Text Mining with R:-

Tokens and Tokenization:-

- Tokens are building blocks of text in text analysis.
- They represent individual units of text that have a meaning and can be analyzed.
- Often a token is a word, but it could also be a sentence, paragraph or even a group of words like n-gram.
- n-gram: a sequence of n words, such as "New york" for 2 words or "once upon a time" for 4 words.
- The act of breaking down a large piece of text into these tokens is called tokenization.
- It the act of taking a sentence, paragraph or document and splitting it into these smaller pieces.

Why tokenization is important:-

- Tokenization allows us to examine the content and structure of text data in a manageable and analyzable format.

For instance, we can:

- Count the occurances of each word in a text to identify frequently used terms.
- Filter out common but less meaningful words (like "the" or "and") to focus on keywords.
- Analyze patterns in text, like word frequency or sequence of words.

Corpus:- (plural - corpora)

A corpus is a structured collection of text documents. Its often created for text analysis or natural language processing tasks.

- Each document in a corpus is typically annotated with metadata, which provides additional info like author, publication date, language or genre.

Document-Term Matrix (DTM)

A document-Term matrix is a type of matrix representation for a corpus, where

- Rows represent documents in corpus.
- Columns represent terms found in corpus.
- Values in the matrix are frequencies or important measures, such as term frequency (count of a term in a document) or tf-idf (term frequency - inverse document frequency, which reflects importance).

Stopwords :-

Stopwords are common words, like "the", "of", "and", "is", which frequently appear but dont add much meaningful information to analysis.

Sentiment Analysis :- (Opinion Mining)

- A method to evaluate the emotional tone of a text by identifying and measuring positive, negative or nuanced emotions (joy, anger etc).

Sentiment lexicons :-

- Sentiment lexicons are predefined lists of words that are associated with specific emotional or opinion-based meanings.

(1) **AFINN lexicon** :- Score words on a scale from -5 (very negative) to +5 (very positive)

(2) Bing lexicon: Classifies words into binary categories of "positive" or "negative".

(3). NRC lexicon:- Categorizes words into various emotion (like joy, anger, trust) as well as binary "positive" and "negative" categories.

Sentiment Scores and calculations

- Unigram:- A single word, most lexicons use unigram-based sentiment scores.
- Sentiment Score:- The numeric or categorical value assigned to each word based on its emotional content (eg. -5 to +5 in affin).
- Net sentiment:- The overall sentiment for a section of text, often calculated as the difference b/w positive and negative sentiment score.

Chunking text for Sentiment Analysis:-

Chunking:- Dividing text into manageable segments (eg. by paragraph or sentences) for sentiment scoring, as overall sentiment can vary across the text.

Integer division: A technique used to define sections in the text by dividing line numbers to create groups [illegible].

One-Row-per-token-format:-

The structure for tidy text data where each token (eg. word) occupies its own row in a table, which makes the data compatible with tidy tools.

Chapter 3:- Analyzing word and document frequency.

(1) Term frequency:-

- Term frequency measures how often a word or "term", appears within a single document.
- Its calculated as the count of a word in a document divided by total number of words in that document.
- TF is useful for identifying the most frequent terms within a document but does not distinguish b/w common, less meaning-ful words and content specific words.

(2) Inverse Document frequency:-

- Inverse document frequency (IDF) adjusts for words that are common across a collection of documents, giving a higher weight to

terms that are less frequent across the entire collection.

- IDF helps downweight common words and upweight unique words that might be more meaningful in certain documents.

- $idf(term) = \ln\left(\frac{\text{no. of documents}}{\text{no. of documents containing term}}\right)$

- **3. TF-IDF :**

- TF-IDF is a metric that combines Term frequency and Inverse document frequency to highlight words that are important within individual documents but not too common across the document collection.

- It calculated as

$$tf\text{-}idf(term, document) = tf(term, document) * idf(term)$$

4. Zipf's Law:-

- Zipf's Law describes a pattern where frequency of a word is inversely proportional to its rank, in terms of frequency.
- This law results in a "long-tailed" distribution where a few words are very common, while many words are used infrequently.
- In a graph of log-transformed term frequencies against ranks, Zipf's Law is observed as a line with negative slope, often around -1.

