

# Topic Modeling with R

Manika Lamba  
RLadies Urmia (Iran)  
27th April 2022

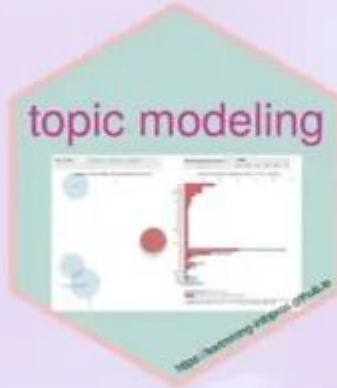


## R-Ladies Urmia

چهارشنبه 7 اردیبهشت ساعت 11:45 ق.ظ



Topic Modeling  
(Text Mining)  
with R





Manika Lamba


Online Free Zoom Event

[www.meetup.com/rladies-Urmia/](https://www.meetup.com/rladies-Urmia/)

لینک ثبت نام:

 @RLadiesUrmia

 rladiesurmia

 @RLadiesIran



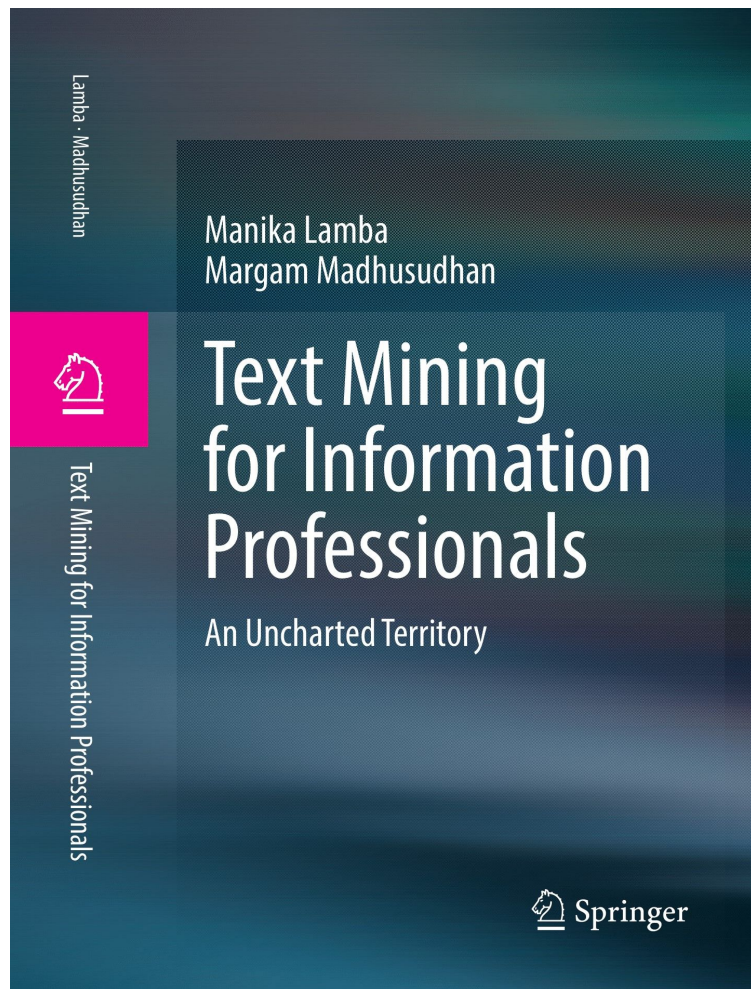
Wednesay, 27th April 2022



<https://manika-lamba.github.io>



@lamba\_manika



## ToC

Chapter 1: The Computational Library

Chapter 2: Text Data and Where to Find Them?

**Chapter 3: Text Pre-Processing**

**Chapter 4: Topic Modeling**

Chapter 5: Network Text Analysis

Chapter 6: Burst Detection

Chapter 7: Sentiment Analysis

Chapter 8: Predictive Modeling

Chapter 9: Information Visualization

Chapter 10: Tools and Techniques for Text Mining and Visualization

Chapter 11: Text Data and Mining Ethics

[Amazon](#)

[Publisher Website](#)

[GitHub](#)

[Author Website](#)

# What is Topic Modeling?

- A topic can be defined as the main idea discussed in a text, i.e., the theme or subject of different granularities
- In contrast, topic modeling acts as a text mining approach to understand, organize, process, extract, manage, and summarize knowledge
- There are no machine-readable annotations that can tell the topic modeling programs about the semantic meaning of the words in the text. Thus, it infers abstract topics based on “similar patterns of word usage in each document”
- Topics are simply groups of words from the collection of documents that represents the information in the collection in the best way

