TEXT MINING: PRINCIPLES AND APPLICATIONS

John A. Atkinson A

RESUMEN

En este trabajo, se exploran algunas de las posibilidades que ofrece Data Mining sobre información textual electrónica. Se describen algunas de las posibilidades del análisis automático de texto o Text Mining, que surge debido a que los textos representan un tipo de información rica en conocimiento pero a la vez no estructurada, lo que dificulta su decodificación por medios automáticos. Se describen algunos de los aspectos fundamentales tras la problemática y se delinean algunas de las ideas principales de cómo llevar a cabo este tipo de sistemas. Finalmente, se discuten algunos de los enfoques actuales, sus características y aplicaciones.

ABSTRACT

In this paper, some of the many possibilities of Data Mining on text collection are explored. The possibilities for automatic analysis of text or Text Mining are untapped. Text represents a kind of unstructured and at the same time, rich information which is difficult to decode by automatic means. Some background on this problem is described and the main ideas about how to pursue Text Mining are briefly outlined. Some of the main approaches, common architectural features and their applications and uses are discussed too.

INTRODUCTION

Knowledge Discovery in Databases (KDD) -or Data Mining (DM), focuses on the computerized exploration of large amounts of data and on the discovery of interesting patterns within them. Thus, DM is defined as “The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” [3], [13].

While most work on DM has been concerned with structured databases, there has been little work on handling the huge amounts of information that is available only in unstructured textual form. A table showing a classification of DM and TM applications can be seen in Fig. 1. These can be divided into two groups: those where techniques are applied to find some fixed patterns in data and those where it is expected to discover new knowledge, relationship or associations on data. As an analogy to find “gold” in mines, the latter is called Nuggets. Even though it is not a real case in DM, it presents at least a good starting point to realize where we are going. If patterns are to be found in structured databases (non-textual) then standard DM techniques are applied. Otherwise, it is necessary a completely different kind of technique to handle the textual information (corpus linguistics). As far, the type of techniques to be applied is clear enough, however, problems go up when it is required to discover new knowledge from textual data. It could be chosen to apply standard corpus linguistics but unfortunately, there is not fixed patterns to match, instead, we have to find and discover these ones, to analyse them and associate them.

Thus, specialized techniques specifically operating on textual data become necessary to extract information from such kind of collection of texts. These techniques are gathered under the name of Text Mining (TM) and, in order to discover and use the implicit structure of the texts (eg. grammar structure), they may integrate some specific Natural Language Processing (Computational Corpus Linguistics) [4], [7], [8], [9].

TM applications impose strong constraints on the usual NLP tools: as they involve large volumes of textual data, they do not allow to integrate complex treatments, semantic models for the application domains are rarely

---

1 Institute for Communicating and Collaborative Systems, School of Artificial Intelligence, University of Edinburgh, Edinburgh, Scotland, UK. e-mail: atkinson@dai.ed.ac.uk
available, and this implies strong limitations on the sophistication of the semantic and pragmatic level of the linguistic models [7], [12].

<table>
<thead>
<tr>
<th>Finding Pattern</th>
<th>Finding Nuggets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Textual</td>
<td>Novel</td>
</tr>
<tr>
<td>Stack Data Mining</td>
<td>Database queries</td>
</tr>
<tr>
<td>Textual/Textual</td>
<td>Non-novel</td>
</tr>
<tr>
<td>Conceptual/Corpus Linguistics</td>
<td>Red TM</td>
</tr>
<tr>
<td>Information Retrieval</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 1.- DM versus TM Applications

TM shares many characteristics with classical DM, but also differs in many ways. On the other side, many classical DM algorithms, such as value prediction or decision trees, are irrelevant or ill suited for the textual applications because they rely on the structuring of data, the availability of many information and so on [4], [5], [7], [12], [13]. On the other hand, there are special mining tasks, such as concept relationship analysis which are unique to TM.

Thus, Text Mining is defined as: “The knowledge-discovery process which looks for identifying and analyzing useful information on data which is interesting to users from big amounts of textual data”. From this perspective, Information Extraction (IE) [2], [9], [10] and Text Mining may be complementary tasks but they differ in many ways. While IE relies in matching some fixed patterns to get the required information from the text and then translate it into a structured media (ie. database, templates, etc), TM or Information Analysis [1] relies in the fact that the value of the raw information comes from the competence can be got to analyze and to produce “elaborated” information. This is, both a high level information/knowledge which could be implicit but not present on data, and a potentially useful information/knowledge for the decision making process in a certain activity domain.

Therefore, the tasks of TM on textbases could include the following:

- To cluster knowledge/information into charts/maps.
- To summarize information.
- To identify hidden structures between groups of objects.
- To extract hidden associations between elements on the texts.
- To provide an overview of the contents of a large document collection.
- To categorize texts by discovering relevant groupings.

