Asian Journal of Applied Research (AJAR) 2015

E-ISSN: 2395-1702 P-ISSN: 2395-0382
Volume 01- Issue 01-, pp-23-28

Research Paper Open Access

TEXT MINING IN-R UTILIZING TM-PACKAGE FOR TAMIL DOCUMENT USING THIRUKKURAL

BALAJI K [1] B.MAGESH [2]
M E Department of Computer Science & Engineering
Arunai Engineering College, Tiruvannamalai
Tamilnadu, India.
balajjee.mecse@gmail.com, magesheera@gmail.com

***Abstract*—**Tamil contains a huge amount of online text documents, it is nearly impossible to manually organize such vast data. The necessity to extract useful and relevant information such large data sets has managed to an important need to develop computationally efficient text mining techniques. The competence of text analytics applies analytic tools to acquire from collection of text documents using automated process can learn from massive amounts of texts, much more than human can. The tm package provides a structure for text mining applications facilities within R to be carried out a typical applications.

Keywords—Data import, count-based analysis, text analysis, text classification, corpus handling, term-document matrices,

1. Introduction

Text mining is characterized by two or more field of study between data mining, linguistics, computational statistics, and computer science. The standard techniques are text classification, text clustering, document summarization and corpus analysis. The **tm** package offers a structure for text mining applications within R in order to use text mining facilities. The objective is similar to humans learning by reading books. This paper deals with Thirukkural, which received world class honors, is the most famous literature among the many moral literatures that were identified and were still in practice till date in Tamil language. The unique meaning and the simplest form described attention of the worldly scholars and philosophers across continents and nations. This is called as Kural because of its shortest meter which is not more than 1 ¾ in size. The aim is Tamil document particularly Thirukkural to feasibly summaries the main themes and to identifying most interest contents of Thirukkural.

2. ‘R’ frame work of tm- package

Traditional applications in text mining from the data mining communal, like document clustering and document classification. The beneficial is to convert the text into a structured format based on term frequencies and sequentially applies standard data mining techniques. Exhibiting the qualities of applications in document clustering, specific distance measures, like the Cosine, play an important role. The other advanced text mining methods have been used in several fields, e.g., in linguistic. The standard text mining structure with the large amount of valuable information in texts which is not available for structured data formats. Statistical contexts for text mining applications include latent semantic analysis techniques, used for statistical methods automatically investigating grading such as ideas for new products' provides a major statistical computing product deals text mining capabilities, and many recognized data mining products provides solutions for text mining tasks. R provides the competencies and features includes the tm package.

a. *Pre-process*: data preparation, importing, cleaning and general pre-processing.

b. *Associate:* association analysis that, is finding associations for a given term based on counting co-occurrence frequencies.

c. *Cluster:* clustering the text document of similar documents into the same groups.

d. *Summarize:* summarization of important terms and highlighting high frequency- terms.

e. *Categorize:* classification of texts into predefined categories.

In **R** the extension package **ttda** provides some methods for textual data analysis. A text mining framework provides the open source statistical computing environment in R centered on the new extension package **tm**. This open source package, extended based on common functions and object-oriented concepts, provides the basic infrastructure needed to organize, transform, and analyze textual data. R provides one of the most versatile statistical computing environments available for standard text mining methodology. These methods was limited to "classical" structured input data formats in R. The tm package facilitate a framework that allows to existing methods for text data structures as well as advanced text mining methods used beyond the scope of most commercial products, like string kernels or latent semantic analysis, to be facilitated via the tm package, Such as kernlab or lsa, from the natural language processing community. Hence the tm package provides a framework for flexible integration of statistical methods from R, in order to interfaces for well-known open source text mining infrastructure methods, and a modularized extension mechanism for text mining purposes.

3. Theoretical procedure and background

A text mining analysis involves several process steps in texts documents from a computer perspective. The text data's are unstructured collections of words with a set of highly unrelated input texts. The first step is to import these texts into computing environment in R directly. It is important to organize and structure the texts to be able to access them in a distinctive manner. Once the texts are ordered in a source file, the second step is cleanup up the texts, as well as preprocessing the texts in to a convenient representation for further analysis. This step involves text transformation of eliminating whitespace removal, punctuation removal, or stemming processes. The third step involves the analyzed text documents should be able to transform the preprocessed texts into an organized manner. The "classical" text mining tasks, typically implies the creation of a term-document matrix, perhaps the most common format for text computation process. Finally the analyzed text document should be carried out for standard computation techniques. The computation on texts with standard techniques from statistics and text mining within the document clustering or document classification methods. A text mining framework managing text documents, and conceptual process of document manipulation and makes easier to use of various text formats to call a text document collection or corpus.