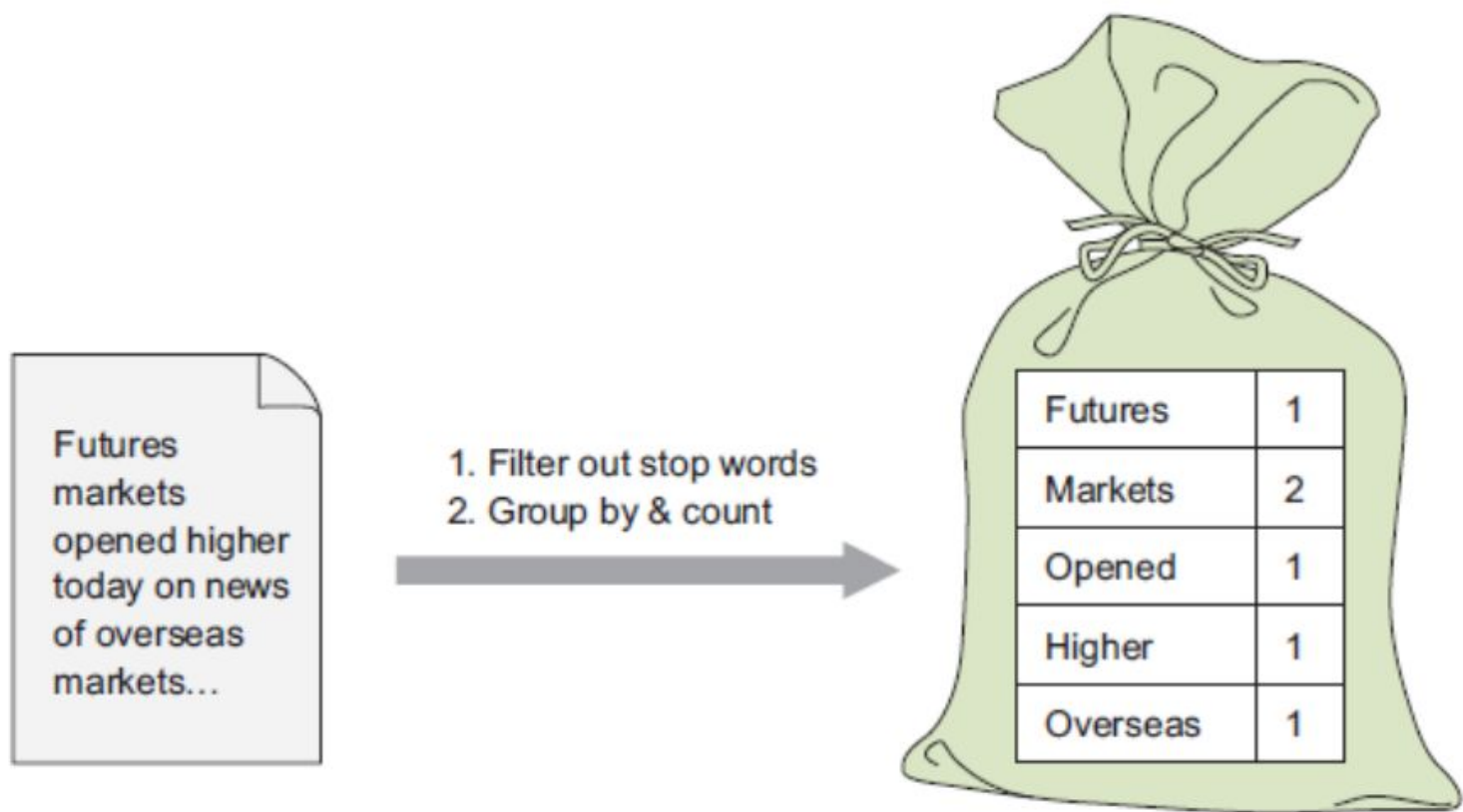
# When to Use Topic Modeling

- When you have a vast collection of text documents
- When the collection belongs to a specific subject
- When the collection has a similar type of documents, such as when all files in the collection are newspaper articles

## When NOT to Use Topic Modeling

- When you have a relatively small number of documents
- When you do not have any idea about your collection. In this case, clustering will be a better option than using topic modeling
- When the collection has a mixture of different types of documents, such as when the collection is composed of newspaper archives, journal articles, and ETDs.

