A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques

[1] R. Sagayam, [2] S.Srinivasan, [3] S. Roshni
[1, 2, 3] Department Of Computer Science
Govt. Arts College (Autonomous)
Salem-7
[2] Periyar University
Salem-636011

Abstract
Text mining is the analysis of data contained in natural language text. Text mining works by transposing words and phrases in unstructured data into numerical values which can then be linked with structured data in a database and analyzed with traditional data mining techniques. Text information retrieval and data mining has thus become increasingly important. In this paper a survey of Text mining have been presented.

Keyword: Information Retrieval, Information Extraction and Indexing Techniques

1. Introduction

Text mining is a variation on a field called data mining, that tries to find interesting patterns from large databases. Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web . Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases. Data stored in most text databases are semi structured data in that they are neither completely unstructured nor completely structured. For example, a document may contain a few structured fields, such as title, authors, publication date, category, and so on, but also contain some largely unstructured text components, such as abstract and contents. There have been a great deal of studies on the modeling and implementation of semi structured data in recent database research. Moreover, information retrieval techniques, such as text indexing methods, have been developed to handle unstructured documents. Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Typically, only a small fraction of the many available documents will be relevant to a given individual user. Without knowing what could be in the documents, it

Is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users need tools to compare different documents, rank the importance And relevance of the documents, or find patterns and trends across multiple documents. Thus, text mining has become an increasingly popular and essential theme in data mining.

2. Information Retrieval

Information retrieval is a field that has been developing in parallel with database systems for many years. Unlike the field of database systems, which has focused on query and transaction processing of structured data, information retrieval is concerned with the organization and retrieval of information from a large number of text-based documents. Since information retrieval and database systems each handle different kinds of data, some database system problems are usually not present in information retrieval systems, such as concurrency control, recovery, transaction management, and update. Also, some common information retrieval problems are usually not encountered in traditional database systems, such as unstructured documents, approximate search based on keywords, and the notion of relevance. Due to the abundance of text information, information retrieval has found many applications. There exist many information retrieval systems, such as on-line library catalog systems, on-line document management systems, and the more recently developed Web search engines. A typical information retrieval problem is to locate relevant documents in a document collection based on a user's query, which is often some keywords describing an information need, although it could also be an example relevant document. In such a search problem, a user takes the initiative to "pull" the relevant information out from the collection; this is most appropriate when a user has some ad hoc information need, such as finding information to buy a used car. When a user has a long-term information need , a retrieval system may also take the initiative to "push" any newly arrived information item to a user if the item is judged as being relevant to the user's information need. Such an information access process is called information filtering, and the corresponding systems are often called filtering systems or recommender systems. From a technical viewpoint, however, search and filtering share many common techniques. Below we briefly discuss the major techniques in information retrieval with a focus on search techniques.

2.1. MEASURES FOR TEXT RETRIEVAL

The set of documents relevant to a query be denoted as *{*Relevant*}*, and the set of documents retrieved be denoted as {Retrieved}. The set of documents that are both relevant and retrieved is denoted as {Relevant }∩{Retrieved}, as shown in the Venn diagram of Figure 1. There are two basic measures for assessing the quality of text retrieval. Precision: This is the percentage of retrieved documents that are in fact relevant to the query . It is formally defined as

Precision=|{Relevant}∩{Retrieved}|/|{Retrieved}|

Recall: This is the percentage of documents that are relevant to the query and were, in fact, retrieved. It is formally defined as

Recall=|{Relevant}∩{Retrieved}|/|{Relevant} |

An information retrieval system often needs to trade off recall for precision or vice versa. One commonly used trade-off is the F-score, which is defined as the harmonic mean of recall and precision

F-score =recall×precision/(recall+ precision)/2

The harmonic mean discourages a system that sacrifices one measure for another too drastically

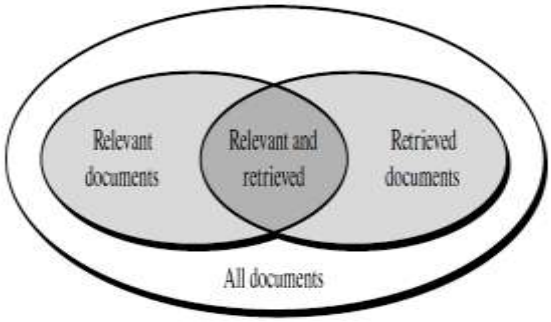

Figure 1. Relationship between the set of relevant documents and the set of retrieved documents.

