumes that the information to be “mined” is already in the form of a relational database. Unfortunately, for many applications, available electronic information is in the form of unstructured natural-language documents rather than structured databases. Consequently, the problem of text mining, i.e. discovering useful knowledge from unstructured text, is becoming an increasingly important aspect of KDD.

Much of the work in text mining does not exploit any form of natural-language processing (NLP), treating documents as an unordered “bag of words” as is typical in information retrieval. The standard a vector space model of text represents a document as a sparse vector that specifies a weighted frequency for each of the large number of distinct words or tokens that appear in a corpus. Such a simplified representation of text has been shown to be quite effective for a number of standard tasks such as document retrieval, classification, and clustering however, most of the knowledge that might be mined from text cannot be discovered using a simple bag-of-words representation. The entities referenced in a document and the properties and relationships asserted about and between these entities cannot be determined using a standard vector-space representation. Although full natural-language understanding is still far from the capabilities of current technology, existing methods in information extraction (IE) are, with reasonable accuracy, able to recognize several types of entities in text and identify some relationships that are asserted between them. Therefore, IE can serve an important technology for text mining. If the knowledge to be discovered is expressed directly in the documents to be mined, then IE alone can serve as an effective approach to text mining. However, if the documents contain concrete data in unstructured form rather than abstract knowledge, it may be useful to first use IE to transform the unstructured data in the document corpus into a structured database, and then use traditional data mining tools to identify abstract patterns in this extracted data. In this article, we review these two approaches to text mining with information extraction, using one of our own research projects to illustrate each approach. First, we introduce the basics of information extraction. Next, we discuss using IE to directly extract knowledge from text. Finally, we discuss discovering knowledge by mining data that is first extracted from unstructured or semi-structured text.

2. INFORMATION EXTRACTION

2.1 IE Problems

Information Extraction (IE) concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from unstructured text. One type of IE, named entity recognition, involves identifying references to particular kinds of objects such as names of people, companies, and locations. Fig. 1 shows part of a sample abstract in which the protein names are underlined. In addition to recognizing entities, an important problem is extracting specific types of relations between entities. For example, in newspaper text, one can identify that an organization is located in a particular city or that a person is affiliated with a specific organization. In biomedical text, one can identify that a protein interacts with another protein or that a protein is located in a particular part of the cell. For example, identifying protein interactions in the abstract excerpt in Fig 1 would require extracting the relation: interacts (NOSIP, eNOS).

Sample Job Posting:

Job Title: Senior DBMS Consultant

Location: Dallas,TX

Responsibilities:

DBMS Applications consultant works with project teams to define DBMS based solutions that support the enterprise deployment of Electronic Commerce, Sales Force Automation, and Customer Service applications.

Desired Requirements:

3-5 years exp. developing Oracle or SQL Server apps using

Visual Basic, C/C++, PowerBuilder, Progress, or similar.Recent experience related to installing and configuring Oracle or SQL Server in both dev. and deployment environments.

Desired Skills:

Understanding of UNIX or NT, scripting language. Know principles of structured software engineering and project management

Filled Job Template:

title: Senior DBMS Consultant

state: TX

city: Dallas

country: US

language: Powerbuilder, Progress, C, C++, Visual Basic

platform: UNIX, NT

application: SQL Server, Oracle

area: Electronic Commerce, Customer Service

required years of experience: 3

desired years of experience: 5

Figure 1: Sample Job Posting and Filled Template

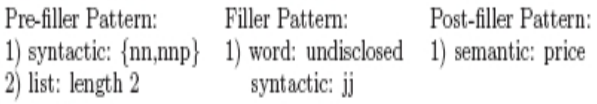
IE can also be used to extract fillers for a predetermined set of slots (roles) in a particular template (frame) relevant to the domain. Fig 1 shows a sample message from the newsgroup and the filled computer-science job template where several slots may have multiple fillers. For example, slots such as languages, platforms, applications, and areas usually have more than one filler, while slots related to the job's title or location usually have only one filler.

