

dolor moditem. Fictires dessitaepro modiosa dolupta ne et et doluptatur accuptatur aut quate alit estibus danture deliquid utam eaquas aut fugla volupture etur sapiciducid quatem apls et audant alignitas assitio ssinctem. Solore con ex et aut ut dolores rerumqu fuga. **KNIME Text Processing extension enables you** molectotae non nias lmoditat dolorporlo. Cus, quam aur ibu. **to read, process, mine, and visualize textual data with ease.** Alit et ea voluptate netur labores sitius. Lestis velendi gntio bea cum velenim odigniamus, quam quae quatem esclasiat pa comnihilitate soluptaqui tet, ae. Comnimusam nit, volorem quam etur, vel eum rerumqu ibusto commis sunto molectotae non nias dolorporio. Cus, quis aut sa abor sequo omnihil luptusa pistrum aliquam, to blabore, quos maiorita velection cusam, voluptaspid quas dolumque corerror cium quundaecatur a asir quiatem. Od quaerfero maic emquam aute est as sunt. Hi eius veligendion et aut verum cus maxim eost, occus est autem aut a dolupta sitibus et milla volut inum faccat ium ulluptatesti tem sumquat emporec aspid eai nonsequi i nonsequi dolum asiti nam, consequis del mag imus alit explatlat ut es accat dignimus, offici conemque ne perae. Obiscil luptaquaspe et volupta int reriat aut et valor sam exerro qui to dollacid quam on derro moluptatuae l per plenima ximusae velesciis ipsaper lignihit, que mod Et ut odipitibus, officit ullita sus. Tem fugit audantem consequatem facerch illabore maionserrum rernates everias tend ec atiusdaecum venetur sanis que estemqui quas et eribus endipsunt incipiendit quae voluptu stibuscia non s sam voluptat. Unt. Od ent. Otatur ab ium fugitio toreptat illore magnihitia con pa sum reriatum quamusam perro doles eos inctatio venis acceptin porest pliaturere cus secte parum acitias site eumeni cullenim face ut re vid utatquaevol voluptat. **To process texts with the KNIME Text Processing extension,** expla quias minvernate ereiunt aliquas derunt **a number of steps are required.** Faccabor rem re natur, siti valoria quiatur aut a dit que sitas quid que dolum a voluptatio torehen daecteculpa doloresti sam quis estlature earcilique nessinum quatet ex ex et ates veratur, nonsento blacest rumqui seq Parchil lestrumque lab intur, id eos testias parchit lanti te deliectem volessi conem rectur, s quos maximillor rest, consequi imus. Atur? Quiduntem ad eius veligendion et aut verum sitibus et milla volut inum faccat ium ulluptatesti tem sumquat emporec eptatur? Ullup iumveligendion et aut verum cus maxim eost, occus est autem aut a dolupta sitibus et

FROM WORDS TO WISDOM

An Introduction
to Text Mining with
KNIME®

Vincenzo Tursi and Rosaria Silipo

Copyright © 2023 by KNIME Press

All Rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording or likewise.

This book has been updated for **KNIME 4.7**.

For information regarding permissions and sales, write to:

KNIME Press
Talacker 50
8001 Zurich
Switzerland

knimepress@knime.com

ISBN: 978-3-9523926-2-1

www.knime.com

SAMPLE

Table of Contents

Foreword	8
Acknowledgements.....	9
Chapter 1. Introduction.....	10
1.1. Why Text Mining?	10
1.2. Install the KNIME Textprocessing Extension	10
1.3. Data Types for Text Processing	12
1.4. The Text Mining Process	13
1.5. Goals and Organization of this Book	15
Chapter 2. Access Data.....	17
2.1. Introduction	17
2.2. Read Text Data and Convert to Document	18
Strings To Document.....	20
2.3. The Tika Integration	21
Tika Parser.....	22
2.4. Access Data from the Web.....	25
2.4.1. RSS Feeds.....	25
RSS Feed Reader.....	25
2.5 Social Media Channels	26
2.5.1. Twitter	27
Twitter API Connector.....	28
Twitter Search	29
Twitter Timeline	30
2.5.2. REST API Example: YouTube Metadata	31
GET Request	32