Precision, recall, and F-score are the basic measures of a retrieved set of documents. These three measures are not directly useful for comparing two ranked lists of documents because they are not sensitive to the internal ranking of the documents in a retrieved set. In order to measure the quality of a ranked list of documents, it is common to compute an average of precisions at all the ranks where a new relevant document is returned. It is also common to plot a graph of precisions at many different levels of recall; a higher curve represents a better-quality information retrieval system. For more details about these measures, readers may consult an information retrieval textbook, such as [3].

3. Text Indexing Techniques

There are several popular text retrieval indexing techniques, including inverted indices and signature files. An inverted index is an index structure that maintains two hash indexed or B+-tree indexed tables: document table and term table, where document table consists of a set of document records, each containing two fields: doc id and posting list, where posting list is a list of terms (or pointers to terms) that occur in the document, sorted according to some relevance measure. term table consists of a set of term records, each containing two fields: term id and posting list, where posting list specifies a list of document identifiers in which the term appears. With such organization, it is easy to answer queries like "Find all of the documents associated with a given set of terms," or "Find all of the terms associated with a given set of documents." For example, to find all of the documents associated with a set of terms, we can first find a list of document identifiers in term table for each term, and then intersect them to obtain the set of relevant documents. Inverted indices are widely used in industry. They are easy to implement. The posting lists could be rather long, making the storage requirement quite large. They are easy to implement, but are not satisfactory at handling synonymy (where two very different words can have the same meaning) and polysemy (where an individual word may have many meanings). A signature file is a file that stores a signature record for each document in the database. Each signature has a fixed size of *b* bits representing terms. A simple encoding scheme goes as follows. Each bit of a document signature is initialized to 0. A bit is set to 1 if the term it represents appears in the document. A signature *S*1 matches another signature *S*2 if each bit that is set in signature *S*2 is also set in *S*1. Since there are usually more terms than available bits, multiple terms may be mapped into the same bit. Such multiple-to one mappings make the search expensive because a document that matches the signature of a query does not necessarily contain the set of keywords of the query. The document has to be retrieved, parsed, stemmed, and checked. Improvements can be made by first performing frequency analysis, stemming, and by filtering stop words, and then using a hashing technique and superimposed coding technique to encode the list of terms into bit representation. Nevertheless, the problem of multiple-to-one mappings still exists, which is the major disadvantage of this approach. Readers can refer to [2] for more detailed discussion of indexing techniques, including how to compress an index.

4. Query Processing Techniques

Once an inverted index is created for a document collection, a retrieval system can answer a keyword query quickly by looking up which documents contain the query keywords. Specifically, we will maintain a score accumulator for each document and update these accumulators as we go through each query term. For each query term, we will fetch all of the documents that match the term and increase their scores. More sophisticated query processing techniques are discussed in [2].When examples of relevant documents are available, the system can learn from such examples to improve retrieval performance. This is called relevance feedback and has proven to be effective in improving retrieval performance. When we do not have such relevant examples, a system can assume the top few retrieved documents in some initial retrieval results to be relevant and

extract more related keywords to expand a query. Such feedback is called pseudo-feedback or blind feedback and is essentially a process of mining useful keywords from the top retrieved documents. Pseudo-feedback also often leads to improved retrieval performance. One major limitation of many existing retrieval methods is that they are based on exact keyword matching. However, due to the complexity of natural languages, keyword based retrieval can encounter two major difficulties. The first is the synonymy problem: two words with identical or similar meanings may have very different surface forms. For example, a user's query may use the word "automobile," but a relevant document may use "vehicle" instead of "automobile." The second is the polysemy problem: the same keyword, such as mining, or Java, may mean different things in different contexts.

