International Journal of Advanced Computer and Mathematical Sciences
ISSN 2230-9624. Vol 3, Issue 3, 2012, pp 348-354
http://bipublication.com

TEXT PATTERN ANALAYSIS

[1]A. Aarthi, [2]P.Rajaprakash Rao, [3]Vijay Dahiya

[1,2]CSE, TRR Engineering College, Hyderabad, AP, India
[3]CSE, Deenbandhu Chhotu Ram University Of Science And Technology, India

[Received-09/09/2012, Accepted-23/09/2012]

ABSTRACT

Unstructured data refers to information that either does not have a pre-defined data-model or does not fit well into data bases. Unstructured information is typically text-heavy, but may contain data such as dates, numbers, and facts [2] resulting in irregularities and ambiguities that make it difficult to understand using traditional computer programs as compared to data stored in fielded form in databases or annotated in documents.
Unstructured information can then be enriched and combined to address ambiguities and relevancy-based techniques that used to facilitate search and discovery. Examples of "unstructured data" may include books, journals, documents, meta-data, health records, audio, video, files, and unstructured text [4] such as the body of an e-mail message, Web page, or word processor document.
In 1998, Merrill Lynch cited estimates that as much as 80% of all potentially usable business information originates in unstructured form. Such estimates may not be based on primary research, but they are nonetheless widely accepted. Recently, multiple analysts have estimated that data will grow 800% over the next five years. Unstructured information [5] accounts for more than 70%–80% of all data in organizations and is growing 10–50x more than structured data.

INTRUDUCTION

Difficulties with Unstructured Data

The challenge of unstructured data is a top priority for organizations that are looking for ways to search, sort, analyze and extract knowledge from masses of documents they store and create daily. Most organizations dream of paperless office, but still generate and receive 1 millions of print documents. Digitizing these documents and intelligently using them is a universal enterprise challenge [6]. Major scanning providers offer solutions that analyze scanned documents and then store detected information in document management systems [1]. This works well with pre-defined forms, but human interaction is required when scanning unstructured text.
In the area of forensics, intelligence and security, manual monitoring of masses of unstructured data is not feasible. The ability of automatically identify people's names, addresses, credit card

and bank account numbers and other entities is the key.

In health care, although Electronic Health Records (EHRs) [5] have been increasingly becoming available over the past two decades, patient confidentiality and privacy concerns have been acting as obstacles from utilizing the incredibly valuable information they contain to further medical research. Several approaches have been reported in assigning unique encrypted identifiers to patients' ID but each comes with drawbacks.

Structured Data and Need for it

The term structured data refers to data that is identifiable because it is organized in a structure. Data that resides in fixed fields within a record or file. Relational databases and spreadsheets are examples of structured data [3].

Since, unstructured data can't be understood by computers, doesn't have any identifiable structure and also not efficiently organized for human reading too. So, in order to make the computer understand the data properly the data should be organized properly.

Dealing with Unstructured Data

Data mining and text analytics and noisy text analytics techniques are different methods used to find patterns in, or otherwise interpret, this information. Common techniques for structuring text usually involve manual tagging with meta-data for further text mining-based structuring. Several commercial solutions are available for analyzing and understanding unstructured data for business applications. This includes products from companies like SAS, IxReveal, Inxight and SPSS [6], as well as more specialized offerings such as Attensity360 and Sysomos, which focuses on analyzing unstructured social media data.

LIONSHER and problem with that version

Lionsher, the SaaS and cloud-based Learning management system with a unique online examination system which helps different organization's to choose the best possible employees for them.

As we have seen above that, all of the systems work with only structured data which is given as a input to the system in the format like xml, csv, xls, which later on goes and sits in the database.

Similarly, it was a similar process for Lionsher [3] too in its version. The inputs of the question papers for an exam were given in the document format by the client. But since the doc format given by the client was never in the structured format, as the client just simply creates the paper on the intensity of mental model.

Advantages of using Documents as the Input

Doc formats, if prepared in right way, can bring possible benefits:

· Less confusion to prepare the input file.

· Time saved, which was wasted during the rework of conversion of doc to xls or csv.

· Human power reduced, as additional person was required to do the rework.

· Decreased error rate of the process.

· Overall, Increased efficiency of the system.

Proposed system

What to do with Unstructured Information. The answer is making the unstructured data into structured data.

So, believing in that, the proposal is why not to have such a system which reads doc file format along with the rules of structuring data, which will be close to human mental model. Due to which it will not be a difficult task for a user to prepare the doc.

CLASSIFIER

The present work is an attempt to develop a parser for extracting the useful information out of the doc files and use that information in an efficient way for the on-line examination system of Lionsher [3].

The CLASSIFIER provides us with an efficient way of differentiating the useful data from the useless data. It has advantage of reducing the

human power and make the whole process to be done by the system.

The present work is targeted in this direction keeping in mind the ease of use of the system which can help the organization for working in an effective and optimistic way. The task is attempted using Ruby on Rails on GNU/Linux platform as it facilitates tremendous features to build the proposed system.