Similar applications include extracting relevant sets of predefined slots from university colloquium announcements or apartment rental ads. Another application of IE is extracting structured data from unstructured or semi-structured web pages. When applied to semi-structured HTML, typically generated from an underlying database by a program on a web server, an IE system is typically called a wrapper and the process is number, publisher, and price of book from an Amazon web page. IE systems can also be used to extract data or knowledge from less-structured web sites by using both the HTML text in their pages as well as the structure of the hyperlinks between their pages.

2.2 IE Methods

There are a variety of approaches to constructing IE systems. One approach is to manually develop information-extraction rules by encoding patterns (e.g. regular expressions) that reliably identify the desired entities or relations. However, due to the variety of forms and contexts in which the desired information can appear, manually developing patterns is very difficult and tedious and rarely results in robust systems. Consequently, supervised machine-learning methods trained on human annotated corpora has become the most successful approach to developing robust IE systems. A variety of learning methods have been applied to IE.

One approach is to automatically learn pattern-based extraction rules for identifying each type of entity or relation. In Wrapper induction and Boosted Wrapper Induction (BWI), regular-expression type patterns are learned for identifying the beginning and ending of extracted phrases. Inductive Logic programming (ILP) has also been used to learn logical rules for identifying phrases to be extracted from a document. An alternative general approach to IE is to treat it as a sequence labeling task in which each word (token) in the document is assigned a label (tag) from a fixed set of alternatives. For example, for each slot, X, to be extracted, we include a token label BeginX to mark the beginning of a filler for X and InsideX to mark other tokens in a filler for X. Finally, we include the label other for tokens that are not included in the filler of any slot. Given a sequence labeled with these tags, it is easy to extract the desired fillers. One approach to the resulting sequence labeling problem is to use a statistical sequence model such as a Hidden Markov Model (HMM) or a Conditional Random Field (CFR). Several earlier IE systems used generative HMM mod- els; however, discriminately-trained CRF models have recently been shown to have an advantage over HMM's. In both cases, the model parameters are learned from sometimes referred to as screen scraping. A typical application is extracting data on commercial items from web stores for a comparison shopping agent (shopbot) such as MySimon (www.mysimon.com) or Froogle (froogle.google.com).



Pre-filler Pattern:	Filler Pattern:	Post-filler Pattern:
1) syntactic: {nn,nnp}	1) word: undisclosed	1) semantic: price
2) list: length 2	syntactic: jj	

Figure 3: Sample Extraction Rule Learned by RAPIER

For example, a wrapper may extract the title, author, ISBN a supervised training corpus and then an efficient dynamic programming method based on the Viterbi algorithm is used to determine the most probable tagging of a complete test document. Another approach to the sequence labeling problem for IE is to use a standard feature-based inductive classifier to predict the label of each token based on both the token itself and its surrounding context. Typically, the context is represented by a set of features that include the one or two tokens on either side of the target token as well as the labels of the one or two preceding tokens (which will already have been classified when labeling a sequence from left to right). Using this general approach, IE systems have been developed that use many different trained classifiers such as decision trees, boosting and memory-based learning (MBL), support-vector machines (SVMs), maximum entropy (MaxEnt), transformation-based learning (TBL) and many others.Many IE systems simply treat text as a sequence of uninterrupted tokens; however, many others use a variety of other NLP tools or knowledge bases. For example, a number of systems pre-process the text with a part-of-speech (POS) tagger and use words' POS (e.g. noun, verb, and adjective) as an extra feature that can be used in handwritten patterns, learned extraction rules, or induced classifiers. Several IE systems use phrase chunkers (to identify potential phrases to extract. Others use complete syntactic parsers, particularly those which try to extract relations between entities by examining the syntactic relationship between the phrases describing the relevant entities. Some use lexical semantic databases, such asWordNet, which provide word classes that can be used to define more general extraction patterns. As a sample extraction pattern, Fig 3 shows a rule for extracting the transaction amount from a newswire concerning a corporate acquisition. This rule extracts the value “undisclosed” from phrases such as “sold to the bank for an undisclosed amount” or “paid Honeywell an undisclosed price”. The pre-filler pattern matches a noun or proper noun (indicated by the POS tags ’nn’ and ’pn’, respectively) followed by at most two other unconstrained words. The filler pattern matches the word “undisclosed” only when its POS tag is “adjective.” The post-filler pattern matches any word in WordNet's semantic class named“price”.