5. Information Extraction

The general purpose of Knowledge Discovery is to "extract implicit, previously unknown, and potentially useful information from data". Information Extraction IE mainly deals with identifying words or feature terms from within a textual file. Feature terms can be defined as those which are directly related to the domain.

Figure 2. A layered model of the Text Mining Application

These are the terms which can be recognized by the tool. In order to perform this function optimally, we had to look into few more aspects which are as follows:

5.1 Stemming

Stemming refers to identifying the root of a certain word. There are basically two types of stemming techniques, one is inflectional and other is derivational. Derivational stemming can create a new word from an existing word, sometimes by simply changing grammatical category (for example, changing a noun to a verb) [Wikipedia]. The type of stemming we were able to implement is called Inflectional Stemming. A commonly used algorithms is the 'Porter's Algorithm' for stemming. When the normalization is confined to regularizing grammatical variants such as singular/plural or past/present, it is referred to inflectional stemming [10].To minimize the effects of inflection and morphological variations of Words (stemming), our approach has pre-processed each word using a provided version of the Porter stemming algorithm with a few changes towards the end in which we have omitted some cases.

e.g. apply – applied – applies

print – printing – prints – printed

In both the cases, all words of the first example will be treated as 'apply' and all words of the second example will be treated as 'print'.

5.2 Domain dictionary

In order to develop tools of this sort, it is essential to provide them with a knowledge base. A collective set of all the 'feature terms' is the Domain dictionary (our source was www.webopedia.com). The structure of the Domain dictionary which we implemented consisted of three levels in the hierarchy. Namely, Parent Category, Sub-category and word. Parent categories define the main category under which any sub-category or word falls. A parent category will be unique on its level in the hierarchy. Sub-categories will belong to a certain parent category and each sub-category will consist of all the words associated with it. As an example, consider the following

Table 1.Structure of the Domain Dictionary

Parent Category	Sub-Category	Words
Hardware	Data Storage	Grabber
	Input devices	Light pen
	Modems	Joystick
	Motherboards	Contact image sensor
	Networking	Digital camera

Table 1 is an example that shows how we identify words which belong to the Parent Category 'Hardware' and Sub-category 'Input Devices'.

5.3 Exclusion List

A lot of words in a text file can be treated as unwanted noise. To eliminate these, we devised a separate file which includes all such words. These include words such as the, a, an, if, off, on etc.

6. Research Directions

With abundant literature published in research into frequent pattern mining, one may wonder whether we have solved most of the critical problems related to frequent pattern mining so that the solutions provided are good enough for most of the data mining tasks. However, based on our view, there are still several critical research problems that need to be solved before frequent pattern mining can become a cornerstone approach in data mining applications. First, the most focused and extensively studied topic in frequent pattern mining is perhaps scalable mining methods. The set

