

Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example)

Gaurav Gupta
Assistant Professor
Department of Computer Engineering
University College of Engineering,
Punjabi University
Patiala (Punjab), India

Sumit Malhotra
Assistant Professor
Department of Computer Science and
Engineering
Bhai Gurdas College of Engineering and
Technology,PTU
Sangrur (Punjab), India

ABSTRACT

Text mining, at times alluded to as content information mining, is harshly equal to content investigation, which alludes to the procedure of determining astounding data from content. RapidMiner is unquestionably the world-leading open-source system for data mining. It is available as a stand-alone application for data analysis and as a data mining engine for the integration into own products. Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The word frequency counter allows you to count the frequency usage of each word in your document. Applying tokenization and word frequency counter for a text document (resume in this case) helps us find out occurrence of each word in a document but there is no provision to find a particular word frequency occurrence according to user choice.

Keywords

RapidMiner, RapidMiner Text Processing, RapidMiner Process Document from File operator, RapidMiner Transform case operator, RapidMiner Tokenize operator.

1. INTRODUCTION

1.1 Text Mining and Analysis

Boundless measures of new data and information are produced ordinary through investment, scholarly and social exercises. This ocean of information, anticipated to expand at a rate of 40% p.a., has noteworthy potential financial and societal quality. Organizations utilize such strategies to dissect client and contender information to enhance intensity; the pharmaceutical industry mines patents and research articles to improve drug discovery; within academic research, mining and analytics of large datasets are delivering efficiencies and new knowledge in areas as diverse as biological science, particle physics and media and communications. As the volume of insightful yield expands, we perceive that researchers are increasingly interested in using tools such as Text mining to explore patterns and trends across large databases of content. Text mining transposes words and expressions into numerical qualities.

Text analysis involves information retrieval, distributions, lexical examination to study word recurrence cognizance, labeling/annotation, data extraction, information mining methods including connection and affiliation investigation, visualization, and prescient examination. The overall

objective is, basically, to transform content into information for investigation.

1.2 Rapid Miner

Rapid-miner is certainly the world-heading open-source framework for information mining. It is accessible as a stand-alone application for information investigation and as a data mining engine for the integration into own products. Rapid Miner uses a client/server model with the server offered as Software as a Service or on cloud infrastructures. RapidMiner provides data mining and machine learning procedures including: data loading and transformation (Extract, transform and load (ETL)), data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment.

Rapidminer is composed in the Java programming dialect. RapidMiner provides a GUI to design and execute analytical workflows. Those workflows are called "Process" in RapidMiner and they consist of multiple "Operators". Each operator is performing a single task within the process and the output of each operator forms the input of the next one. Alternatively, the engine can be called from other programs or used as an API. Individual functions can be called from the command line. RapidMiner provides learning schemes and models and algorithms from Weka and R scripts that can be used through extensions.

1.3 Features

1. Open source and Operating system independence.
2. Compelling high-dimensional plotting facilities with Multi-layered data view concept ensures efficient data handling.
3. Including features of WEKA data mining tool.
4. Access information from database like Excel, Access, Oracle, IBM Db2, Microsoft SQL, Sybase and so forth.
5. Provision of nesting operator chains for complex tasks.

2. IMPLEMENTATION

2.1 Process Document from Files

Consider a resume in text format as shown in Figure 2.1.

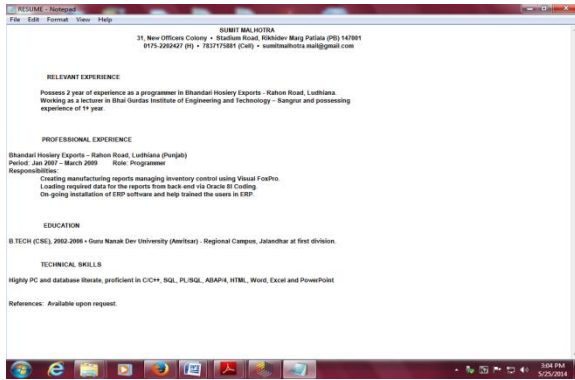


Figure 2.1 - Sample Resume to be processed in .txt format.