3. EXTRACTING KNOWLEDGE

If the information extracted from a corpus of documents represents abstract knowledge rather than concrete data, IE itself can be considered a form of “discovering” knowledge from text. For example, an incredible wealth of biological knowledge is stored in published articles in scientific journals. Summaries of more than 11 million such articles are available in the Medline database;1 however, retrieving and processing this knowledge is very difficult due to the lack of formal structure in the natural-language narrative in these documents. Automatically extracting information from biomedical text holds the promise of easily consolidating large amounts of biological knowledge in computer-accessible form. IE systems could potentially gather information on global gene relationships, gene functions, protein interactions, gene-disease relationships, and other important information on biological processes. Consequently, a growing number of recent projects have focused on developing IE systems for biomedical literature. CRF's capture the dependence between the labels of adjacent words; it does not adequately capture long-distance dependencies between potential extractions in different parts of a document. For example, in our protein-tagging task, repeated references to the same protein are common. If the context surrounding one occurrence of a phrase is very indicative of it being a protein, then this should also influence the tagging of another occurrence of the same phrase in a different context which is not typical of protein references. Therefore, we studied a new IE method based on Relational Markov Networks (RMN's) that captures dependencies between distinct candidate extractions in a document. Experimental evaluation confirmed that this approach increases accuracy of human-protein recognition compared to a traditional CRF.

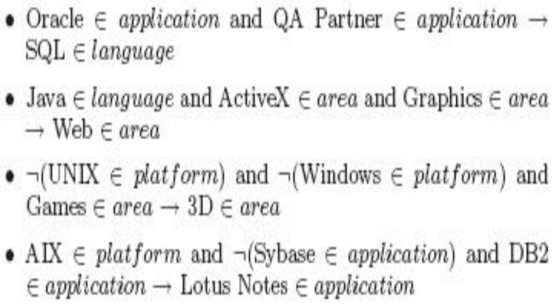
We have also evaluated several approaches to extracting protein interactions from text in which protein names have already been identified manually developed patterns for extracting interacting proteins, where a pattern is a sequence of words (or POS tags) and two protein-name tokens. Between every two adjacent words is a number indicating the maximum number of arbitrary words that can be skipped at this position. Below is a sample pattern that it learned for extracting protein interactions: - (7) interactions (0) between (5) PROT (9) PROT (17) .where PROT matches a previously tagged protein name. The induced patterns were able to identify interactions more precisely than the human-written ones.

Another approach we have taken to identifying protein interactions is based on co-citation. This approach does not try to find specific assertions of interaction in the text, but rather exploits the idea that if many different abstracts reference both protein A and protein B, then A and B are likely to interact. Particularly, if two proteins are co-cited significantly more often than one would expect if they were cited independently at random, then it is likely that they interact. In order to account for the case where the co-citing abstracts do not actually concern protein interactions but cite multiple proteins for other reasons, we also used a Bayesian “bag of words” text classifier trained to discriminate between abstracts that discuss protein interactions from those that do not. In order to find interactions, protein pairs that are highly co-cited were filtered for those which are specifically co-cited in abstracts that the Bayesian text-classifier assigns high-probability of discussing protein interactions. Using these techniques, we recently completed the initial phase of a large-scale project to mine a comprehensive set of human protein interactions from the biomedical literature. By mining 753,459 human-related abstracts from Medline with a combination of a CRF-based protein tagger, co citation analysis, and automatic text classification, we extracted a set of 6,580 interactions between 3,737 proteins. By utilizing information in existing protein databases, this automatically extracted data was found to have accuracy comparable to manually developed data sets. Based on comparisons to these existing protein databases, the co citation plus text-classification approach was found to be more effective at identifying interactions than our IE approach based on ELCS. By consolidating our text-mined knowledge with existing manually-constructed biological databases, we have assembled a large, fairly comprehensive, database of known human protein interactions containing 31,609 interactions amongst 7,748 proteins. More details on our database of protein interactions have been published in the biological literature and it is freely available on the web.2 Therefore, using automated text mining has helped build an important knowledge base of human proteins that has been recognized as a contribution worthy of publication in Genome Biology and will hopefully become a valuable resource to biologists.

