TURING 图灵原版计算机科学系列

CAMBRIDGE

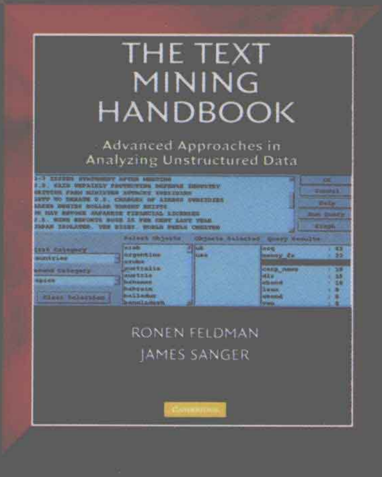
The Text Mining Handbook

Advanced Approaches in Analyzing Unstructured Data

文本挖掘

（英文版）

[以] Ronen Feldman
[美] James Sanger 著



THE TEXT MINING HANDBOOK

Advanced Approaches in Analyzing Unstructured Data

RONEN FELDMAN
JAMES SANGER

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原版计算机科学系列

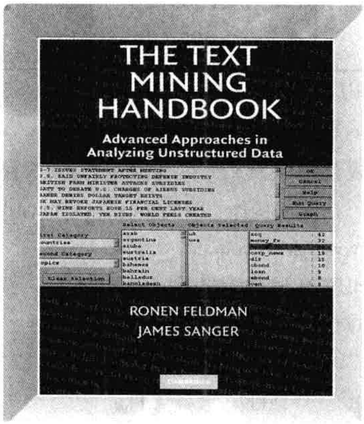
The Text Mining Handbook

Advanced Approaches in Analyzing Unstructured Data

文本挖掘

（英文版）

[以] Ronen Feldman
[美] James Sanger 著



THE TEXT MINING HANDBOOK

Advanced Approaches in Analyzing Unstructured Data

RONEN FELDMAN
JAMES SANGER

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

文本挖掘＝The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data：英文/（以）费尔德曼（Feldman, R.），（美）桑格（Sanger, J.）著. —北京：人民邮电出版社，2009.8
（图灵原版计算机科学系列）
ISBN 978-7-115-20535-3

I. 文… II. ①费… ②桑… III. 数据采集－研究－英文 IV. TP274

中国版本图书馆CIP数据核字（2009）第104203号

内容提要

本书是一部文本挖掘领域名著，作者为世界知名的权威学者。书中涵盖了核心文本挖掘操作、文本挖掘预处理技术、分类、聚类、信息提取、信息提取的概率模型、预处理应用、可视化方法、链接分析、文本挖掘应用等内容，很好地结合了文本挖掘的理论和实践。

本书非常适合文本挖掘、信息检索领域的研究人员和实践者阅读，也适合作为高等院校计算机及相关专业研究生的数据挖掘和知识发现等课程的教材。

图灵原版计算机科学系列

文本挖掘（英文版）

◆ 著 [以] Ronen Feldman [美] James Sanger
责任编辑 杨海玲
◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷
◆ 开本：800×1000 1/16
印张：26.5
字数：506千字 2009年8月第1版
印数：1-2 000册 2009年8月北京第1次印刷
著作权合同登记号 图字：01-2009-2918号
ISBN 978-7-115-20535-3/TP

定价：69.00元

读者服务热线：（010）51095186 印装质量热线：（010）67129223
反盗版热线：（010）67171154

版 权 声 明

Preface

The information age has made it easy to store large amounts of data. The proliferation of documents available on the Web, on corporate intranets, on news wires, and elsewhere is overwhelming. However, although the amount of data available to us is constantly increasing, our ability to absorb and process this information remains constant. Search engines only exacerbate the problem by making more and more documents available in a matter of a few key strokes.

Text mining is a new and exciting research area that tries to solve the information overload problem by using techniques from data mining, machine learning, natural language processing (NLP), information retrieval (IR), and knowledge management. Text mining involves the preprocessing of document collections (text categorization, information extraction, term extraction), the storage of the intermediate representations, the techniques to analyze these intermediate representations (such as distribution analysis, clustering, trend analysis, and association rules), and visualization of the results.