of frequent patterns derived by most of the current pattern mining methods is too huge for effective usage. There are proposals on reduction of such a huge set, including closed patterns, maximal patterns, approximate patterns, condensed pattern bases, representative patterns, clustered patterns, and discriminative frequent patterns, as introduced in the previous sections. However, it is still not clear what kind of patterns will give us satisfactory pattern sets in both compactness and representative quality for a particular application, and whether we can mine such patterns directly and efficiently. Much research is still needed to substantially reduce the size of derived pattern sets and enhance the quality of retained patterns. Frequent pattern mining: current status and future directions. Second, although we have efficient methods for mining precise and complete set of frequent patterns, approximate frequent patterns could be the best choice in many applications. For example, in the analysis of DNA or protein sequences, one would like to find long sequence patterns that approximately match the sequences in biological entities, similar to BLAST. Much research is still needed to make such mining more effective than the currently available tools in bioinformatics. Third, to make frequent pattern mining an essential task in data mining, much research is needed to further develop pattern-based mining methods. For example, classification is an essential task in data mining. Fourth, we need mechanisms for deep understanding and interpretation of patterns, e.g., semantic annotation for frequent patterns, and contextual analysis of frequent patterns. The main research work on pattern analysis has been focused on pattern composition (e.g., the set of items in item-set patterns) and frequency. The semantic of a frequent pattern includes deeper information: what is the meaning of the pattern; what are the synonym patterns; and what are the typical transactions that this pattern resides? In many cases, frequent patterns are mined from certain data sets which also contain structural information. Finally, applications often raise new research issues and bring deep insight on the strength and weakness of an existing solution. This is also true for frequent pattern mining. On one side, it is important to go to the core part of pattern mining algorithms, and analyze the theoretical properties of different solutions. On the other side, although we only cover a small subset of applications in this article, frequent pattern mining has claimed a broad spectrum of applications and demonstrated its strength at solving some problems. Much work is needed to explore new applications of frequent pattern mining. For example, bioinformatics has raised a lot of challenging problems, and we believe frequent pattern mining may contribute a good deal to it with further research efforts.

7. Conclusion

Most of knowledge hidden in electronic media of an organization is encapsulated in documents. Acquiring this knowledge implies effective querying of the documents as well as the combination of information pieces from different textual sources (e.g.: the World Wide Web). Discovering such hidden know ledge is an essential requirement for many corporations, due to its wide spectrum of applications. In this short survey, the notion of text mining have been introduced and several techniques available have been presented. Due to its novelty, there are many potential research areas in the field of Text Mining, which includes finding better intermediate forms for representing the outputs of information extraction, an XML document may be a good choice. Mining texts in different languages is a major problem, since text mining tools should be able to work with many languages and multilingual documents. Integrating a domain knowledge base with a text mining engine would boost its efficiency, especially in the information retrieval and information extraction phases.

References

[1] M. A. Hearst. What is text mining? http://www.sims.berkeley.edu/~hearst/text-mining.html, Oct. 2003.

[2] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[3] R.Baeza-Yates and B.Ribeiro-Neto, Modern Information Retrieval addition-Wesley,Boston,1999

[4] Jochen Dorre, Peter Gersti, Roland Seiffert (1999), Text Mining: Finding Nuggets in Mountains of Textual Data, ACM KDD 1999 in San Diego, CA, USA.

[5] Ah-Hwee Tan, (1999), Text Mining: The state of art and the challenges, In proceedings, PAKDD'99 Workshop on Knowledge discovery from Advanced Databases (KDAD'99), Beijing, pp. 71-76, April 1999.

[6] Danial Tkach, (1998), Text Mining Technology Turning Information Into Knowledge A white paper from IBM .

[7]. Helena Ahonen, Oskari Heinonen, Mika Klemettinen, A. Inkeri Verkamo, (1997), Applying Data Mining Techniques in Text Analysis, Report C-1997-23, Department of Computer Science, University of Helsinki, 1997

[8]. Mark Dixon,(1997), An Overview of Document Mining Technology, http://www.geocities.com/ResearchTriangle/Thinktank/1997/mark/writings/dixm 97_dm.ps

[9] Juan José García Adeva and Rafael Calvo, "Mining Text with Pimiento", University of Sydney

[10] Text Mining Application Programming by Manu Konchadi. Published by Charles River Media. ISBN: 1584504609

[11] Automated Concept Extraction From Plain Text. Boris, GARAGe Michigan State University, East Lansing MI 48824.

[12] Dr. Antoine Spinakis; Asanoula Chatzimakri, "Comparison Study of Text Mining Tools".

[13] Raymond J. Mooney and Razvan Bunescu, "Mining Knowledge from Text Using Information Extraction", University of Texas at Austin

[14] Tova, Milo, "Active views for Electronic commerce". Paper number: EUROPE64.