# Bag of Words Assumption



# Term-Document Matrix

Doc	Text
[1]	text analysis is fun
[2]	I like doing text analysis
[3]	I like puppies, they are fun



	[1]	[2]	[3]
text	1	1	0
analysis	1	1	0
fun	1	0	1
like	0	1	1
do	0	1	0
puppy	0	0	1
be	1	0	1



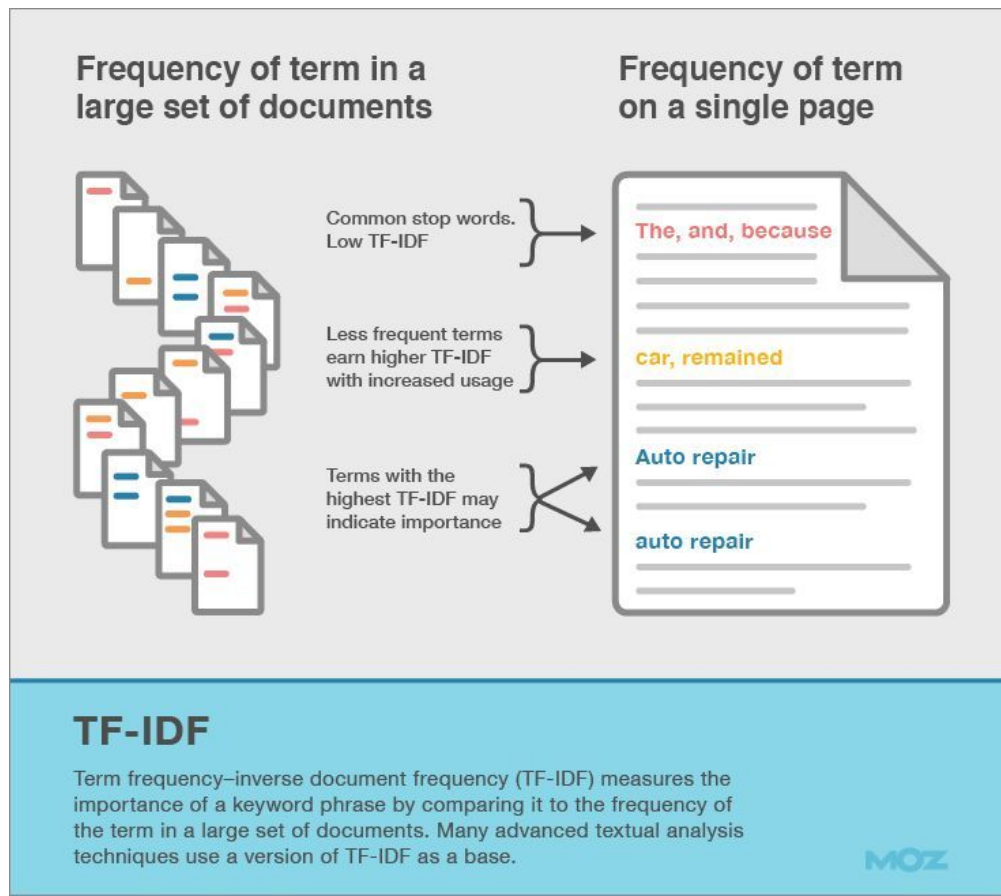
# Document-Term Matrix

Doc	Text
[1]	text analysis is fun
[2]	I like doing text analysis
[3]	I like puppies, they are fun



	text	analysis	fun	like	do	puppy	be
[1]	1	1	1	0	0	0	1
[2]	1	1	0	1	1	0	0
[3]	0	0	1	1	0	1	1