This book presents a general theory of text mining along with the main techniques behind it. We offer a generalized architecture for text mining and outline the algorithms and data structures typically used by text mining systems.

The book is aimed at the advanced undergraduate students, graduate students, academic researchers, and professional practitioners interested in complete coverage of the text mining field. We have included all the topics critical to people who plan to develop text mining systems or to use them. In particular, we have covered preprocessing techniques such as text categorization, text clustering, and information extraction and analysis techniques such as association rules and link analysis.

The book tries to blend together theory and practice; we have attempted to provide many real-life scenarios that show how the different techniques are used in practice. When writing the book we tried to make it as self-contained as possible and have compiled a comprehensive bibliography for each topic so that the reader can expand his or her knowledge accordingly.

BOOK OVERVIEW

The book starts with a gentle introduction to text mining that presents the basic definitions and prepares the reader for the next chapters. In the second chapter we describe the core text mining operations in detail while providing examples for each operation. The third chapter serves as an introduction to text mining preprocessing techniques. We provide a taxonomy of the operations and set the ground for Chapters IV through VII. Chapter IV offers a comprehensive description of the text categorization problem and outlines the major algorithms for performing text categorization.

Chapter V introduces another important text preprocessing task called text clustering, and we again provide a concrete definition of the problem and outline the major algorithms for performing text clustering. Chapter VI addresses what is probably the most important text preprocessing technique for text mining – namely, information extraction. We describe the general problem of information extraction and supply the relevant definitions. Several examples of the output of information extraction in several domains are also presented.

In Chapter VII, we discuss several state-of-the-art probabilistic models for information extraction, and Chapter VIII describes several preprocessing applications that either use the probabilistic models of Chapter VII or are based on hybrid approaches incorporating several models. The presentation layer of a typical text mining system is considered in Chapter IX. We focus mainly on aspects related to browsing large document collections and on issues related to query refinement. Chapter X surveys the common visualization techniques used either to visualize the document collection or the results obtained from the text mining operations. Chapter XI introduces the fascinating area of link analysis. We present link analysis as an analytical step based on the foundation of the text preprocessing techniques discussed in the previous chapters, most specifically information extraction. The chapter begins with basic definitions from graph theory and moves to common techniques for analyzing large networks of entities.

Finally, in Chapter XII, three real-world applications of text mining are considered. We begin by describing an application for articles posted in *BioWorld* magazine. This application identifies major biological entities such as genes and proteins and enables visualization of relationships between those entities. We then proceed to the GeneWays application, which is based on analysis of *PubMed* articles. The next application is based on analysis of U.S. patents and enables monitoring trends and visualizing relationships between inventors, assignees, and technology terms.

The appendix explains the DIAL language, which is a dedicated information extraction language. We outline the structure of the language and describe its exact syntax. We also offer several code examples that show how DIAL can be used to extract a variety of entities and relationships. A detailed bibliography concludes the book.

ACKNOWLEDGMENTS

This book would not have been possible without the help of many individuals. In addition to acknowledgments made throughout the book, we feel it important to

take the time to offer special thanks to an important few. Among these we would like to mention especially Benjamin Rosenfeld, who devoted many hours to revising the categorization and clustering chapters. The people at ClearForest Corporation also provided help in obtaining screen shots of applications using ClearForest technologies – most notably in Chapter XII. In particular, we would like to mention the assistance we received from Rafi Vesserman, Yonatan Aumann, Jonathan Schler, Yair Liberzon, Felix Harmatz, and Yizhar Regev. Their support meant a great deal to us in the completion of this project.

Adding to this list, we would also like to thank Ian Bonner and Kathy Bentaieb of Inxight Software for the screen shots used in Chapter X. Also, we would like to extend our appreciation to Andrey Rzhetsky for his personal screen shots of the GeneWays application.

A book written on a subject such as text mining is inevitably a culmination of many years of work. As such, our gratitude is extended to both Haym Hirsh and Oren Etzioni, early collaborators in the field.