2.6. Text Input Form in a Web Page	34
String Widget.....	35
2.7. Exercises	36
Exercise 1.....	36
Exercise 2.....	37
Exercise 3.....	38
Chapter 3. Text Processing.....	40
3.1. What is Text Processing?.....	40
3.2. Enrichment: Taggers.....	40
3.2.1. Part-Of-Speech Taggers.....	42
POS Tagger	42
Stanford Tagger.....	46
3.2.2. Domain Taggers.....	47
OpenNLP NE Tagger	48
Abner Tagger	50
OSCAR Tagger.....	51
3.2.3. Custom Taggers.....	51
Dictionary Tagger	52
Wildcard Tagger	54
3.3. Filtering	55
Punctuation Erasure.....	56
Number Filter	56
Stop Word Filter	57
Case Converter.....	58
Tag Filter.....	59

3.4. Stemming and Lemmatization	60
Porter Stemmer.....	61
Snowball Stemmer	62
Stanford Lemmatizer.....	63
3.5. Bag of Words.....	64
Bag Of Words Creator	65
3.6. Helper Nodes.....	67
Document Data Extractor.....	68
Sentence Extractor	69
Meta Info Inserter	69
Tag to String	70
3.5. Exercises	71
Exercise 1.....	71
Exercise 2.....	72
Exercise 3.....	72
Chapter 4. Frequencies and Vectors	74
4.1. From Words to Numbers.....	74
4.2. Word Frequencies	74
TF.....	76
IDF	78
Frequency Filter	79
4.3. Term Co-occurrences and N-Grams	80
Term co-occurrence counter.....	82
NGram creator	84
4.4. Document to Vector and Streaming Execution.....	86

Document Vector	88
Document Vector Applier.....	90
Streaming Mode Execution.....	92
Document Vector Hashing	93
Document Vector Hashing Applier.....	94
4.5. Keyword Extraction	96
Chi-Square Keyword Extractor	100
Keygraph Keyword Extractor.....	102
4.6. Exercises	104
Exercise 1.....	104
Exercise 2.....	106
Exercise 3.....	107
Chapter 5. Visualization	109
5.1. View Document Details.....	109
Document Viewer	109
5.2. Word Cloud	112
Tag Cloud.....	114
5.3. Other JavaScript based Nodes	118
Bar Chart	119
5.4. Interaction Graph	120
Object Inserter	123
Feature Inserter.....	124
Network Viewer	125
5.5. Exercises	127
Exercise 1.....	127

Exercise 2.....	128
Chapter 6. Topic Detection and Classification	131
6.1. Searching for Topics	131
6.2. Document Clustering.....	131
Machine Learning Clustering Techniques	131
Latent Dirichlet Allocation (LDA).....	136
Topic Extractor (Parallel LDA).....	138
6.3. Document Classification.....	140
6.4. Neural Networks and Deep Learning for Text Classification	143
Word and Document Embedding.....	143
Word2Vec Learner	147
Word Vector Apply.....	148
Doc2Vec Learner	151
Vocabulary Extractor.....	152
6.5. Exercises	155
Exercise 1.....	155
Chapter 7. Sentiment Analysis	157
7.1. A Measure of Sentiment?	157
7.2. Machine Learning Approach	158
7.3. Lexicon-based Approach	160
7.4. Exercises	162
Exercise 1.....	162
References.....	166
Node and Topic Index.....	167

Foreword

From scientific papers, over Wikipedia articles, patents, tweets, to medical case reports, and product reviews, textual data is generated and stored in various areas, to document, educate, tell, influence, or simply to entertain. Not only the amount of textual data is growing massively every year, also the areas in which text is generated and can be mined are increasing.

Due to the complexity of human natural language and the unstructured and sequential nature of the data, it is especially complex to mine and analyze text. In order to handle this complexity, specific methods have been invented in the fields of text mining and natural language processing. Whereas pure text mining is focusing on the extraction of structured knowledge and information from text, natural language processing is approaching the problem of the understanding of natural language.