Chapter 4:- Relationship between words:- n-grams and correlations:-

(i) N-grams:-

- N-grams are sequences of n consecutive words in a text, often used to capture the context in which words occur:-
  - Bigram - An n-gram of two words. (eg. "happy day").
  - Trigram - An n-gram of three words. (eg. "beautiful sunny day").
- N-grams allow us to understand common word pairs or sequences in a document, providing more context than analyzing individual words alone.

Important functions:-

(1) unnest_tokens() :-

A function from tidytext that converts a text coloum into tokens, creating a row for each token.

(2) antijoin() :-

A function in dplyr used to filter out stop words from text data by joining it with a list of stopwords and keeping only the non-matching terms.

(3). Wordcloud : (wordcloud()) :-

- Visual display of word frequencies where the size of each word represents its frequency, often used to show prominent words by sentiment.

(previous chapter functions ↑, from now functions used in chapter 4.

(4) Seperate() and unite() from tidyr:-

- Seperate() - splits a single coloumn into multiple coloumns based on a delimiter.
- unite() - combines multiple coloumns into a single coloumn.

(5) ggraph package:- (ggraph())

- ggraph is an extension of ggplot2 designed for creating network plots, allowing us to visualize relationships b/w words (eg. bigrams) as graphs.
- Network visualizations can show how words connect, helping us see clusters, common phrases or key phrases that structure a text.

- geom_edge_link(), geom_node_point(), geom_node_text(), ggraph().

6. igraph package :- (graph-from-data-frame())

- igraph is a package used to create and manipulate graph structures
- It helps prepare network data before visualizing it with ggraph.
- Graphs in igraph are connected with nodes (individual words) and edges (connections between words) with optional weights (such as bigram frequency).

7. pairwise_count() from widyr :-

- pairwise_count() counts the no. of times each pair of items (eg. words) appears together within the same unit of text (eg. a document or section). This is useful for exploring co-occurances of words that appear within the same text sections, but not necessarily as adjacent words.

8. Pairwise_cor() from widyr:-

- pairwise_cor() calculates the phi coefficient or binary correlation b/w two items (eg. words) to measure their likelihood of appearing together relative to how often they appear seperately, This is hdpful for identifying strong for identifying strong relationship/association within text, even if they are adjacent.

- phi-coefficient:- This is a measure of association for binary data.
  - It quantifies how much more likely two words are to cooccur, offering insights into significant word pairings within a corpus.

- phi coefficient is equivalent pearson correlation.

2. tdf-Idf for bigrams

- tdf-Idf can be applied to n-grams to identify contextually significant word pairs or phrases in document, similar to how it highlights important single words. This helps identify word pairs unique to specific documents or sections within a text collection.

3. Negation handling in sentiment analysis :-

- It is an approach to ~~handle~~ adjust sentiment analysis by identifying words that change the sentiment of following terms. (like "not", "never")
- By using bigrams we can capture phrases such as "not happy" to reverse or reverse or adjust the sentiment score, providing a more accurate sentiment analysis.

4. Markov chains:-

- A markov chain is a model where the occurance of each word depends only on the preceding word, making it relevant for bigram analysis.
- Network graphs built from bigrams can be seen as visualizations of a Markov chain, providing insights into sequences and transitions b/w words.

5. Conceptual Co-Occurance Analysis:-

- Conceptual co-occurance analysis uses pair of words that appear within the same section, chapter or document.
- This can reveal thematic patterns or topics within a text by identifying words that frequently appear together.

Chapter 5. Converting to and from non-tidy formats:-

1. Document-Term Matrix & Sparse Matrix:-

- DTMs are usually sparse matrixes (most values are zeroes) because not all terms appear in all documents.

- Sparse matrix is a matrix with many zero values, used to represent DTM's efficiently. Rather than storing all values, sparse matrices save space by storing only non-zero values.

2. tidy()

- tidy() function from broom package converts non-tidy objects like DTM into a tidy dataframe format.