In addition, we would like to thank Lauren Cowles of Cambridge University Press for reading our drafts and patiently making numerous comments on how to improve the structure of the book and its readability. Appreciation is also owed to Jessica Farris for help in keeping two very busy coauthors on track.

Finally it brings us great pleasure to thank those dearest to us – our children Yael, Hadar, Yair, Neta and Frithjof – for leaving us undisturbed in our rooms while we were writing. We hope that, now that the book is finished, we will have more time to devote to you and to enjoy your growth. We are also greatly indebted to our dear wives Hedva and Lauren for bearing with our long hours on the computer, doing research, and writing the endless drafts. Without your help, confidence, and support we would never have completed this book. Thank you for everything. We love you!

Contents

I

Introduction to Text Mining

I.1 DEFINING TEXT MINING

Text mining can be broadly defined as a knowledge-intensive process in which a user interacts with a document collection over time by using a suite of analysis tools. In a manner analogous to data mining, text mining seeks to extract useful information from data sources through the identification and exploration of interesting patterns. In the case of text mining, however, the data sources are document collections, and interesting patterns are found not among formalized database records but in the unstructured textual data in the documents in these collections.

Certainly, text mining derives much of its inspiration and direction from seminal research on data mining. Therefore, it is not surprising to find that text mining and data mining systems evince many high-level architectural similarities. For instance, both types of systems rely on preprocessing routines, pattern-discovery algorithms, and presentation-layer elements such as visualization tools to enhance the browsing of answer sets. Further, text mining adopts many of the specific types of patterns in its core knowledge discovery operations that were first introduced and vetted in data mining research.

Because data mining assumes that data have already been stored in a structured format, much of its preprocessing focus falls on two critical tasks: Scrubbing and normalizing data and creating extensive numbers of table joins. In contrast, for text mining systems, preprocessing operations center on the identification and extraction of representative features for natural language documents. These preprocessing operations are responsible for transforming unstructured data stored in document collections into a more explicitly structured intermediate format, which is a concern that is not relevant for most data mining systems.

Moreover, because of the centrality of natural language text to its mission, text mining also draws on advances made in other computer science disciplines concerned with the handling of natural language. Perhaps most notably, text mining exploits techniques and methodologies from the areas of information retrieval, information extraction, and corpus-based computational linguistics.

I.1.1 The Document Collection and the Document

A key element of text mining is its focus on the *document collection*. At its simplest, a document collection can be any grouping of text-based documents. Practically speaking, however, most text mining solutions are aimed at discovering patterns across very large document collections. The number of documents in such collections can range from the many thousands to the tens of millions.

Document collections can be either *static*, in which case the initial complement of documents remains unchanged, or *dynamic*, which is a term applied to document collections characterized by their inclusion of new or updated documents over time. Extremely large document collections, as well as document collections with very high rates of document change, can pose performance optimization challenges for various components of a text mining system.

An illustration of a typical real-world document collection suitable as initial input for text mining is PubMed, the National Library of Medicine's online repository of citation-related information for biomedical research papers. PubMed has received significant attention from computer scientists interested in employing text mining techniques because this online service contains text-based document abstracts for more than 12 million research papers on topics in the life sciences. PubMed represents the most comprehensive online collection of biomedical research papers published in the English language, and it houses data relating to a considerable selection of publications in other languages as well. The publication dates for the main body of PubMed's collected papers stretch from 1966 to the present. The collection is dynamic and growing, for an estimated 40,000 new biomedical abstracts are added every month.

Even subsections of PubMed's data repository can represent substantial document collections for specific text mining applications. For instance, a relatively recent PubMed search for only those abstracts that contain the words *protein* or *gene* returned a result set of more than 2,800,000 documents, and more than 66 percent of these documents were published within the last decade. Indeed, a very narrowly defined search for abstracts mentioning *epidermal growth factor receptor* returned more than 10,000 documents.

The sheer size of document collections like that represented by PubMed makes manual attempts to correlate data across documents, map complex relationships, or identify trends at best extremely labor-intensive and at worst nearly impossible to achieve. Automatic methods for identifying and exploring interdocument data relationships dramatically enhance the speed and efficiency of research activities. Indeed, in some cases, automated exploration techniques like those found in text mining are not just a helpful adjunct but a baseline requirement for researchers to be able, in a practicable way, to recognize subtle patterns across large numbers of natural language documents.

