Proceedings of the First International Conference on Information Technology and Knowledge Management pp. 71–75
DOI: 10.15439/2018KM46
ISSN 2300-5963 ACSIS, Vol. 14

Importance of Text Data Preprocessing & Implementation in RapidMiner

Vaishali Kalra[1], Dr. Rashmi Aggarwal[2]

Manav Rachna International Institute of Research & Studies, Faridabad
[1]arya.vaishali17@gmail.com, [2]rashmi.fca@mriu.edu.in

***Abstract*—Data preparation is an important phase before applying any machine learning algorithms. Same with the text data before applying any machine learning algorithm on text data, it requires data preparation. The data preparation is done by data preprocessing. The preprocessing of text means cleaning of noise such as: cleaning of stop words, punctuation, terms which doesn't carry much weightage in context to the text, etc. In this paper, we describe in detail how to prepare data for machine learning algorithms using RapidMiner tool. This preprocessing is followed by conversion of bag of words into term vector model and describe about the various algorithms which can be applied in RapidMiner for data analysis and predictive modeling. We also discussed about the challenges and applications of text mining in recent days**

***Index Terms*—RapidMiner, Preprocessing of text, TF-IDF, Term Vector Model.**

I. Introduction

THE TEXT data available on web and on computers is increasing rapidly, managing that text data requires intelligent algorithm to retrieve relevant information from the data repositories. The retrieval of information is called text mining. Text mining is not a new concept, it is evolved from data mining and all the data mining algorithms can be applied on the textual data. The difference between the two is data mining is applied on structured data and relational data whereas textual mining deals with all unstructured and semi-structured data well. Nowadays text mining depends upon whether we are looking for the context of the text or content of the text.

If we are looking for context of the text and want to provide the environment for the user to collect the similar patterns together then clustering, visualization and navigation can be applied and if the user want to find out the relationship between the patterns, discover the new relationships and to summarize the text then content is analyzed and techniques like classification, summarization is applied with the support of information retrieval, information extraction and natural language processing.

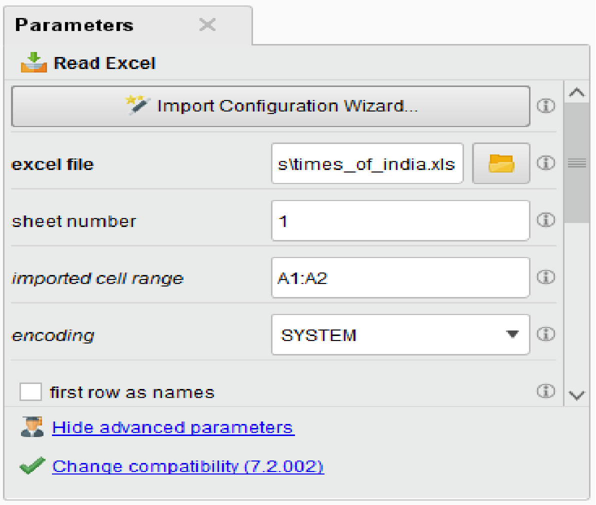
As we said the text mining works well on unstructured data. Actually to make this possible, the data is to be converted into semi structured format or in structured format so the data mining machine learning algorithms can be applied easily. This conversion of data is done by preprocessing of the data. The preprocessing of the text data is an essential step as there we prepare the text data ready for the mining. If we do not apply then data would be very inconsistent and could not generate good analytics results. So in preprocessing of the data all the punctuation, unimportant words are removed and words can be grouped into groups, words can stemmed to their roots, all missing values can be replaced with some values, case of text could be replaced into a single one, depending upon the requirement of the application we can apply different steps. After the preprocessing of the data, the data is to be converted in to vector space model and on to that vector-space model, various algorithms works.

In this paper we are going to discuss the preprocessing of the data in detail using Rapidminer tool in section (a), in section (b) we will discuss how the term vector space model is used to convert the term into vector, in next section (c) the various algorithms used for mining task will be discussed and in section (d) Applications of text mining and (e) Conclusion and Future Work.