Understanding some Important Keywords

In order to understand the above mentioned project, it is necessary that we should be aware of some of the important keywords on whose bases and understanding this project has been built down. They are:

- Human Mental Model
- Text Mining
- Pattern analysis or Recognition
- Text Categorization and Text Classification

Mental Model

A **mental model** is an explanation of someone's thought process about how something works in the real world. It is a representation of the surrounding world, the relationships between its various parts and a person's intuitive perception about his or her own acts and their consequences. Mental models can help shape behaviour and set an approach to solving problems and doing tasks.

Text Mining

Text mining, sometimes alternately referred to as *text data mining*, roughly equivalent to text analytics, refers to the process of deriving high-quality information from text. High quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining [4] usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database) [3], deriving patterns within the structured data, and finally evaluation and interpretation of the output.

'High quality' in text mining usually refers to some combination of relevance, novelty, and interestingness. Typical text mining tasks include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modeling (i.e., learning relations between named entities) [6].

Text analysis involves information retrieval, lexical analysis to study word frequency distributions, pattern recognition, tagging/annotation, information extraction, data mining techniques including link and association analysis, visualization, and predictive analytics. The overarching goal is, essentially, to turn text into data for analysis via application of natural language processing (NLP) [3] and analytical methods.

The term text analytics describes a set of linguistic, statistical, and machine learning techniques that model and structure the information content of textual sources for business intelligence, exploratory data analysis, research, or investigation. The term is roughly synonymous with text mining; indeed, Ronen Feldman modified a 2000 description of "text mining" in 2004 to describe "text analytics." The latter term is now used more frequently in business settings while "text mining" is used in some of the earliest application areas, dating to the 1980s, notably life-sciences research and government intelligence.

The term text analytics also describes that application of text analytics to respond to business problems, whether independently or in conjunction with query and analysis of fielded, numerical data. It is a truism that 80 percent of business-relevant information originates in unstructured form, primarily text. These techniques and processes discover and present knowledge – facts, business rules, and relationships – that is otherwise locked in textual form, impenetrable to automated processing.

Pattern Analysis or Recognition

In machine learning, pattern recognition is the assignment of a label to a given input value. An example of pattern recognition [3] is classification, which attempts to assign each input value to one of a given set of *classes* (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence) [2]; and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to do "fuzzy" matching of inputs. This is opposed to pattern-matching algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is generally not considered a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing similar quality output to the sort provided by pattern-recognition algorithms.

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. Supervised learning assumes that a set of training data (the training set) [1] has been provided, consisting of a set of instances that have been properly labeled by hand with the correct output. A learning procedure then generates a model that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data. Unsupervised learning, on the other hand, assumes training data that has not been hand-labeled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances.

A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labeled and unlabeled data (typically a small set of labeled data combined with a large amount of unlabeled data). Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, the data to be labeled is the training data.

Note that sometimes different terms are used to describe the corresponding supervised and unsupervised learning procedures for the same type of output. For example, the unsupervised equivalent of classification is normally known as clustering, based on the common perception of the task as involving no training data to speak of, and of grouping the input data into clusters based on some inherent similarity measure (e.g. the distance between instances, considered as vectors in a multi-dimensional vector space), rather than assigning each input instance into one of a set of pre-defined classes. Note also that in some fields, the terminology is different: For example, in community ecology, the term "classification" is used to refer to what is commonly known as "clustering".

The piece of input data for which an output value is generated is formally termed an instance. The instance is formally described by a vector of features, which together constitute a description of all known characteristics of the instance. (These feature vectors can be seen as defining points in an appropriate multidimensional space, and methods for manipulating vectors in vector spaces can be correspondingly applied to them, such as computing the dot product or the angle between two vectors.) Typically, features are

either categorical (also known as nominal, i.e. consisting of one of a set of unordered items, such as a gender of "male" or "female", or a blood type of "A", "B", "AB" or "O"), ordinal (consisting of one of a set of ordered items, e.g. "large", "medium" or "small") [2], integer-valued (e.g. a count of the number of occurrences of a particular word in an email) [3] or real-valued (e.g. a measurement of blood pressure) [4]. Often, categorical and ordinal data are grouped together; likewise for integer-valued and real-valued data. Furthermore, many algorithms work only in terms of categorical data and require that real-valued or integer-valued data be discretized into groups (e.g. less than 5, between 5 and 10, or greater than 10).

Many common pattern recognition algorithms are probabilistic in nature, in that they use statistical inference to find the best label for a given instance. Unlike other algorithms, which simply output a "best" label, oftentimes probabilistic algorithms also output a probability of the instance being described by the given label. In addition, many probabilistic algorithms output a list of the N-best labels with associated probabilities, for some value of N, instead of simply a single best label. When the number of possible labels is fairly small (e.g. in the case of classification), N may be set so that the probability of all possible labels is output. Probabilistic algorithms have many advantages over non-probabilistic algorithms.