Text mining systems, however, usually do not run their knowledge discovery algorithms on unprepared document collections. Considerable emphasis in text mining is devoted to what are commonly referred to as *preprocessing operations*. Typical text mining preprocessing operations are discussed in detail in Chapter III.

Text mining preprocessing operations include a variety of different types of techniques culled and adapted from information retrieval, information extraction, and computational linguistics research that transform raw, unstructured, original-format content (like that which can be downloaded from PubMed) into a carefully structured, intermediate data format. Knowledge discovery operations, in turn, are operated against this specially structured intermediate representation of the original document collection.

The Document

Another basic element in text mining is the *document*. For practical purposes, a document can be very informally defined as a unit of discrete textual data within a collection that usually, but not necessarily, correlates with some real-world document such as a business report, legal memorandum, e-mail, research paper, manuscript, article, press release, or news story. Although it is not typical, a document can be defined a little less arbitrarily within the context of a particular document collection by describing a *prototypical document* based on its representation of a similar class of entities within that collection.

One should not, however, infer from this that a given document necessarily exists only within the context of one particular collection. It is important to recognize that a document can (and generally does) exist in any number or type of collections – from the very formally organized to the very ad hoc. A document can also be a member of different document collections, or different subsets of the same document collection, and can exist in these different collections at the same time. For example, a document relating to Microsoft's antitrust litigation could exist in completely different document collections oriented toward current affairs, legal affairs, antitrust-related legal affairs, and software company news.

"Weakly Structured" and "Semistructured" Documents

Despite the somewhat misleading label that it bears as *unstructured data*, a text document may be seen, from many perspectives, as a structured object. From a linguistic perspective, even a rather innocuous document demonstrates a rich amount of semantic and syntactical structure, although this structure is implicit and to some degree hidden in its textual content. In addition, typographical elements such as punctuation marks, capitalization, numerics, and special characters – particularly when coupled with layout artifacts such as white spacing, carriage returns, underlining, asterisks, tables, columns, and so on – can often serve as a kind of "soft markup" language, providing clues to help identify important document subcomponents such as paragraphs, titles, publication dates, author names, table records, headers, and footnotes. Word sequence may also be a structurally meaningful dimension to a document. At the other end of the "unstructured" spectrum, some text documents, like those generated from a WYSIWYG HTML editor, actually possess from their inception more overt types of embedded metadata in the form of formalized markup tags.

Documents that have relatively little in the way of strong typographical, layout, or markup indicators to denote structure – like most scientific research papers, business reports, legal memoranda, and news stories – are sometimes referred to as *free-format* or *weakly structured* documents. On the other hand, documents with extensive and consistent format elements in which field-type metadata can be more easily inferred – such as some e-mail, HTML Web pages, PDF files, and word-processing files with heavy document templating or style-sheet constraints – are occasionally described as *semistructured* documents.

I.1.2 Document Features

The preprocessing operations that support text mining attempt to leverage many different elements contained in a natural language document in order to transform it from an irregular and implicitly structured representation into an explicitly structured representation. However, given the potentially large number of words, phrases, sentences, typographical elements, and layout artifacts that even a short document may have – not to mention the potentially vast number of different senses that each of these elements may have in various contexts and combinations – an essential task for most text mining systems is the identification of a simplified subset of document features that can be used to represent a particular document as a whole. We refer to such a set of features as the *representational model* of a document and say that individual documents are *represented by* the set of features that their representational models contain.

Even with attempts to develop efficient representational models, each document in a collection is usually made up of a large number – sometimes an exceedingly large number – of features. The large number of features required to represent documents in a collection affects almost every aspect of a text mining system's approach, design, and performance.

Problems relating to high *feature dimensionality* (i.e., the size and scale of possible combinations of feature values for data) are typically of much greater magnitude in text mining systems than in classic data mining systems. Structured representations of natural language documents have much larger numbers of potentially representative features – and thus higher numbers of possible combinations of feature values – than one generally finds with records in relational or hierarchical databases.