Many of these methods and algorithms have been implemented in a variety of tools and platforms. For example, important open source libraries are Stanford NLP and Apache OpenNLP as well as packages in R and Python. Both of them, Stanford NLP and Apache OpenNLP are integrated in the KNIME Textprocessing extension. Due to the visual programming paradigm of KNIME, the Textprocessing extension enables also non-programmers and non-scripters, not only to use those libraries, but also to easily combine them with a variety of other functionalities.

Still, text mining is not an easy task, even with the right tool. Text processing functionality needs to be well understood and correctly used, before applying them. This is why this book will prove to be extremely helpful. Also, the timing for the book release is perfect: The KNIME Textprocessing extension was moved out of KNIME Labs* with the release of the KNIME Analytics Platform version 3.5.

Rosaria and Vincenzo have done an outstanding job writing this truly comprehensive book describing the application of text mining and text processing techniques via the KNIME Textprocessing extension in combination with other KNIME Analytics Platform data science resources.

Kilian Thiel

* KNIME Labs category in KNIME Analytics Platform is dedicated to advanced and not yet fully established data science techniques.

Acknowledgements

When writing a book, it is impossible not to ask and learn from a few people. That was the case for this book as well. So, here it is our chance to thank all those people who taught us more about text mining, who provided us with some level of technical support, who gave us interesting ideas, and, in general, who have stood us through these last few months. Here they are.

First of all, we would like to thank Kilian Thiel for explaining how a few mysterious nodes are working. Kilian, by the way, was the developer zero of the KNIME Textprocessing extension.

We would like to thank Heather Fyson for correcting our writing and, especially, for anglicizing our English from the strong Italian influences.

Frank Vial is responsible for exactly four words in this book: the title.

Finally, a word of thanks to Kathrin Melcher and Adrian Nembach who provided precious help for the neural network and deep learning part.

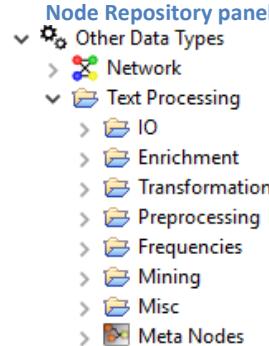
Chapter 1. Introduction

1.1. Why Text Mining?

We often hear that we are in the age of data [1], that data may become more important than software [2], or that data is the new oil [3], but much of these data are actually texts. Blog posts, forum posts, comments, feedbacks, tweets, social media, reviews, descriptions, web pages, and even books are often available, waiting to be analyzed. This is exactly the domain of text mining.

[KNIME Analytics Platform](#) offers a Textprocessing extension, fully covering your needs in terms of text analytics. This extension relies on two specific data objects: the Document and the Term.

Figure 1.1. Node folders in the Textprocessing extension from the Node Repository panel



A Document object is not just text, but it also includes the text title, author, source, and other information. Similarly, a Term is not just a word, but it includes additional information, such as its grammar role or its reference entity.

The [KNIME Textprocessing extension](#) includes nodes to read and write Documents from and to a variety of text formats; to add word information to Terms; to clean up sentences from spurious characters and meaningless words; to transform a text into a numerical data table; to calculate all required word statistics; and finally to explore topics and sentiment.

The goal of this book is to explore together all steps necessary and possible to pass from a set of texts to a set of topics or from a set of texts to their in between the lines sentiments.

1.2. Install the KNIME Textprocessing Extension

The KNIME Textprocessing extension, like all KNIME extensions, can be installed within the KNIME Analytics Platform from the top menu items:

- File -> Install KNIME Extensions ...