Process Document from Files operator in RapidMiner generates word vectors from a text collection stored in multiple files. In text directories arbitrary directories can be specified. All files matching the given file ending will be loaded and assigned to the class value provided with the directory. As shown in Figure. 2.2. Directory containing resume file in .txt format shown in Figure 2.1 is selected for processing.

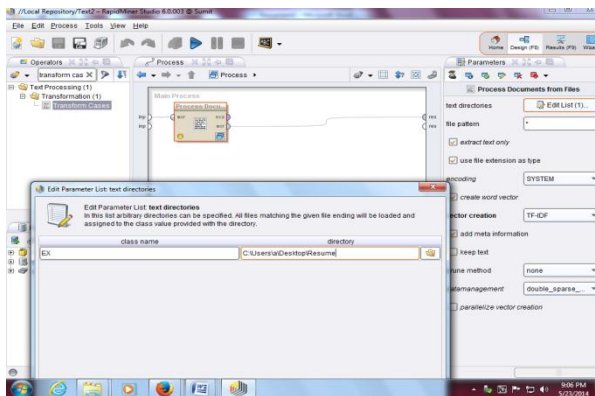


Figure 2.2 - Process Document Operator and text document selection.

Double click Process Document from Files operator to add components Transform cases and Tokenize as shown below in Figure 2.3.

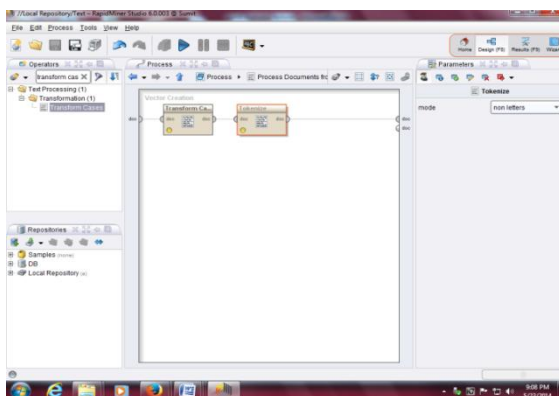


Figure 2.3 – Transform case and tokenize operator interconnection.

2.2 Transform cases

This operator transforms all characters in a document to either lower case or upper case, respectively as shown in Figure 2.4. Transform case is necessary in order to avoid confusion between similar words that differ in lowercase or uppercase. For example - 'exception' and 'Exception'. The "doc" node of the process to the "doc" input node of the operator

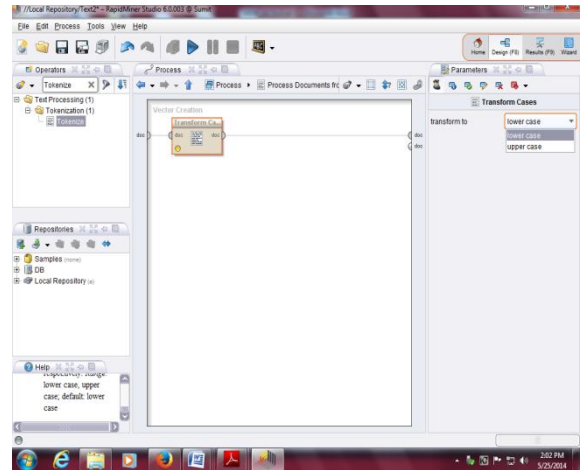


Figure 2.4 - Transform case operator with lower case mode selection.

In this case and by default also lower case is chosen from parameter transform to.

2.3 Tokenize

This operator splits the text of a document into a sequence of tokens. There are several options how to specify the splitting points. The default setting is non-letter that will result in tokens consisting of one single word and it's frequency of occurrence. Other modes available for Tokenize are specifying character, regular expression, linguistic sentences and linguistic tokens. In this case the mode opts is non-letter. If you are going to build windows of tokens or something like that, you will probably split complete sentences, this is possible by setting the split mode to specify character and enter all splitting characters. Default character is ' '. The third option let's you define regular expressions and is the most flexible for very special cases.

After you have inserted new operators, you can interconnect the operators inserted as shown in Figure 2.5, where the results of Transform case will be transferred to Tokenize in the form of document.