The tm package provides text document classification tools and algorithms for efficiently work with the documents. The framework helps to perform common tasks, like whitespace removal, stemming or stopword deletion. This function effectively work on text document collections as well as transformations process. The important concept is filtering involves predicate functions on collections to extract patterns and compute the operation of joining text document collections. Text mining most common approach is to create a term-document matrix holding frequencies of distinct terms for each document.

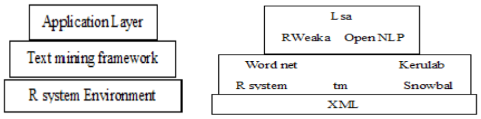


Figure 1: Conceptual layers and packages

4. Data structures and algorithms

The implementation data framework for computing a text document in R on a middle ware of text mining classes provide access to various text mining methods, where the methods operates without explicitly knowing the inside information details of internal text data structures. The text mining modules are written and abstracted functions are possible. So it is easy to add new methods for formalising on the application layer. The R system environment is made up of the R core functions and the XML package for handling XML documents internally. The text mining framework facilitate the new tm package with help of Rstem or Snowball for stemming the document ordering purpose. The framework is provides a key structure for latent semantic analysis (LSA) for finding a term-document matrix easily exported from tm package. The lsa provides its own routines for generating term-document matrices, to enhance with tm for handling complex input formats, pre-processing, and text manipulations.

A. Data structures

Text Document collection

The structure of text document collection for managing in text mining is called text document collection, also called as corpus in linguistics. The corpus represents a collection of text documents and can be manipulated as a database for texts. The collected corpus are holding Text Documents with actual text corpora and local dataset. The text document collection stored locally and globally for database support. The metadata contains two types specifically Document Metadata and Collection Metadata. The Document metadata (DMetaData) is used for specific information to text documents and holds with an own entity. The Collection metadata (CMetaData) is used for global information metadata on the collection level but not essentially related to single text documents. The database (DBControl) controls is used to store the backend information of metadata. The tm package holds few bits of memory is able to work with very large text collections, and support the objects are stored in to the disk which other objects influenced for another use.

ReaderControl

Loading a corpus directly into memory or when required to load on demand in to the tm package the method get Reader ( ) used for several readers can load the corpus or text document collection.

DbControl

A method used to support the list of three components for controlling useDb, dbName and dbType setting the particular DBControl values must be activated, the file name to the database, and the database type.

Text documents

The basic required information are managed by a text document collections like topics are separated by unique ID for identification that helps further analysis for common file formats.

Text repositories

The TextRepository is used to keep tracked stored information and backtracked with original input data. The dynamic RepoMetaData to save the history of a text document collection.

B. Term-document matrices

The term-document matrices is representing the texts for computations finding the bag-of-words which is ordering the terms with matrix formation. The document IDs as rows, terms as columns and the elements are term frequencies. The tm package provide TermDocMatrix a term-document matrix for a given Corpus elements and Data of the formal class matrix from package Matrix to hold the frequencies in compressed sparse matrix format.

C. Term Frequency Matrices

The term frequency (weightTf) used to different weightings of the term elements and makes easier by calling a weighting function on the matrix elements. The tm facilitating weighting schemes contain the binary frequency (weightBin) method which is used to eliminates multiple entries, or the inverse document frequency (weightTfIdf) weighting method facilitating more importance to perceive the difference to irrelevant terms.

D. Term Frequency and Inverse Document Frequency

The information retrieval system is using the TF-IDF principle to annotate the document collection with the two composite components (i) Term Frequency and (ii) Inverse Document Frequency. The term frequency is the frequency count of terms occurs in a document and IDF of a term measures how important collections term in the whole corpus.

TF (term, doc) = COUNT (term, doc)

IDF (term, Collection) = $¿ d € C , term € d \vee ¿$

$$\frac{\log(|C|)}{¿}$$

IF-IDF (term, doc, col) =TF (term, d).IDF (term, Col)

The similarity among the documents have been estimated based on this TF-IDF and or cosine similarity as given below.

SIM_TF-IDF (d1, d2, C) =

CosineSim (TFIDF (d1, C), TFIDF (d2, C))

CosineSim(X, Y) = $\frac{X . Y}{\|X\|\|Y\|}$

$$\frac{X . Y}{\|X\|\|Y\|}$$

The TermDocMatrix method which used to creating a term document matrix form a text document collection. The method provides a modular structure for creating a matrix from documents calling by control argument. E.g., tokenization of single words calling a method by (NGramTokenizer).