# Term frequency-inverse document frequency (TF-IDF)

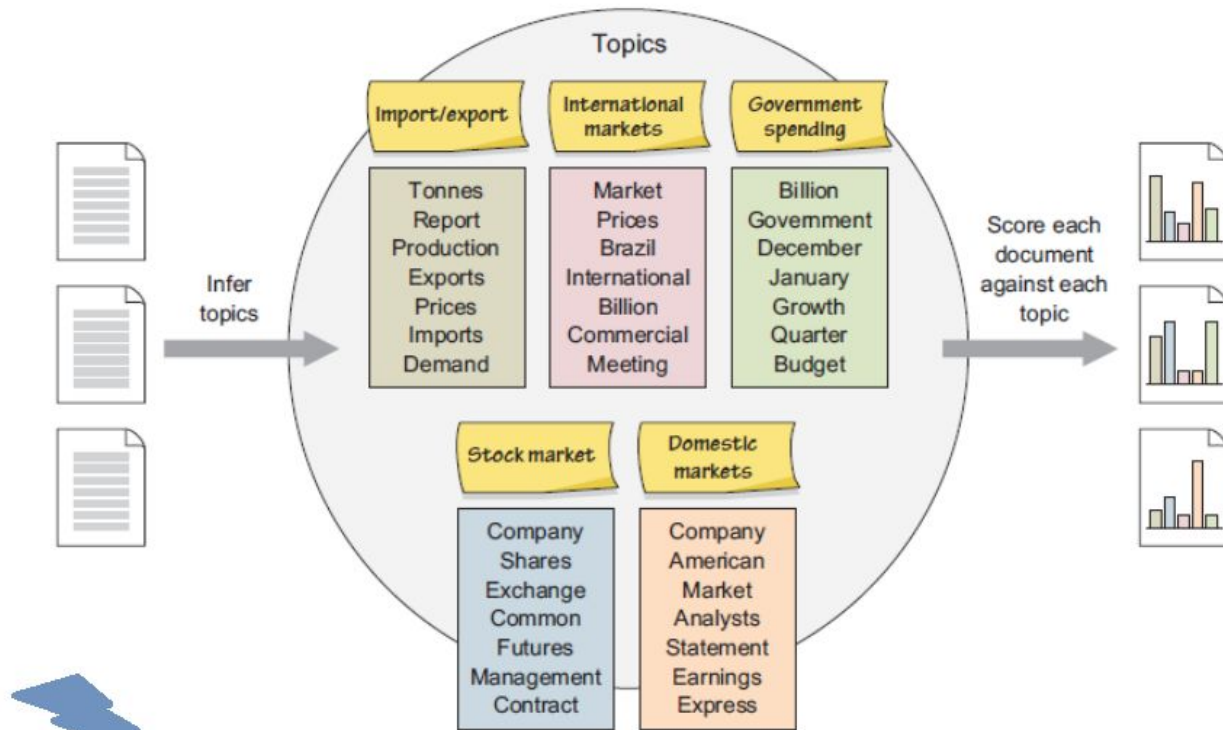


Source:  
<https://www.tmbblast.com/blog/seo/what-and-why-is-tf-idf-important-for-seo/>

# Important Concepts

1. **Term-document matrix (TDM)** represents terms as a table or matrix of numbers for a given corpus. In TDM, terms are represented as rows and documents as columns for a corpus where the number of occurrences of terms in the document is entered in the boxes
2. **Document-term matrix (DTM)** represents terms as a table or matrix of numbers for a given corpus. It is a transposition of TDM; therefore, in DTM, each document is a row, and each word is the column
3. The term **weighting** is popularly used in information retrieval and supervised machine learning tasks like text classification. It assumes that a word that occurs the most is the word that best describes that particular document. Therefore, it makes a list of more discriminative terms than others and assigns a weight to each highly occurring term
4. **Term frequency-inverse document frequency (TF-IDF)** evaluates the relevancy of a term for a document in a corpus and is the most popular weighting scheme in information retrieval

# How Topic Modeling Works?



**Latent Dirichlet Allocation.** The topics are the latent variables and are determined automatically by the algorithm. The names of those topics shown in the thin strips are human-inferred and human-applied; the algorithm has no inherent capability to name the topics. Each document expresses each latent variable (topic) to a varying degree.