Section (a): Preprocessing of data

Before starting the preprocessing of the data, we need to fetch the data from the source or data repositories. We have chosen a web link for retrieval of information i.e. a news site "The Times of India" a particular link which is containing the 44th president Obama's letter for the new president of U.S.A. This link does not contain only the letter content but also contains other information like hyperlinks to other webpages, some ads given by google and some pictures. This is why this data is called unstructured data. To fetch the data from the site, we have applied getPages function; this function uses a get method to send the request to server and fetches pages mentioned in the input source. Input source provided to this function can be any excel file, csv file, read from Url etc. There are various functions available in Rapidminer to read the data from various sources and providing it to getPages function. We have applied ReadExcel operator, parameters of this function are shown in figure 1. This operator take input as excel fie containing links to retrieve from the web. In the excel file you can set the attribute, like if you want to scan more than one link from the file under the attribute named "link-to-scan" , then name can be assigned in the first row. This operator considers the first row as the name of the attributes. After seeding the excel file, configuration wizard need to be set. This configuration wizard configure which particular sheet of excel file to be set to read the data and selects the number of rows to read from the excel data. When you set this configuration wizard, the getPages automatically retrieves the attributes to be scanned from excel sheet, you can select those attributes and process the getPages function. After applying this function another function getContent is required to get the textual data from html pages.

After retrieval of data, the data is ready for the preprocessing. For preprocessing of data another function we have applied is "Process Document from Data". This function converts the data into word vector which makes the text data ready for applying the various data mining algorithms. This module calculates the TF-IDF of the word set, which we will discuss later in section (b)[8].Firstly we will discuss the preprocessing steps:


Parameters
Read Excel
Import Configuration Wizard...
Hide advanced parameters
Change compatibility (7.2.002)

excel file: s\times_of_india.xls

sheet number: 1

imported cell range: A1:A2

encoding: SYSTEM

☐ first row as names

Figure 1: ReadExcel operator parameters

1) Tokenization:
This process split the sequence of strings into words. It removes all the punctuations from the text data and gives words of text which is called tokens. RapidMiner tool provides three ways of splitting; one is default which is commonly used called non letter character where it split on the basis of non-letter of character like spaces, commas, full-stops etc. The second mode is specify character, wherein you can specify characters based on which sentence is split into tokens and the third one is regular expression, wherein the regular expression is provided to split the sentence into tokens. We have chosen the default mode i.e. non letter character mode. The complete process screenshot is shown in figure 2. And in figure 3, inside view of Process Document from data module is shown.

2) Filter Stop-Words:
This process removes words from the document which does not play any important in giving intelligent pattern or information. Eg: the words like "How" "What", "are" etc. There also we can apply different types of functions which are supported by RapidMiner like Filter Stopwords based on Dictionary, English Language, German, French, Arabic etc. we have applied for English.

3) Filter Tokens by Length:

This is very interesting function, using which you can filter token of specific length, the length attribute needed to be provided in the parameters min chars and max chars; wherein min chars you can specify the minimum length of each token to be in the document to its maximum range.

4) Stemming :
In this process you can stem the words to its root, it roves all the suffixes like tens suffixes: responded to response, eaten to eat and plurals to root like women to woman, men to men and horses to horse. RapidMiner supports variety of algorithms for stemming. Porter, snowball, lovins, dictionary, Arabic to support Arabic language etc. we have used the most popular one snowball in our process. Snowball comprised of 41 rules to stem the words. Although stemming sometime loses the meaning of actual word but still works well in most of the cases.

5) Transform Cases:
This step is useful in normalizing the text. The text would get converted into a single case either to Uppercase or lowercase. This is not always necessary to apply; it depends upon the requirement of a particular problem. We have not applied this operator in our process.

After the completion of preprocessing, the document is to be converted into vector model. In rapid miner this is itself done in “Process Document from Data” by choosing the parameter vector creation to TF-IDF. But we will separately discuss how this vector model is created in our next section.

Section (b): Term Vector space model

The preprocessing of document gives us a document with bag of words only. On which we cannot apply the algorithms directly. We need to convert this bag of words into term vector. The term vector gives a numeric values corresponding to each term appearing in a document which is very helpful in feature selection.