E. Sources

The tm package interfaces with source file document or corpus for input process and allows to facilitate structured document format. A source document virtually created and translated machine understandable file format and stored in virtual abstract class. The method LoDSupport indicating load on demand, and Position holds internal guidance for DefaultReader function and encoding the collected corpus used by internal R routines for accessing texts through the source (defaults to UTF-8 for all sources) file format.

(B). Algorithm

The collection of text documents separated by directory and load the method DirSource into memory. The class PlainTextDocument used for default reader function for reads the elements. The method and class functions automatically valid the text collection. The length () method used to returns the number of text documents collection. The show () method is used to summarizing the text collection. The method summary () used detailed message collection and summarizing the text document collection. The method inspect () function allows to show the structure which is hidden by show () and summary () methods.

F. Transformations

The process of transformation operate on each text document collection applying a function to representation of the whole text document collection. Filter operations used to find subsets of the text document collection. A subset is defined by a function applied to each text document resulting in a Boolean answer. The filter function is just a predicate function easily identify documents with common characteristics. The process of transformations and filters of a text document collection with d1.d2, d3….dn consisting of corpus data (Data). Transformations are perform the tmMap () function which applies a function FUN to all elements of the collection on single text documents.

5. Preprocessing

The preprocessing methods for cleaning up and structuring the input text for further analysis, to perform typical preprocessing steps in the tm package.

A. Data import

The default encoding used by sources is always assumed to be UTF-8 manually set the encoding via the encoding parameter creating a connection with encoding which is passed to the source (corpus).

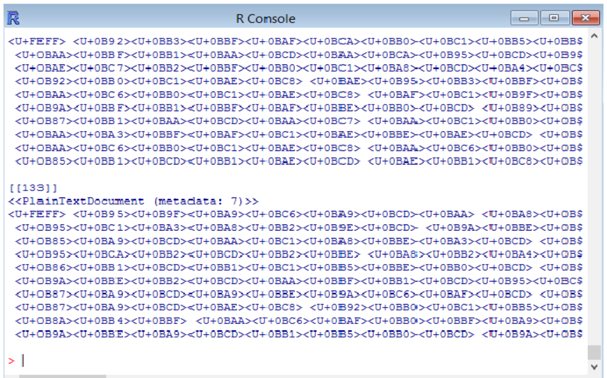
B. Whitespace elimination

The preprocessing steps removal of white space tasks tm provides transformations can be used with tmMap ().

6. Result

The experimental results taken for Tamil document classification using Thirukkural and finding the most frequent terms, document term matrix and set the frequencies of each terms elements are found throw tm package from the R environment.