Every document consists of a mix of topics



100% Topic A



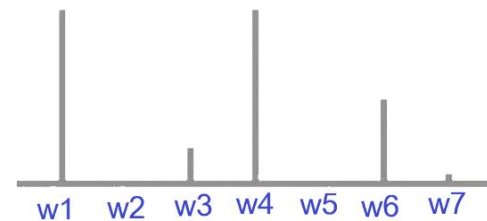
100% Topic B



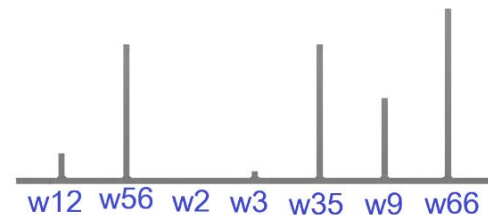
60% Topic A  
40% Topic B

Every topic consists of a mix of words

Topic A



Topic B

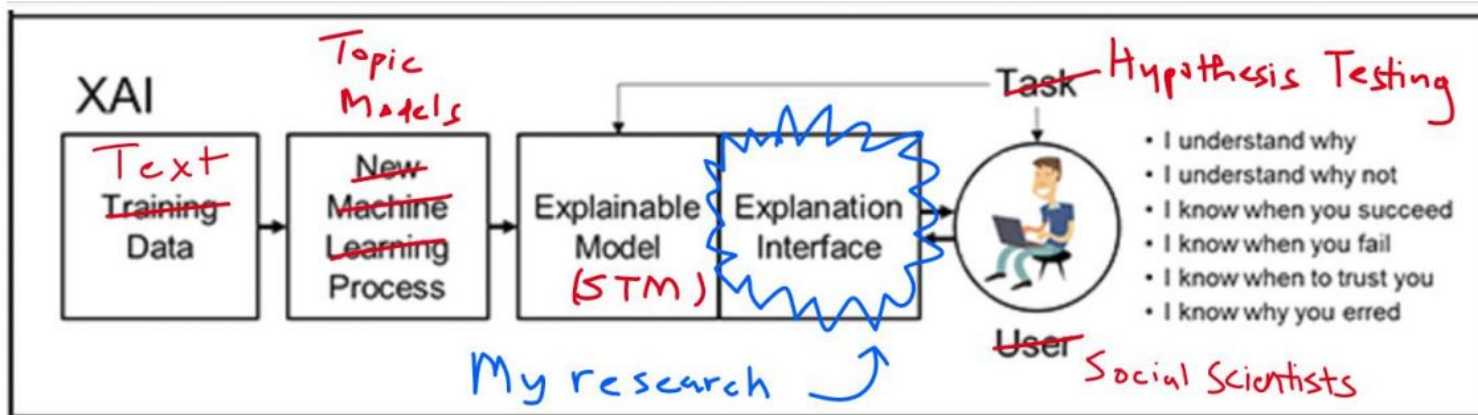
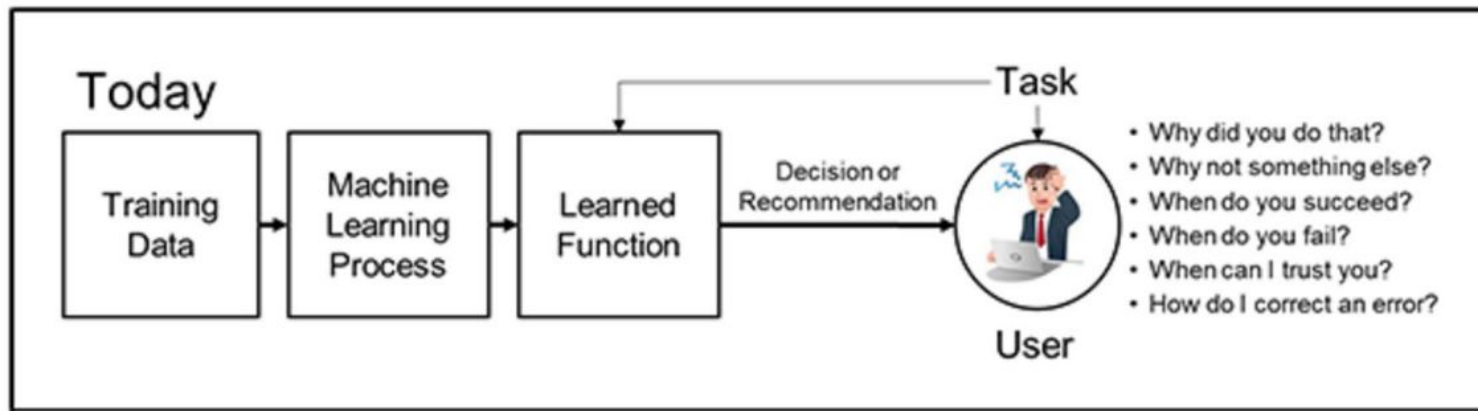


Some of the popular packages in R to perform topic modeling include ***quanteda***, ***stm***, ***tm***, ***lda***, ***topicmodels***, ***text2vec***, ***topicdoc***, ***BTM***, ***tidytext***, and ***textmineR***

# Procedure

1. Obtain dataset (e.g. webscraping, API, etc.)
2. Preparing a corpus
3. Pre-processing (Tokenization, stemming, n-grams)
4. Exploratory analysis (Word clouds, clustering)
5. Determining the number of topics
6. Selecting the appropriate algorithm
7. Iterating the whole process till the algorithm fits the model

# Explainable AI



# Data

Dataset consists of nearly 2,500 research abstracts from six years worth of publications by University of North Carolina at Charlotte (UNCC) researchers in Social Science (across dept/college) and Computing & Informatics (the entire college CCI).



# Research Questions

*What are the topics?*

Part 1: Use LDA (Latent Dirichlet Allocation)

*How are they interrelated?*

Part 2: Use CTM (Correlated Topic Model)

*What is the effect discipline (social sci. vs computing) & year has on the topics?*

Part 3: Use STM (Structural Topic Model)