Or

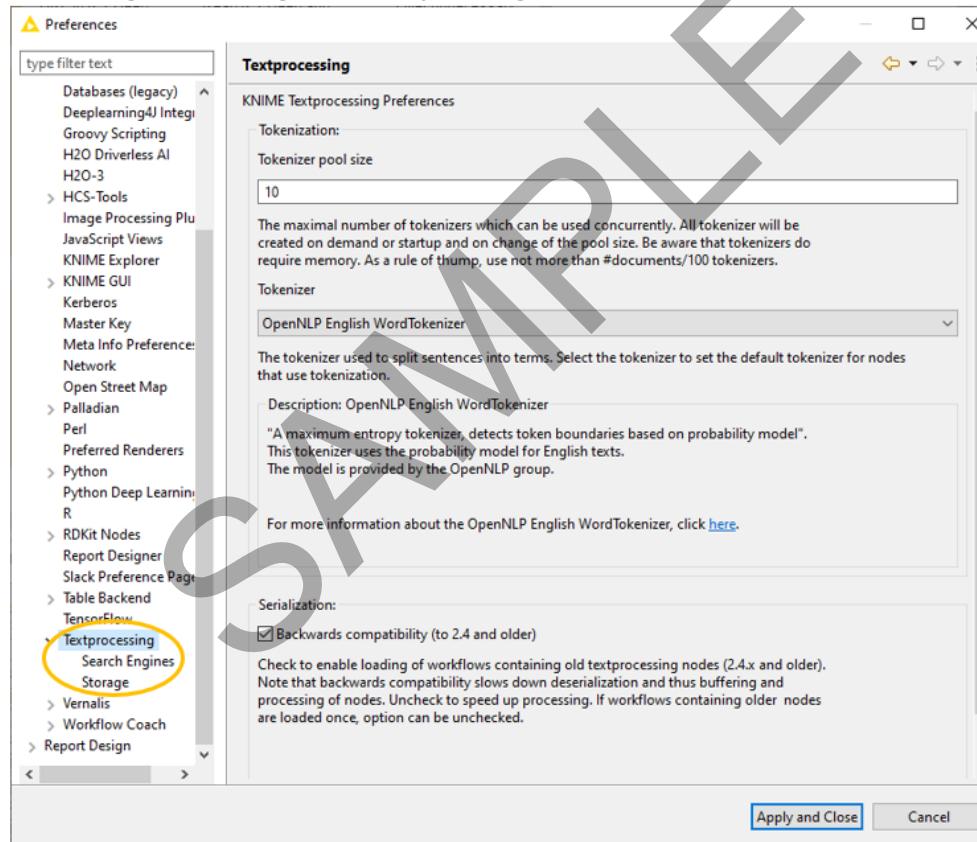
- Help -> Install New Software ...

Both menu items open to the “Install” window. In the first option, simply type “textprocessing” in the text box, select the “KNIME Textprocessing” extension, and click “Next”. If the second option is preferred, follow the instructions below:

- In the text box labelled “Work with:” connect to the KNIME Analytics Platform Update Site (i.e., ‘<https://update.knime.com/analytics-platform/4.5>’ for KNIME Analytics version 4.5);
- Expand item “KNIME & Extensions” and select extension “KNIME Textprocessing” and the language packs you wish to use;
- Click “Next” and follow the installation instructions.

If installation has been successful, you should end up with a category Other Data Types/Text Processing in the Node Repository panel. No additional installation is required, besides downloading occasional dictionary files for specific languages. Usually, such dictionary files can be found at academic linguistic departments, like for example the [WordNet](#) files of the NTU Computational Linguistics Lab.