For even the most modest document collections, the number of word-level features required to represent the documents in these collections can be exceedingly large. For example, in an extremely small collection of 15,000 documents culled from Reuters news feeds, more than 25,000 nontrivial word stems could be identified.

Even when one works with more optimized feature types, tens of thousands of concept-level features may still be relevant for a single application domain. The number of attributes in a relational database that are analyzed in a data mining task is usually significantly smaller.

The high dimensionality of potentially representative features in document collections is a driving factor in the development of text mining preprocessing operations aimed at creating more streamlined representational models. This high dimensionality also indirectly contributes to other conditions that separate text mining systems from data mining systems such as greater levels of pattern overabundance and more acute requirements for postquery refinement techniques.

Another characteristic of natural language documents is what might be described as *feature sparsity*. Only a small percentage of all possible features for a document collection as a whole appears in any single document, and thus when a document is represented as a binary vector of features, nearly all values of the vector are zero.

The tuple dimension is also sparse. That is, some features often appear in only a few documents, which means that the support of many patterns is quite low.

Commonly Used Document Features: Characters, Words, Terms, and Concepts

Because text mining algorithms operate on the feature-based representations of documents and not the underlying documents themselves, there is often a tradeoff between two important goals. The first goal is to achieve the correct calibration of the volume and semantic level of features to portray the meaning of a document accurately, which tends to incline text mining preprocessing operations toward selecting or extracting relatively more features to represent documents. The second goal is to identify features in a way that is most computationally efficient and practical for pattern discovery, which is a process that emphasizes the streamlining of representative feature sets; such streamlining is sometimes supported by the validation, normalization, or cross-referencing of features against controlled vocabularies or external knowledge sources such as dictionaries, thesauri, ontologies, or knowledge bases to assist in generating smaller representative sets of more semantically rich features.

Although many potential features can be employed to represent documents,[1] the following four types are most commonly used:

- ***Characters.*** The individual component-level letters, numerals, special characters and spaces are the building blocks of higher-level semantic features such as words, terms, and concepts. A character-level representation can include the full set of all characters for a document or some filtered subset. Character-based representations without positional information (i.e., bag-of-characters approaches) are often of very limited utility in text mining applications. Character-based representations that include some level of positional information (e.g., bigrams or trigrams) are somewhat more useful and common. In general, however, character-based representations can often be unwieldy for some types of text processing techniques because the feature space for a document is fairly unoptimized. On the other hand, this feature space can in many ways be viewed as the most complete of any representation of a real-world text document.
- ***Words.*** Specific words selected directly from a "native" document are at what might be described as the basic level of semantic richness. For this reason, word-level features are sometimes referred to as existing in the *native feature space* of a document. In general, a single word-level feature should equate with, or have the value of, no more than one linguistic token. Phrases, multiword expressions, or even multiword hyphenates would not constitute single word-level features. It is possible for a word-level representation of a document to include a feature for each word within that document – that is the "full text," where a document is represented by a complete and unabridged set of its word-level features. This can

[1] Beyond the three feature types discussed and defined here – namely, words, terms, and concepts – other features that have been used for representing documents include linguistic phrases, nonconsecutive phrases, keyphrases, character bigrams, character trigrams, frames, and parse trees.

lead to some word-level representations of document collections having tens or even hundreds of thousands of unique words in its feature space. However, most word-level document representations exhibit at least some minimal optimization and therefore consist of subsets of representative features filtered for items such as stop words, symbolic characters, and meaningless numerics.

- ***Terms.*** *Terms* are single words and multiword phrases selected directly from the corpus of a native document by means of *term-extraction* methodologies. Term-level features, in the sense of this definition, can *only* be made up of specific words and expressions found within the native document for which they are meant to be generally representative. Hence, a term-based representation of a document is necessarily composed of a subset of the terms in that document. For example, if a document contained the sentence

  > President Abraham Lincoln experienced a career that took him from log cabin to White House,

  a list of terms to represent the document could include single word forms such as "Lincoln," "took," "career," and "cabin" as well as multiword forms like "President Abraham Lincoln," "log cabin," and "White House."