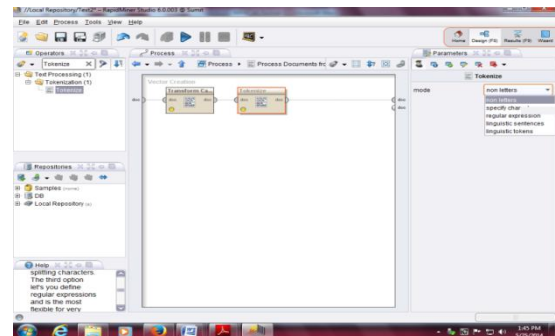
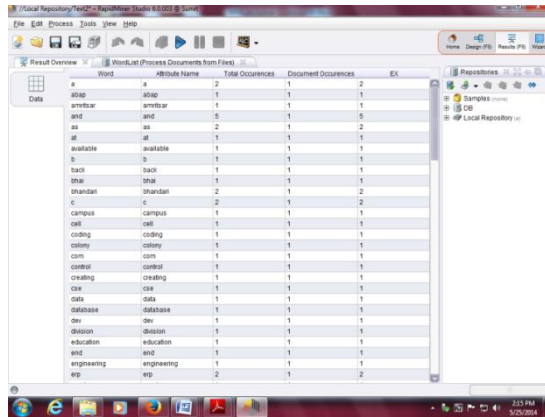


Figure 2.5 - Tokenize operator with non letter mode selection.

On clicking to run button the result is displayed as shown in Figure 2.6 where the occurrence of each keyword in a document is displayed in numeric format. The result is displayed in sorted order by default.



The screenshot shows the RapidMiner interface with a table of words and their frequency counts. The table has columns for Word, Absolute Name, Total Occurrences, and Document Occurrences. The words are sorted by Total Occurrences in descending order.

Word	Absolute Name	Total Occurrences	Document Occurrences
a	a	2	1
abad	abad	1	1
admitar	admitar	1	1
and	and	5	5
an	an	2	1
an	an	1	1
available	available	1	1
b	b	1	1
back	back	1	1
band	band	1	1
bandar	bandar	2	2
c	c	2	2
campus	campus	1	1
cell	cell	1	1
coding	coding	1	1
copy	copy	1	1
com	com	1	1
control	control	1	1
creating	creating	1	1
cia	cia	1	1
data	data	1	1
database	database	1	1
dev	dev	1	1
decision	decision	1	1
education	education	1	1
end	end	1	1
engineering	engineering	1	1
erp	erp	2	2

Figure 2.6 – Words and their frequency count.

3. CONCLUSION AND FUTURE SCOPE

In this paper, word frequency count of text document is done using RapidMiner tool - Transform case and Tokenize operators, with their interconnection. In order to find the frequency of occurrence of particular word the user has to scroll the scrollbar. There should be provision for the user to find the frequency of occurrence of particular keyword of interest by specifying the keyword of interest.

Moreover there should be provision to compare two text documents by comparing their keyword frequency occurrence. In case the text document is a resume, the comparison can be used to find the better candidate, by searching required keyword frequency occurrence.

4. REFERENCES

- [1] Textminingfromhttp://en.wikipedia.org/wiki/Text_mining.
- [2] RapidMinerfrom<http://en.wikipedia.org/wiki/RapidMiner>.
- [3] RapidMinerStudiofromhttp://rapidminer.com/product_s/_rapidminer-studio/.
- [4] To find frequency of the words using RapidMiner(2012).Retrieved June 22, 2012, from <http://gunjanaaggarwal.blogspot.in/2012/07/words-frequency-text-analytics.html>.
- [5] Value and benefits of text mining from <http://www.jisc.ac.uk/reports/value-and-benefits-of-text-mining>.
- [6] Tanu Verma,Renu,Deepti Gaur,"Tokenization and FilteringProcess in RapidMiner", International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 ,Volume 7– No. 2, April 2014.
- [7] Jordan Shterev,"Demo: Using RapidMiner for Text Mining",Digital Presentation and Preservation of Cultural and ScientificHeritage (Digital Presentation and Preservation of Cultural andScientific Heritage), issue: III / 2013, pages: 254256
- [8] TipawanSilwattananusarnand Assoc.Prof. Dr.KulthidaTuamsuk,"Data Mining and Its Applications for KnowledgeManagement::A Literature Review from 2007to 2012"International Journal of Data Mining & KnowledgeManagement Process(IJDKP) Vol.2, No.5, September 2012.