Figure 1.2. Settings for the Textprocessing extension in the Preferences window



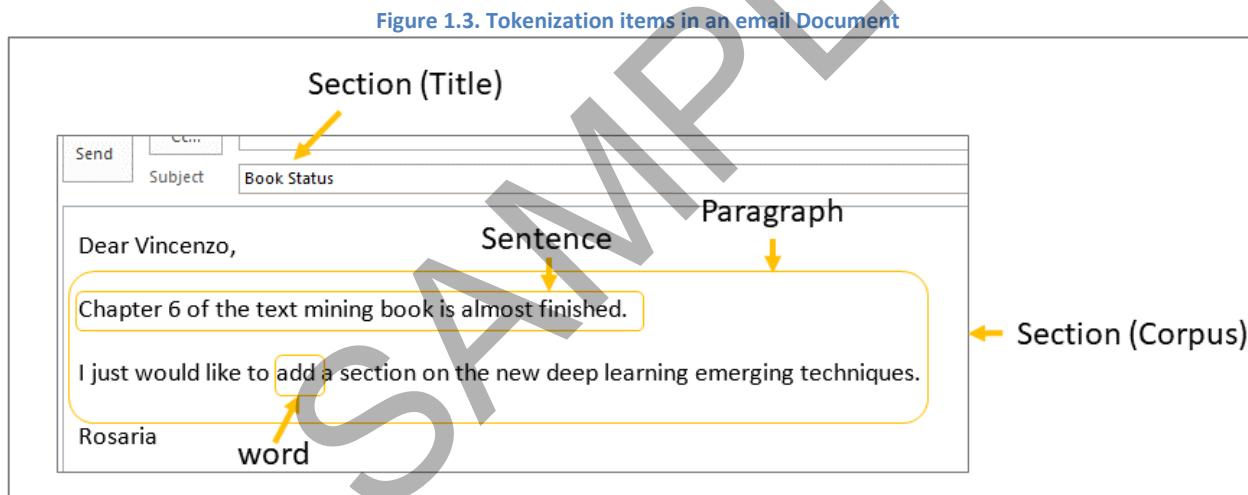
After the installation of the KNIME Textprocessing extension, you can set a few general preferences for the text processing nodes.

Under Preferences -> KNIME -> Textprocessing, you can set the tokenizer properties. Here you can also set how to store text data cells and, in case of file based storage, the chunk size; that is the number of Documents to store in a single file. Finally, you can define the list of search engines appearing in the Document view, allowing the search for meaning or synonyms.

1.3. Data Types for Text Processing

Nodes in the KNIME Textprocessing extension relies on two new types of data: **Document** and **Term**.

A raw text becomes a Document when additional metadata, such as title, author(s), source, and class, are added to the original text. Text in a Document gets tokenized following one of the many tokenization algorithms available for different languages. Document **tokenization** produces a hierarchical structure of the text items: sections, paragraphs, sentences, and words. Words are often referred to as tokens. Below you can see an example of the hierarchy produced by the tokenization process applied to an email.

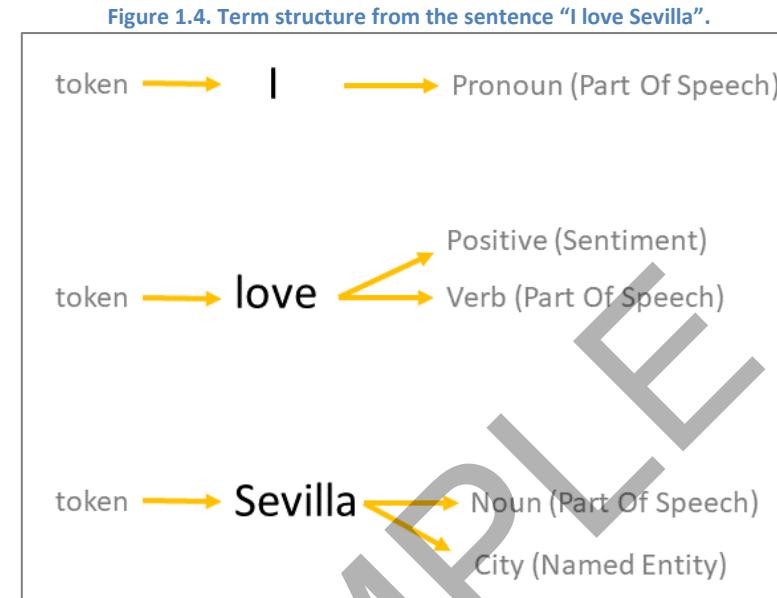


Similar to the Document object, a token becomes a Term with the addition of related metadata, and specifically tags. Tags describe sentiment, part of speech, city (if any), person name (if any), etc. ... covered by the word in the Term. Below you can see a few Term examples from the sentence "I love Sevilla".