They output a confidence value associated with their choice. (Note that some other algorithms may also output confidence values, but in general, only for probabilistic algorithms is this value mathematically grounded in probability theory. Non-probabilistic confidence values can in general not be given any specific meaning, and only used to compare against other confidence values output by the same algorithm.)

Correspondingly, they can abstain when the confidence of choosing any particular output is too low. Because of the probabilities output, probabilistic pattern-recognition algorithms can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of error propagation.

Techniques to transform the raw feature vectors are sometimes used prior to application of the pattern-matching algorithm. For example, feature extraction algorithms attempt to reduce a large-dimensionality feature vector into a smaller-dimensionality vector that is easier to work with and encodes less redundancy, using mathematical techniques such as principal components analysis (PCA). Feature selection algorithms, attempt to directly prune out redundant or irrelevant features. The distinction between the two is that the resulting features after feature extraction has taken place are of a different sort than the original features and may not easily be interpretable, while the features left after feature selection are simply a subset of the original features.

Text Categorization and Text Classification

Data classification is the categorization of data for its most effective and efficient use. In a basic approach to storing computer data, data can be classified according to its critical value or how often it needs to be accessed, with the most critical or often-used data stored on the fastest media while other data can be stored on slower (and less expensive) media. This kind of classification tends to optimize the use of data storage for multiple purposes - technical, administrative, legal, and economic.

Data can be classified according to any criteria, not only relative importance or frequency of use. For example, data can be broken down according to its topical content, file type, operating platform, average file size in megabytes or gigabytes, when it was created, when it was last accessed or modified, which person or department last accessed or modified it, and which personnel or departments use it the most. A well-planned data classification system makes essential data easy to find. This can be of

particular importance in risk management, legal discovery, and compliance with government regulations.

Document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories. This may be done "manually" (or "intellectually") or algorithmically. The intellectual classification of documents has mostly been the province of library science, while the algorithmic classification of documents is used mainly in information science and computer science. The problems are overlapping, however, and there is therefore also interdisciplinary research on document classification.

The documents to be classified may be texts, images, music, etc. Each kind of document possesses its special classification problems. When not otherwise specified, text classification is implied.

Documents may be classified according to their subjects or according to other attributes (such as document type, author, printing year etc.). In the rest of this article only subject classification is considered. There are two main philosophies of subject classification of documents: The content based approach and the request based approach.

Architecture

The system design of project is relatively simple. MySql database is the main storage repository for all data. These data are managed by the business logic layer of the application. This layer gives the underlying raw data more structure and meaning, preparing it for human consumption. The now human readable data are then presented to the support technician by another software application or web page. The end-user of the system can give the document for parsing.

Once the parser module is over, the user sees the preview of the parsed document.

CONCLUSION

CLASSIFIER is an open source document parsing system that allows us to provide the unstructured data to the system in the form of structured data. User can provide an input in the form of a question paper, where he/she can add various question formats like MCQ, MAQ, PTQ, FIB etc under various topics and sub topics. CLASSIFIER provides us with fast and efficient system, where a user can upload various questions using various question papers. The coding of classifier is done in such a way that if there is any error during the uploading or parsing process, then he/she can view the error in log file or in the errors page and he/she can easily solve that error. The output of the upload questions can be seen on the preview page and user can have a cross check whether the questions being uploaded are same or not.

CLASSIFIER is installed in GNU/Linux environment, though it can work on Windows with a bit of extra work. It runs on Ruby on Rails with Mongrel web server, and uses MySQL as a back-end database. It has been thoroughly tested for usability.

Adding images without any tag, adding match the column type questions, adding multiple threads to the system for increased performance remain the future tasks associated with CLASSIFIER that are to be carried out.

REFERENCES

[1] Braid Ediger, “Advanced Rails”,O'Reilly, First Edition, December 2007.

[2] Kevin Marshall,Chad Pytel,John Yurek, “Pro Active Record”, Apress, First Edition, January 2007.

[3] Chad Flower, “Rails Recipies”, Beta Book, Third Edition, October 2008.

[4] Obie Fernandez, “The Rails Way”, Addision Wesley, First Edition, December 2008.

[5] Dave Thomas,Andy Hunt “Programming Ruby**”,** The Pragmatic Bookshelf.

[6] David A. Black, “Ruby for Rails”, Manning Publications, Third Edition, January 2006.
[7] UNSTRUCTURED Data Motivation

Web Resources

[1] http://ryanbigg.com/2010/12/ubuntu-ruby-rvm-rails-and-you/
[2] http://ruby.railstutorial.org/ruby-on-rails-tutorial-book
[3] http://net.tutsplus.com/tutorials/ruby/getting-started-with-restful-authentication-in-rails/
[4] http://www.webupd8.org/2010/02/how-to-extract-all-text-from-pdfs.html
[5] http://net.tutsplus.com/tutorials/ruby/ruby-for-newbies-working-with-directories-andfiles/
[6] http://guides.rubyonrails.org/association_basics.html
[7] http://rubular.com/
[8] http://twitter.github.com/bootstrap/