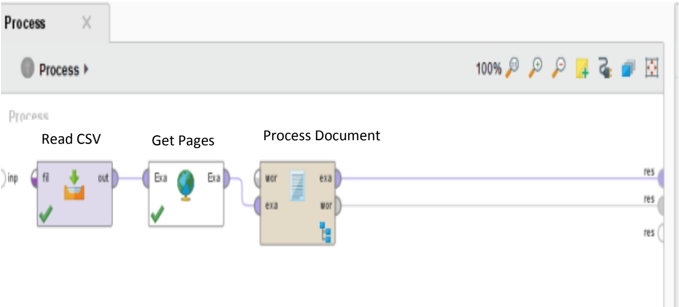


Figure 2: Process for preparing Term Vector Model

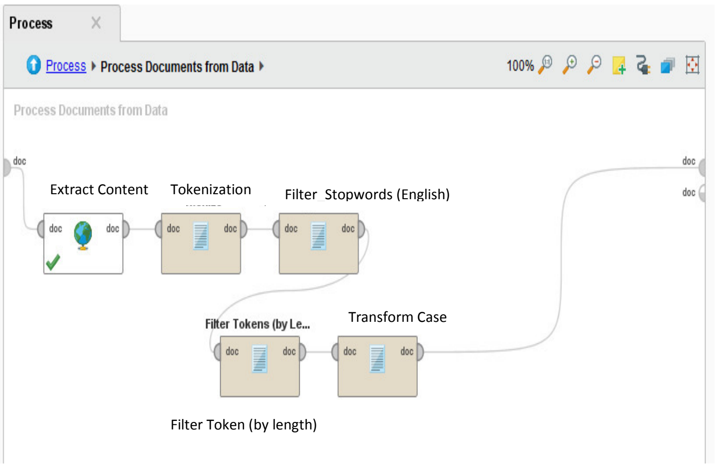


Figure 3: Preprocess

There are three ways for converting terms into vector: Term Frequency, Term Occurrences and Term Frequency-inverted document frequency (TF-IDF). The most useful and popular one is TF-IDF[9]. This gives the higher weightage to the important term and lesser weightage to the unimportant term. Its vector value lies in between of 0 to 1. 0 means the term has no importance in context to the documents in which we are looking for and 1 means the terms are relevant. For calculating the TF-IDF values the document is converted into inverted text file wherein for all documents words are extracted and corresponding to each word a weight is assigned i.e. the term occurrence value and document occurrences using TF-IDF value is calculated.

TF-IDF =

$$\frac{No.of\ times\ appears\ in\ document}{Total\ no\ terms\ in\ document} \times \log \frac{No.of\ documents\ under\ consideration}{No.of\ documents\ that\ contain\ the\ keyword}$$

After preparing the term vector model, various algorithms can be applied on the data. In the section we will discuss about the various algorithms which can be applied on textual data for mining purpose.

Section (c): Algorithms for Text Mining

There are various algorithms for text mining for different tasks, but we are giving only overview some important algorithms here.

I. **Unsupervised Algorithms**: Algorithms in which no training data is provided to train the system. The system itself evolved and generates results depending on the data provided. The most popular algorithm under this category is clustering and topic modelling [10]. Both are used for segmentation of data into group, clustering is known for hard segmentation and topic modelling is soft segmentation. RapidMiner provides support of various segmentation algorithms like K-Means, K-medoid, agglomerative clustering, support vector machine[5]. Most popular clustering algorithms are described below:

- Agglomerative Clustering: This is known as hierarchical clustering, where pairwise documents are grouped together based on some similarity criteria in bottom to up fashion. A tree like structure is formed which is very useful in searching algorithm or in searching the query from the existing databases.
- K-medoid: A center point is chosen from the original data set which is known as a medoid point. Documents are clustered around this point based on similarity measure. After forming the cluster, the medoid point is recalculated until the convergence arrived. This does not work well on textual data as textual data is sparse in nature.
- k-means clustering: In k-means clustering, the value of k is assumed and this k gives the information about number of clusters to be formed for the given dataset. This k also serve as the centroid of a particular cluster, around which documents are clustered based on similarity measure and in each iteration like medoid, k (centroid) is updated. The only disadvantage of this algorithm is the value of k which is seeded initially. For finding the perfect k, various other algorithms can be used to give a supervised k-value.

.

II. **Supervised Algorithms:** Under this category training data is needed to be provided, classification problems comes under this category. Various algorithms like decision tree based[2], rule based, bayesian, neural nets, etc are provided into RapidMiner, which are being used for predictive modelling[1]. Algorithms for classification are described below:

Decision Tree: In this algorithm for classifying the data set, hierarchical process is followed, where at each node portioning is done based on a predicate. It's a top down approach starting from the root. To partition the node various splitting measures are used like single attribute split wherein based on single value or a word phrase node is split, another one is based on document similarity, wherein based on the similarity between the two documents node is split. Third one is for multivalued attribute, where to split the node discriminant function is applied. For convergence of the tree, you can set a certain threshold like the maximum depth of the tree.

Neural Networks: Neural nets works in two phases: learning phase/training phase and testing phase. To train the network, back propagation algorithm is applied in feed forward neural structure and completion of training the network is tested to check the classification results and network is validated. In RapidMiner we can set the number of training cycles, the number of hidden layers and the minimum error on which network converged.