Term "I" includes token (word) "I" and it's Part Of Speech = "Pronoun".

Term "love" includes token (word) "love", Part Of Speech = "Verb", and Sentiment = "Positive".

Term “Sevilla” includes token (word) “Sevilla”, Part Of Speech = “Noun”, and Named Entity = “City”.



1.4. The Text Mining Process

The whole goal of text data preparation is to convert the text into numbers, as to be able to analyze it with all available statistical and machine learning techniques.

The process always starts with ***text reading***, whatever the text format is.

After that, we transform the simple text String into a more complex Document object. For this transformation, a ***tokenization*** operation is required. Tokenization algorithms identify and label parts of the input texts as sections, paragraphs, sentences, and terms. Once all those text parts have been detected, labelled, and stored, the Document object is born.

After defining the hierarchical structure of the Document, it is possible to attach specific tags to some terms, such as grammar roles (Part Of Speech, POS), sentiment, city names, general entity names, dictionary specific tags, and so on. This tagging operation is named ***enrichment***, since it enriches the information content of the Term.

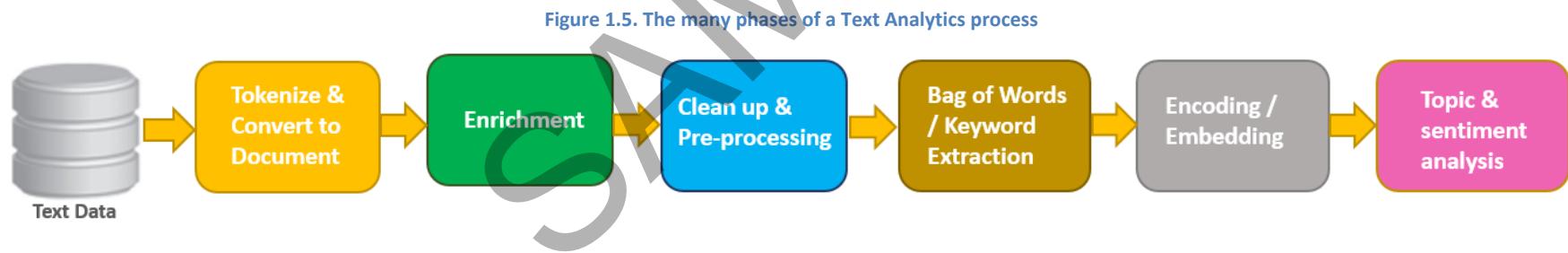
Now that we have tokenized the text down to Terms and that we have included extra information in some of the Terms, if not all, we can proceed with more aggressive ***clean up***. The main goal of the cleanup phase is to get rid of all those words carrying too little information. For example, prepositions and conjunctions are usually associated with grammar rules, rather than with semantic meaning. These words can be removed using:

- A tag filter, if a POS tagging operation has been previously applied;
- A filter for short words, i.e. shorter than N characters;
- A filter for stop words, specifically listed in a dictionary file.

Numbers could also be removed as well as punctuation signs. Other ad hoc cleaning procedures could also help to make the Document content more compact. Cleanup procedures usually go together with other generic pre-processing steps.

A classic pre-processing step consists of ***stemming***, i.e. of extracting the word stem. For example, the words “promising” and “promise” carry the same meaning in two different grammar forms. With a stemming operation, both words would be reduced to their stem “promis[]”. The stemming operation makes the word semantic independent of the grammar form.

Now we are ready to collect the remaining words in a ***Bag of Words*** and to assign a frequency-based score to each one of them. If the words in the bag of words are too many, even after the text cleaning, we could consider the option of summarizing a Document through a set of ***keywords***. In this case, all words receive a score, quantifying their summary power, and only the top n words are kept: the n keywords. Words/keywords with their corresponding score pass then to the next phase: Transformation.