- This allows users to use the tidy data

structure to conduct analysis, such as sentiment analysis or tf-Idf on datasets initially stored as DTM's.

3. cast_* Functions :-

- cast_dtm() : converts tidy data into a DTM object from tm package
- cast_dfm() :- converts tidy data into a dfm (document feature matrix) from the quanteda package.
- cast_sparse() :- converts tidy data into a sparse matrix, which is useful for ML libraries that require sparse matrices as input.
- cast() turns a tidy one-term-per-row data frame into a matrix.

4. Quanteda package :-

- quanteda is a text mining package in R that focuses on quatitative analysis of textual data.
- It provides an alternative format for DTM known as dfm's.

5. tm package :-

- tm (text mining) is another package in R for NLP that works with DTM and corpus objects.
- Its widely used for text processing, including tokenization and creating DTM's.

~~6. tokenizers package :-~~

6. Loughran and McDonald dictionary:-

- This dictionary is a sentiment lexicon developed for analyzing financial text.
- It categorizes words into sentiments like positive, negative, litigious, uncertain, constraining and superflous.
- It is designed to prevent false positive or negative readings in finance contexts where words like "growth" or "stakes" might have neutral meanings.

Chapter 6:- Topic Modelling

1. Topic Modelling:-

- This is an unsupervised learning technique used to discover the underlying themes or "topics" within a collection of documents.
- It automatically groups related words and documents into topics based on patterns in text data.
- Its useful for organizing large sets of documents, like news article, books or tweets, where each topic may contain multiple overlapping topics.

2. Latent Dirichlet Allocation (LDA):-

LDA assumes:-

- Each document is a mixture of several topic's and.
- Each topic is a mixture of specific words.

Understanding topics and words:-

- In LDA, topics are generally probability distribution over words.
- This means each topic is represented by a set of words, where each word has the probability of belonging to that topic.
- Similarly, documents are probability distributions over topics.
- Each topic is thus made up of certain proportion of words from each topic.

Generative process:-

- LDA is a generative model, which means it assumes that the documents were generated based on probabilistic process. LDA essentially tries to reverse this process to discover the topics:-

1. Choose the distribution of topics for each document.
2. Choose a distribution of words for each topic.
3. Generate words for each document.

parameters in LDA :-

LDA relies on 3 parameters :-

1. No. of topics (k) :- This is usually often chosen in advance based on the no. of distinct topics you expect to find.

- For instance, if analyzing new articles, you might choose k=10 if you think there are 10 distinct subjects like politics, business and sports. etc.

2. Alpha (α) :- Controls the per document topic distribution.

- A higher alpha means documents will be

composed of more topics in similar proportions, while lower alpha means documents will lean more heavily on few specific topics.

Beta (B) :- Controls the per-topic word distribution

- A higher beta means topics will include a broader range of words, while lower beta means topics will be more concentrated with a smaller number of frequent words.

---

augment ( ) :

- The augment() function assigns model results to each observation in the original data.
- In topic modelling, augment() assigns each word in each document to a topic creating a dataset that shows which topics are most associated with specific words within document.

(words within)

Twitter data project - Covid-19 tweets:-

1. Data preprocessing:-

*. Text cleaning:-

- Remove Url's
- Remove user mention (@username)
- Remove Hashtags (#).
- Remove punctuation, special characters, and Numbers.
- Convert to lowercase.
- Remove emoji's.

* Tokenization:-

- Breakdown each tweet into individual words (tokens) for easier processing

* Stopword Removal

* Stemming and Lemmatization

* Handling Negation.

2. Feature Engineering:-

* Bag-of-words (BoW)
* TF-IDF
* Word embeddings.
  - use pretrained embedding's like Word2Vec, GloVe or FastText
  - BERT embeddings.
* Sentiment specific features:-
  - Emoji Counts.
  - Exclamation marks Count.
  - Sentiment lexicon scores.

3. EDA:-

* Sentiment distribution.
* Word cloud for +ve and -ve tweets.
* Top hashtags.
* Top n-grams.