INSPECTING DOCUMENTS WITH MACHINE TRANSLATION

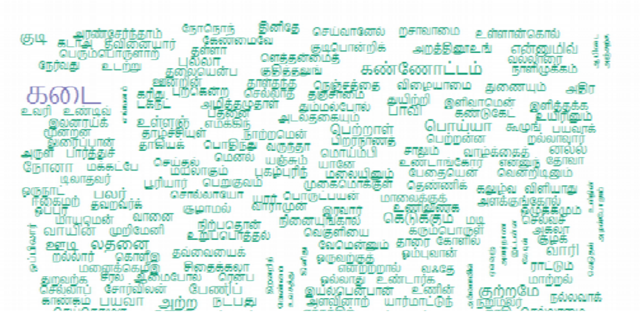


```
<U+FEFF> <U+0B92><U+0BB3><U+0BBF><U+0BAF><U+0BCA><U+0BB0><U+0BC1><U+0BB5><U+0BB$
 <U+OBAA><U+0BBF><U+0BB1><U+0BAA><U+0BCD><U+0BAA><U+0BCA><U+0B95><U+0BCD><U+0B9$
 <U+OBAE><U+0BC7><U+0BB2><U+0BBF><U+0BB0><U+0BC1><U+0BA8><U+0BCD><U+0BA4><U+0BC$
 <U+OB92><U+0BB0><U+0BC1><U+0BAE><U+0BC8> <U+0BAE><U+0B95><U+0BB3><U+0BBF><U+OB$
 <U+OBAA><U+0BC6><U+0BB0><U+0BC1><U+0BAE><U+0BC8> <U+0BAF><U+0BC1><U+0B9F><U+OB$
 <U+OB9A><U+0BBF><U+0BB1><U+0BBF><U+0BAF><U+0BBE><U+0BB0><U+0BCD> <U+0B89><U+OB$
 <U+OB87><U+0BB1><U+0BAA><U+0BCD><U+0BAA><U+0BC7> <U+0BAA><U+0BC1><U+0BB0><U+OB$
 <U+OBAA><U+0BA3><U+0BBF><U+0BAF><U+0BC1><U+0BAE><U+0BBE><U+0BAE><U+0BCD> <U+OB$
 <U+OBAA><U+0BC6><U+0BB0><U+0BC1><U+0BAE><U+0BC8> <U+0BAA><U+0BC6><U+0BB0><U+OB$
 <U+OB85><U+0BB1><U+0BCD><U+0BB1><U+0BAE><U+0BCD> <U+0BAE><U+0BB1><U+0BC8><U+OB$

[[133]]
<<PlainTextDocument (metadata: 7)>>
<U+FEFF> <U+0B95><U+0B9F><U+0BA9><U+0BC6><U+0BA9><U+0BCD><U+0BAA> <U+0BA8><U+OB$
 <U+OB95><U+0BC1><U+0BA3><U+0BA8><U+0BB2><U+0B9E><U+0BCD> <U+0B9A><U+0BBE><U+OB$
 <U+OB85><U+0BA9><U+0BCD><U+0BAA><U+0BC1><U+0BA8><U+0BBE><U+0BA3><U+0BCD> <U+OB$
 <U+OB95><U+0BCA><U+0BB2><U+0BCD><U+0BB2><U+0BBE> <U+0BA8><U+0BB2><U+0BA4><U+OB$
 <U+OB86><U+0BB1><U+0BCD><U+0BB1><U+0BC1><U+0BB5><U+0BBE><U+0BB0><U+0BCD> <U+OB$
 <U+OB9A><U+0BBE><U+0BB2><U+0BCD><U+0BAA><U+0BBF><U+0BB1><U+0BCD><U+0B95><U+0BC$
 <U+OB87><U+0BA9><U+0BCD><U+0BA9><U+0BBE><U+0B9A><U+0BC6><U+0BAF><U+0BCD> <U+OB$
 <U+OB87><U+0BA9><U+0BCD><U+0BAE><U+0BC8> <U+0B92><U+0BB0><U+0BC1><U+0BB5><U+OB$
 <U+OB8A><U+0BB4><U+0BBF> <U+0BAA><U+0BC6><U+0BAF><U+0BB0><U+0BBF><U+0BA9><U+OB$
 <U+OB9A><U+0BBE><U+0BA9><U+0BCD><U+0BB1><U+0BB5><U+0BB0><U+0BCD> <U+0B9A><U+OB$

> |
```

FINDING MOST FREQUENT WORDS IN THIRUKKURAL



7. Conclusion

The Text mining applications using R-tm package is easy to use various text formats. The framework provides a various customized demands in data structures and algorithms. The package is facilitating in a modular way to enable easy integration of new file formats for structure text document functions and filter operations. The tm facilitation easy access to preprocessing and manipulation of classical text mining operations. The filter classification is offered to filtering documents and clean-up unnecessary elements from the text collection. The package enables the document-term-matrix use in text mining application for classification clustering. The **tm** package capabilities to support text mining application provide easy compare with other text mining tools. In future to develop a tool for Tamil linguistics because there is no proper classification tool available for Tamil text documents.

References

[1] Bilisoly R (2008). Practical Text Mining with Perl. Wiley Series on Methods and Applications in Data Mining. Wiley. ISBN 9780470382851. URL http://books.google.com.au/books?id=YkMFVbsrdzkC.

[2] Bouchet-Valat M (2013). SnowballC: Snowball stemmers based on the C libstemmer UTF-8library. R package version 0.5, URL http://CRAN. R project.org/package=SnowballC.

[3] Feinerer I, Hornik K (2014). Tm: Text Mining Package. R package version 0.5-10, URL http://CRAN.R project.org/package=tm.

[4] Gentry J, Long L, Gentleman R, Falcon S, Hahne F, Sarkar D, Hansen KD (2014). Rgraphviz: Provides plotting capabilities for R graph objects. R package version2.6.0.

[5] Neuwirth E (2011). RColorBrewer: ColorBrewer palettes.R package version 1.0-5, URL http: //CRAN.R project.org/package=RColorBrewer.

[6] R Core Team (2014). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[7] Williams GJ (2009). \Rattle: A Data Mining GUI for R." The R Journal, 1(2), 45{55. URLhttp://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf.

[8] Williams GJ (2011). Data Mining with Rattle and R: The art of excavating data for knowledge discovery. Use R! Springer, New York. URL http://www.amazon.com/gp/product/