TEXT MINING TECHNIQUES

There are different techniques and methods for TM in order to find new structures, patterns, or associations. Some TM has involved the assumption of an a priori categorization (preprocessing) into attributes and then proceeded via “classical” DM methods, i.e. statistical analysis, associations, etc [5], [12]. Others, investigate the full text of document collection, e.g. categorization used above, or purely analytical results.

A common end-goal of much TM is a more efficient, complete, and/or specific way to browse and search large collections of documents. Thus, the main techniques in TM can be divided according to the tasks they perform in the discovery process: the kind of information they extract and the kind of analysis/association done with them:

KINDS OF INFORMATION EXTRACTED:

1. **Labels**: it has been assumed that associated with each document is a set of labels and knowledge-discovery operations are performed on the labels of each document. In general, it can be assumed that labels correspond to keywords, each of which represents that a given document is about a topic associated with that keyword [6,9].

2. **Words**: in which a document is assumed to be labeled with each of the words that occurs within it.

3. **Terms**: in which for each document is found word sequences that are likely to have meaning in the domain, and then mining is performed on the extracted terms labeling each document. The advantage of this method is that the extracted terms are fewer in number and tend to represent the important information on the text than the previous approaches [4,5].
KIND OF ASSOCIATIONS:

Regular Associations: some known algorithms (in a very simplified view) assume that the textual data is indexed (manually or automatically) with the help of NLP techniques. The indexing structures can be used as a basis for the actual knowledge discovery process. In order to generate association rules, they operate in two phases [12]. Given a set of keywords $A$ and a collection of indexed documents $T$, the extraction of associations satisfying, given some support and confidence constraints is performed.

Concept Hierarchies: in which each document is tagged with terms got from a hierarchy of concepts. Then, a system analyzes content distribution of a term’s descendants relative to other descendants in according to its joint distributions and other comparation measures in order to discover new relationship among them [4], [5]. This kind of relationship can be also used for filtering and summarizing news articles, utilizing concept distributions marked as interesting by some user.

Full Text Mining: unlike the regular associations which exclusively operate on the document indexes, this technique takes advantage of the textual content of the documents. This new TM task which consider full TM for information extraction is called Prototypical Document Extraction, where “prototypical” is defined as information that occurs in a repetitive fashion in the document collection. The working hypothesis is that repetitive document structures provide significant information about the textual base is processed. Basically, the method relies on the identification of frequent sequences of terms in the documents, and uses NLP techniques such as part-of-speech tagging and term extraction to preprocess the textual data. These techniques can be considered as an automated generalized indexing procedure that extracts from the full textual content of the documents linguistically significant structures that will constitute a new basis for frequent set extraction [12].

The system begins with collections of raw texts. Documents are first labeled with terms extracted directly from the documents. Next, the terms and additional higher-level entities are used to support a range of KDD operations on the documents. The overall system starts with a term extraction module, then a TM tool tries to perform the KDD operations and finally a visualization module can be integrated in order to navigate for the extracted knowledge and to perform the exploratory data analysis in a easier way.

A TYPICAL TM SYSTEM

The architecture of each TM system depends on the type of information they are extracting and the tasks they are going to perform. However, a general framework as shown at Fig. 2 can be described in order to understand how they are organized.

Fig. 2.- A Typical TM System Architecture

A typical Term Extraction module (Fig. 3) consists of three main steps: Linguistic Preprocessing, Term Generation and Term Filtering [4]. The Linguistic Preprocessing typically includes tasks such as Tokenization, Part-of-Speech Tagging and Lemmatizations. The goal of the tagging is to automatically associate morphosyntactic categories to the words in the document after being trained with manually-tagged corpus. In the Term Generation stage, sequences of tagged lemmas are selected as potential term candidates on the basis of relevant morphosyntactic patterns (ie. Noun-Noun, Adjective-Noun, etc). The possible combinations of terms are extracted by comparing their association coefficient (ie. “strength” of the sequence over other competing possibilities). The complete iterative procedure is finished when no new terms are generated.

Fig. 3.- The Term Extraction Task

The goal of the Term Filtering step is to reduce the number of term candidates produced by the Term
Generation stage on the basis of some statistical relevance-scoring scheme. After scoring all the terms generated in the previous stage, they are sorted based on their scores and then the top $N$ terms are selected.

The Mining Tool can be grouped into two classes: Filters and Analyzers. Filters are modules which output a subset of the document set. They are usually coupled with a visualization tool for explore/filter the input set. Analyzers apply a standard mining algorithm on their input and extract information regarding that set. The format of this information differs for the different mining tools. Whereas filters are typically used to help in further mining, analyzers are often results within themselves containing important analytical information.

APPLICATIONS AND USES

While TM may work with almost any kind of information [14], it delivers the best results when used with information that meets the following criteria:

**Knowledge worker value:** By providing new insights and a strong foundation of understanding, TM lets you add value to the knowledge base through innovation and decision making.

**Text-based content:** For TM, the information must be textual. Numerical data residing within a database structure are best served by existing DM technologies.