4. MINING EXTRACTED DATA

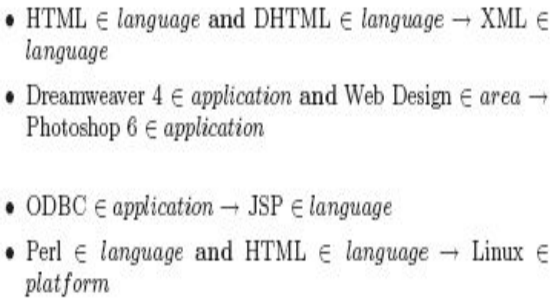
If extracted information is specific data rather than abstract knowledge, an alternative approach to text mining is to first use IE to obtain structured data from unstructured text and then use traditional KDD tools to discover knowledge from this extracted data. Using this approach, we studied a text-mining system called DiscoTEX (Discovery from Text Extraction) which has been applied to mine job postings and resumes posted to USENET newsgroups as well as Amazon book-description pages spidered from the web. In DiscoTEX, IE plays the important role of preprocessing a corpus of text documents into a structured database suitable for mining. DiscoTEX uses two learning systems to build extractors. By training on a corpus of documents annotated with their filled templates, these systems acquire pattern-matching rules that can be used to extract data from novel documents.

After constructing an IE system that extracts the desired set of slots for a given application, a database can be constructed from a corpus of texts by applying the extractor to each document to create a collection of structured records. Standard KDD techniques can then be applied to the resulting database to discover interesting relationships. Specifically, DiscoTEX induces rules for predicting each piece of information in each database field given all other information in a record.



- Oracle $\in$ *application* and QA Partner $\in$ *application* $\rightarrow$ SQL $\in$ *language*
- Java $\in$ *language* and ActiveX $\in$ *area* and Graphics $\in$ *area* $\rightarrow$ Web $\in$ *area*
- $\neg$(UNIX $\in$ *platform*) and $\neg$(Windows $\in$ *platform*) and Games $\in$ *area* $\rightarrow$ 3D $\in$ *area*
- AIX $\in$ *platform* and $\neg$(Sybase $\in$ *application*) and DB2 $\in$ *application* $\rightarrow$ Lotus Notes $\in$ *application*

Figure 4: Sample rules mined from CS job postings.



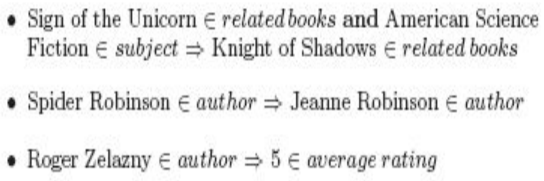
- HTML $\in$ *language* and DHTML $\in$ *language* $\rightarrow$ XML $\in$ *language*
- Dreamweaver 4 $\in$ *application* and Web Design $\in$ *area* $\rightarrow$ Photoshop 6 $\in$ *application*
- ODBC $\in$ *application* $\rightarrow$ JSP $\in$ *language*
- Perl $\in$ *language* and HTML $\in$ *language* $\rightarrow$ Linux $\in$ *platform*

Figure 5: Sample rules mined from CS resumés.

In order to discover prediction rules, we treat each slot-value pair in the extracted database as a distinct binary feature, such as “graphics 2 area”, and learn rules for predicting each feature from all other features. We have applied C4.5 rules to discover interesting rules from the resulting binary data. Discovered knowledge describing the relationships between slot values is written in the form of production rules. If there is a tendency for “Web” to appear in the area slot when “Director” appears in the applications slot, this is represented by the production rule, “Director 2 application! Web 2 area”. Sample rules that C4.5rules mined from a database of 600 jobs that Rapier extracted from the USENET newsgroup austin.jobs are shown in Figure 4. The last rule illustrates the discovery of an interesting concept which could be called “the IBM shop;” i.e. companies that require knowledge of an IBM operating system and DBMS, but not a competing DBMS, also require knowledge of Lotus Notes, another IBM product.