  Several of term-extraction methodologies can convert the raw text of a native document into a series of *normalized terms* – that is, sequences of one or more tokenized and lemmatized word forms associated with part-of-speech tags. Sometimes an external lexicon is also used to provide a controlled vocabulary for term normalization. Term-extraction methodologies employ various approaches for generating and filtering an abbreviated list of most meaningful *candidate terms* from among a set of normalized terms for the representation of a document. This culling process results in a smaller but relatively more semantically rich document representation than that found in word-level document representations.
- ***Concepts.***[2] *Concepts* are features generated for a document by means of manual, statistical, rule-based, or hybrid *categorization* methodologies. Concept-level features can be manually generated for documents but are now more commonly extracted from documents using complex preprocessing routines that identify single words, multiword expressions, whole clauses, or even larger syntactical units that are then related to specific *concept identifiers*. For instance, a document collection that includes reviews of sports cars may not actually include the specific word "automotive" or the specific phrase "test drives," but the concepts "automotive" and "test drives" might nevertheless be found among the set of concepts used to to identify and represent the collection.

  Many categorization methodologies involve a degree of cross-referencing against an external knowledge source; for some statistical methods, this source might simply be an annotated collection of training documents. For manual and rule-based categorization methods, the cross-referencing and validation of prospective concept-level features typically involve interaction with a "gold standard" such as a preexisting domain ontology, lexicon, or formal concept

[2] Although some computer scientists make distinctions between keywords and concepts (e.g., Blake and Pratt 2001), this book recognizes the two as relatively interchangeable labels for the same feature type and will generally refer to either under the label *concept*.

hierarchy – or even just the mind of a human domain expert. Unlike word- and term-level features, concept-level features can consist of words not specifically found in the native document.

Of the four types of features described here, terms and concepts reflect the features with the most condensed and expressive levels of semantic value, and there are many advantages to their use in representing documents for text mining purposes. With regard to the overall size of their feature sets, term- and concept-based representations exhibit roughly the same efficiency but are generally much more efficient than character- or word-based document models. Term-level representations can sometimes be more easily and automatically generated from the original source text (through various term-extraction techniques) than concept-level representations, which as a practical matter have often entailed some level of human interaction.

Concept-level representations, however, are much better than any other feature-set representation at handling synonymy and polysemy and are clearly best at relating a given feature to its various hyponyms and hypernyms. Concept-based representations can be processed to support very sophisticated concept hierarchies, and arguably provide the best representations for leveraging the domain knowledge afforded by ontologies and knowledge bases.

Still, concept-level representations do have a few potential drawbacks. Possible disadvantages of using concept-level features to represent documents include (a) the relative complexity of applying the heuristics, during preprocessing operations, required to extract and validate concept-type features and (b) the domain-dependence of many concepts.[3]

Concept-level document representations generated by categorization are often stored in vector formats. For instance, both CDM-based methodologies and Los Alamos II–type concept extraction approaches result in individual documents being stored as vectors.

Hybrid approaches to the generation of feature-based document representations can exist. By way of example, a particular text mining system's preprocessing operations could first extract terms using term extraction techniques and then match or normalize these terms, or do both, by winnowing them against a list of meaningful entities and topics (i.e., concepts) extracted through categorization. Such hybrid approaches, however, need careful planning, testing, and optimization to avoid having dramatic – and extremely resource-intensive – growth in the feature dimensionality of individual document representations without proportionately increased levels of system effectiveness.

For the most part, this book concentrates on text mining solutions that rely on documents represented by concept-level features, referring to other feature types where necessary to highlight idiosyncratic characteristics or techniques. Nevertheless, many of the approaches described in this chapter for identifying and browsing patterns within document collections based on concept-level representations can also

[3] It should at least be mentioned that there are some more distinct disadvantages to using manually generated concept-level representations. For instance, manually generated concepts are fixed, labor-intensive to assign, and so on. See Blake and Pratt (2001).