**Valuable content:** The value of TM is directly proportional to the value of the data you are mining. The more important the knowledge contained in the text collection, the more value you will derive by mining the data.

**Unstructured:** Highly structured information already resides within a navigable organization; TM is not as valuable in those cases, provided the structure of the information makes some sense. TM is most useful for unorganized bodies of information, particularly those that have an ongoing accumulation and change.

According to these guides, automatic analysis of text information can be used for several different general purposes such as:

- To provide an overview of the contents of a large document collection, i.e., huge clusters in a customer feedback collection could indicate where your products or services need improvement.
- To identify hidden structures between groups of objects, i.e., when your network has grown large you may want to use clustering to make sure that related documents are all connected by hyperlinks.
- To ease the process of browsing to find similar or related information.
- To gather statistics on the co-occurrence of certain words, phrases, concepts or themes on the WWW (Market Research). This information can be useful in estimating market demographics and demand curves.

SOME PRACTICAL TM SYSTEMS

**TextAnalyst:** allows to analyze large volumes of textual information, summarize, efficiently navigate and cluster documents in a textbase. It can provide the ability to perform semantic information retrieval or focus the text exploration around a certain subject. It is based on an integration of a unique linguistic and a neural network which ensures high speed and accuracy in the analysis of unstructured texts [14].

**FACT:** is a system for knowledge discovery from text. It discovers associations—patterns of co-occurrence amongst keywords labeling the items in a collection of textual documents. FACT takes a query-centered view of knowledge discovery, in which a discovery request is viewed as a query over the implicit set of possible results supported by a collection of documents, and where the background knowledge is used to specify constraints on the desired results of this query process. FACT presents the user with a simple-to-use graphical interface to the query language [4], [5].

**Document Explorer:** is a DM system for document collections. Such a collection represents an application domain, and the primary goal of the system is to derive patterns that provide knowledge about this domain. This tool searches for patterns that capture relations between concepts of the domain. The patterns which have been verified as interesting are structured and presented in a visual user interface allowing the user to operate on the results to refine and redirect mining queries or to access the associated documents. The main pattern types the system can search for, are frequent sets, associations, concept distributions and keyword graphs. In addition, there is an enhanced...
version of the system able to perform visual data mining with similar functions which is known as TextVis [4], [5].

**NeuroDoc**: is a Neural Network based platform which is used on bibliographical and textual data for clustering and mapping. The platform includes modules for linguistic preprocessing, term acquisition, hierarchical structuring and finally the analysis itself (the ILC platform - Infometrics, Language and Knowledge). Neurodoc uses the axial k-means methods, so an unsupervised winner-takes-all algorithm producing overlapping clusters, and a principal component analysis for mapping. The system has been used in experiments carried out since 1997 by INIST for CNRS and another institutions in order to analyze hundreds of pieces of research published in several journals indexed on the Science Citation Index. This tool's cluster analysis has been able to generate some important complex concept cluster and their cartography representation by using visualization tools [11].

**The Text Understander**: is a system based on a knowledge-intensive model of concept learning from few, positive-only examples that is integrated with the non-learning mode of text understanding. Its learning model is centered around the “quality” of different forms of linguistic and conceptual evidence which underlies an incremental generation and refinement of alternative concept hypotheses each capturing a different conceptual reading for an unknown lexical item [6].

**CONCLUSIONS**

This paper has attempted to suggest a new emphasis in computational linguistic work: the use of large online text collections to discover new facts and trends about the world itself. It is suggested that to make progress it is not necessary to have fully intelligent text analysis; rather, a mixture of computational and user analysis may open the door to exciting new results [4], [7], [12].

Text Mining is analogous to DM in that it uncovers relationships in information. Unlike DM, Text Mining works with information stored in an unstructured collection of text documents. Instead of just applying statistical models, TM uncovers relationships and leverages the creativity of the knowledge worker to explore these relations and discover new knowledge. It has been shown that both TM and IE tasks are different in many ways [2], [9]. In general, an IE system only relies in either extracting some information according to ends goals specified by users or finding/filtering some grammar patterns in the texts. Then, some template filling task is carried out and different facts are generated to meet users' requirements. Even so, it is a very interesting task which brings a lot of hard problem in computational corpus linguistics, however no data analysis or new knowledge discovering is done and so, that work is left to the end user by using the results of a tabular database generated by the IE system.

Finally, since the TM field is rapidly evolving, some important elements/questions should be considered when selecting among TM applications and solutions:

- Requires large up front, manual categorization, tagging or building of thesauri.
- Delivers automatic identification and indexing of concepts within the text.
- Visually presents a high level view of the entire scope of the text, with the ability to quickly drill down to relevant details.
- Enables users to make new association and relationships, presenting paths for innovation and exploration.
- Scales to process any size data set quickly.
- Handles all types of unstructured data formats and runs on multiple formats.

**REFERENCES**