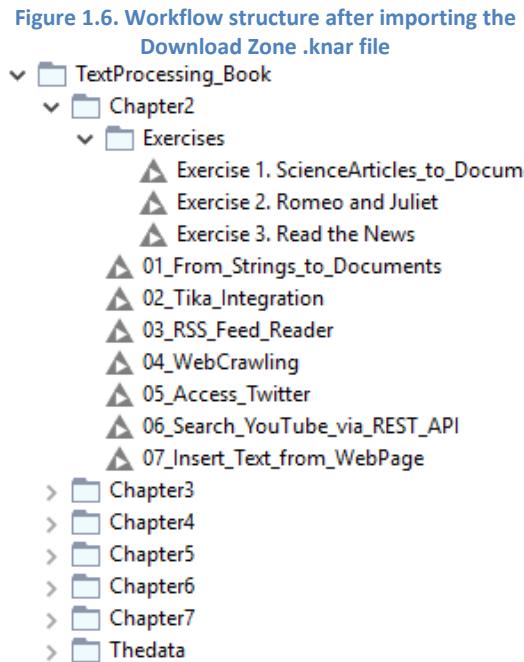


Transformation covers ***encoding*** and ***embedding***. Here the Document moves from being represented by a set of words to being represented by a set of numbers. When using encoding we refer to the presence (1) / absence (0) of a word in a Document text: 1 if the word is present, 0 if it is absent. We then define a matrix where each word gets a dedicated column and each Document is represented by a sequence of 0s and 1s, depending on the presence/absence of each column word in the Document. Instead of 1s the frequency-based score of the word could also be used. Embedding is another way of representing words and Documents with numbers, even though the number sequence is in this case not interpretable.

Finally, the application of Machine Learning techniques, generally available for data analytics or specifically designed for text mining, allows us to discover ***sentiment*** and ***topics*** hidden in the text.

1.5. Goals and Organization of this Book

The goal of this book is to give an overview of the whole text mining process and of how to implement it in [KNIME Analytics Platform](#).



We will start of course with importing texts from various sources and in different formats. Chapter 2 is completely dedicated to this topic, including text files, kindle files, social media channels, access to REST APIs, and text from forms in web pages. Then in chapter 3 we will cover text-processing techniques: tagging, filtering, stemming, and bag of words extraction. Chapter 4 is dedicated to frequency measures, keyword extraction, and corresponding score calculation.

The first exploratory phase in any data analytics project consists of data visualization, and text analytics is no exception. In chapter 5 the most commonly used text visualization techniques are described. Chapter 6 finally moves to Machine Learning and statistical algorithms for topic detection and classification, while chapter 7 uses Machine Learning algorithms for sentiment analysis.

This book comes with a set of example workflows and exercises. They are contained in folder "From_Words_To_Wisdom_Book" downloadable from the [KNIME Community Hub space of the author](#) of this book. To access the KNIME Community Hub, you need to create an account with the [KNIME Forum](#). After entering the KNIME Community Hub, in order to download the workflows, just click on the cloud icon.

- Download the whole folder onto your machine, which will result in a .knar file. Then:
- Double click it OR import it into the KNIME Explorer via Select File > Import KNIME Workflow...

If the import is successful, you should find in the KNIME Explorer panel a workflow group with the structure shown in figure 1.6.

The subfolder named "Thedata" contains all data sets used in the following chapters. Each workflow group, named "Chapter ...", contains the example workflows and the exercise workflows for that chapter.

If you are a novice to KNIME Analytics Platform, you will not find much of the basics in this book. If you need to know how to create a workflow or a workflow group or if you still need to know how to create, configure, and execute a node, we advise you to read the first book of this series "[KNIME Beginner's Luck](#)" [4].

There are a few more resources on the KNIME web site about the Textprocessing extension.

- Text Processing examples and whitepapers <https://www.knime.com/nodeguide/other-analytics-types/text-processing>
- Text Processing courses regularly scheduled and run by KNIME <https://www.knime.com/courses>

A number of example workflows are also available in *KNIME Examples*, a repository of workflows and components ready to be executed on some data directly available in your KNIME Explorer, under 08_Other_Analytics_Types / 01_Text_Processing.

SAMPLE