Rule Based: There are also two phases: growing phase and pruning phase. In growing phase, rules are constructed by considering every possible value of each attribute and in pruning phase, rules are pruned based on pruned metric.

Bayesian Algorithm: It is a probabilistic model, based on Bayes theorem which considers each attribute independently in classification of the data. Small amount of training is required as each attribute plays an independent role in classification of the dataset. In RapidMIner there are two algorithms in support of this are: Naive Bayes and Naive Bayes kernel model.

Section (d): Applications of Text Mining

Text mining applications ranges from information retrieval to link analysis, there is list of applications:

I. Tweeter analysis: analysis of social, political and academia tweets.

II. Sentiment analysis: This is also known as opinion mining, which gives positive, negative, neutral feedback of users. This is widely used nowadays in selecting a hotel, in buying any product from e-commerce site, feedback of any restaurant [4].

III. Biomedical sciences: Mining on biomedical data and bio-informatics is latest research area nowadays.

IV. Industrial applications: in industries it is widely used for data analysis tasks, by testers of the project to stock market, or for the competitive businesses it is generally used.

V. Web search Engines: Text Mining approaches helps a lot in improving the query search criteria in the search engines and improves the information retrieval systems.

Section (e): Conclusion and Future work

Challenges give the future direction of work to be done in improvement of text mining of things. Some of the challenges are listed below[7]:

I. Heterogeneity of data is one of the biggest challenges for text miners to deal.

II. Usage of multiple languages and variety of notations used for a same word causes confusion for text miners and makes task difficult.

III. Sparse data causes over fitting of data into clusters and to different classes and causes wrong analysis for the analysts.

IV. Domain experts are required for the textual data to give a proper analysis and which may not always be possible to have a good domain expert.

V. Proper Statistical analysis of data is difficult to give as the data cannot always be good data, it may have a lots of missing values, so a standard results cannot be given.

Based on the above mentioned challenges, the text miners can work on improving the algorithms for the above defined tasks. Every area still requires efforts in improvement.

References

[1] Charu C. Aggarwal and ChengXiang Zhai: Survey of Text Classification Algorithm, chapter in book “Mining Text Data” DOI: 10.1007/978-1-4614-3223-4_6, pp 163-222-springer US 2012.

[2] S. B. Kotsiantis: Decion Trees: A recent Overview, article published in “Artificial Intelligence Review”, in April 2013, Volume 39, Issue 4, pp 261–283-springer.

[3] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain: Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks,- Expert Systems with Applications Volume 41, Issue 4, Part 2, March 2014, Pages 1937–1946– Elsevier.

[4] R Moraes, J. O. F Valiati, W. P. G. O. Neto: Document-levelsentiment classification: An empirical comparison between SVM and ANN, Expert Systems with Applications Volume 40, pp 621–633 2013 – Elsevier.

[5] Thorsten Joachims: Text categorization with Support Vector Machines: Learning with many relevant features, Support Vector Learning, Machine Learning: ECML-98,Volume 1398 of the series Lecture Notes in Computer Science pp 137-142.

[6] V Bijalwan, V Kumar, P Kumari, J Pascua: KNN based Machine Learning Approach for Text and Document Mining, International Journal of Database Theory and Application Vol.7, No.1 (2014), pp.61-70.

[7] A. Ittoo, L. M. Nguyen, A. van den Bosch: Text analytics in industry: challenges, desiderata and trends, Computers in Industry,Volume 78, May 2016, pp 96-107 - Elsevier.

[8] Gary Miner, John Elder: Practical text mining and statistical analysis of text mining, IV, Thomas Hill, ist edition, ISBN-978-0-386979-1, 2012 - books.google.com.

[9] Li-Ping Jing, Hou-kuan, Hong Boshi: Improved feature selection approach using TF-IDF in Text Mining, in Proceedings of the first Internationl conference on Machine Learning and cybermetics, Bejing, pp-944 to 946, 4-5 November 2002-IEEE.

[10] Charu C. Aggarwal and ChengXiang Zhai: A Survey of Text Clustering Algorithms chapter in book “Mining Text Data”, DOI 10.1007/978-1-4614-3223-4_6, pp 77-128 springer US 2012.

[11] Rashmi Agrawal: K-Nearest Neighbor for Uncertain Data, in International Journal of Computer Applications (0975 – 8887) Volume 105 – No. 11, November 2014.

[12] Rashmi Agrawal, Mridula Batra: A Detailed Study on Text Mining Techniques, in International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.