We also applied Ripper and Apriori to discover interesting rules from extracted data. Sample rules mined from a database of 600 resum´es extracted from the USENET newsgroup misc.jobs.resumes by BWI are shown in Figure 5. The first two rules were discovered by Ripper while the last two were found by Apriori. Since any IE or KDD module can be plugged into the DiscoTEX system, we also used an information extractor (wrapper) manually developed for a book recommending system to find interesting patterns in a corpus of book descriptions. Sample association rules mined from a collection of 1,500 science fiction book descriptions from the online Amazon.com bookstore are shown in Figure 6.

Unfortunately, the accuracy of current IE systems is limited, and therefore an automatically extracted database will inevitably contain a fair number of errors



- Sign of the Unicorn $\in$ *related books* and American Science Fiction $\in$ *subject* $\Rightarrow$ Knight of Shadows $\in$ *related books*
- Spider Robinson $\in$ *author* $\Rightarrow$ Jeanne Robinson $\in$ *author*
- Roger Zelazny $\in$ *author* $\Rightarrow$ 5 $\in$ *average rating*

Figure 6: Sample rules mined from book descriptions.

.

An important question is whether the knowledge discovered from a “noisy” extracted database is significantly less reliable than knowledge discovered from a “clean” manually-constructed database. We have conducted experiments on job postings showing that rules discovered from an automatically extracted database are very close in accuracy to those discovered from a corresponding manually-constructed database [49]. These results demonstrate that mining extracted data is a reliable approach to discovering accurate knowledge from unstructured text .Another potential problem with mining extracted data is that the heterogeneity of extracted text frequently prevents traditional data-mining algorithms from discovering useful knowledge. The strings extracted to fill specific data fields can vary substantially across documents even though they refer to the same real-world entity. For example, the Microsoft operating system may be referred to as “Windows,” “Microsoft Windows,” “MS Windows,” etc.. We developed two approaches to addressing this problem [47]. One approach is to first “clean” the data by identifying all of the extracted strings that refer to the same entity and then replacing sets of equivalent strings with canonical entity names. Traditional KDD tools can then be used to mine the resulting “clean” data. Identifying textually distinct items that refer to the same entity is an instance of the general database “deduping” or record-linkage problem. Another approach to handling heterogeneity is to mine “soft matching” rules directly from the “dirty” data extracted from text. In this approach, the rule induction process is generalized to allow partial matching of data strings in order to discover important regularities in variable textual data.

5. CONCLUSIONS

In this paper we have discussed two approaches to using natural-language information extraction for text mining. First, one can extract general knowledge directly from text. As an example of this approach, we reviewed project which extracted a knowledge base of 6,580 human protein interactions by mining over 750,000 Medline abstracts. Second, one can first extract structured data from text documents or web pages and then apply traditional KDD methods to discover patterns in the extracted data. By exploiting the latest techniques in human-language technology and computational linguistics and combining them with the latest methods in machine learning and traditional data mining, one can effectively mine useful and important knowledge from the continually growing body of electronic documents and web pages.

REFERENCES

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

[2] U. Y. Nahm and R. J. Mooney. Mining soft-matching association rules. In Proceedings of the Eleventh International Conference on Information and Knowledge Management (CIKM-2002), pages 681–683, McLean,VA, Nov. 2002

[3] U. Y. Nahm, M. Bilenko, and R. J. Mooney. Two approaches to handling noisy variation in text mining. In Papers from the Nineteenth International Conference on Machine Learning (ICML-2002) Workshop on Text learning, pages 18–27, Sydney, Australia, July 2002.

[4] L. A. Ramshaw and M. P. Marcus. Text chunking using transformation-based learning. In Proceedings of the Third Workshop on Very Large Corpora, 1995.

[5] R. Baeza-Yates and B. Ribeiro-Neto. Modern Informa tion Retrieval. ACM Press, New York, 1999.

[6] S. Chakrabarti. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, San Francisco, CA, 2002.

[7] R. Ghani, R. Jones, D. Mladeni´c, K. Nigam, and S. Slattery. Data mining on symbolic knowledge extracted from the Web. In D. Mladeni´c, editor, Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (KDD-2000) Workshop on Text Mining, pages 29–36, Boston, MA, Aug.2000