Introduction to Data Science Topic-8

- Instructor: Professor Henry Horng-Shing Lu,

  Institute of Statistics, National Yang Ming Chiao Tung University, Taiwan

  Email: henryhslu@nycu.edu.tw

- WWW: http://misg.stat.nctu.edu.tw/hslu/course/DataScience.htm
- Classroom: ED B27 (新竹市大學路1001號工程四館B27教室)
- References:

  M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

  K.-T. Tsai, Machine Learning for Knowledge Discovery with R: Methodologies for Modeling, Inference, and Prediction, 2021, Chapman and Hall/CRC.
- Evaluation: Homework: 70%, Term Project: 30%
- Office hours: By appointment

Course Outline

10 Topics and 10 Homeworks:

- **Introduction of Data Science**
- **Introduction of R and Python**
- **Cleaning Data into R and Python**
- **Data Visualization**
- **Exploratory Data Analysis**
- **Regression (Supervised Learning)**
- **Classification (Supervised Learning)**
- **Text Mining**
- **Clustering (Unsupervised Learning)**
- **Neural Network and Deep Learning**

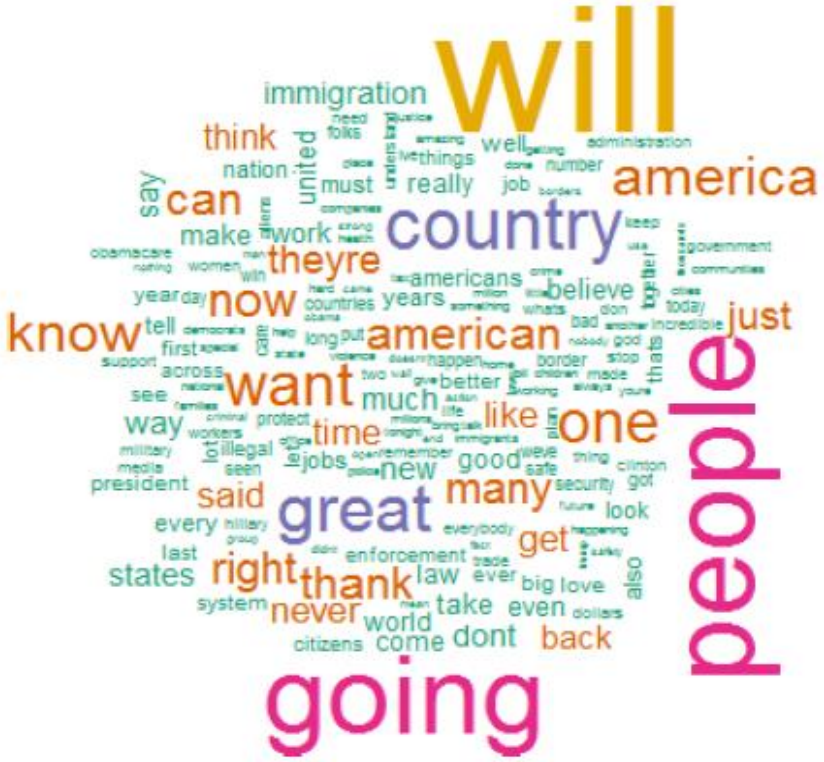
Text Mining with R

References: Ch. 8, M. A. Pathak, Beginning Data Science with R, 2014, Springer-Verlag.

https://github.com/howard-haowen/NLP-demos/tree/main/NSYSU

Text mining?

Well, the election, it came out really well. Next time we' ll triple the number or quadruple it. We want to get it over 51, right? At least 51.Well this is Black History Month, so this is our little breakfast, our little get-together. Hi Lynn, how are you? Just a few notes. During this month, we honor the tremendous history of African-Americans throughout our country. Throughout the world, if you really think about it, right? And their story is one of unimaginable sacrifice, hard work, and faith in America. I' ve gotten a real glimpse—during the campaign, I' d go around with Ben to a lot of different places I wasn' t so familiar with. They' re incredible people. And I want to thank Ben Carson, who' s gonna be heading up HUD. That' s a big job. That' s a job that' s not only housing, but it' s mind and spirit. Right, Ben? And you understand, nobody' s gonna be better than Ben.Last month, we celebrated the life of Reverend Martin Luther King, Jr., whose incredible example is unique in American history. You read all about Dr. Martin Luther King a week ago when somebody said I took the statue out of my office. It turned out that that was fake news. Fake news. The statue is cherished, it' s one of the favorite things in the—and we have some good ones. We have Lincoln, and we have Jefferson, and we have Dr. Martin Luther King. But they said the statue, the bust of Martin Luther King, was taken out of the office. And it was never even touched. So I think it was a disgrace, but that' s the way the press is. Very unfortunate.I am very proud now that we have a museum on the National Mall where people can learn about Reverend King, so many other things. Frederick Douglass is an example of somebody who' s done an amazing job and is being recognized more and more, I noticed. Harriet Tubman, Rosa Parks, and millions more black Americans who made America what it is today. Big impact.I' m proud to honor this heritage and will be honoring it more and more. The folks at the table in almost all cases have been great friends and supporters. Darrell—I met Darrell when he was defending me on television. And the people that were on the other side of the argument didn' t have a chance, right? And Paris has done an amazing job in a very hostile CNN community. He' s all by himself. You' ll have seven people, and Paris. And I' ll take Paris over the seven. But I don' t watch CNN, so I don' t get to see you as much as I used to. I don' t like watching fake news. But Fox has treated me very nice. Wherever Fox is, thank you.We' re gonna need better schools and we need them soon. We need more jobs, we need better wages, a lot better wages. We' re gonna work very hard on the inner city. Ben is gonna be doing that, big league. That' s one of the big things that you' re gonna be looking at. We need safer communities and we' re going to do that with law enforcement. We' re gonna make it safe. We' re gonna make it much better than it is right now. Right now it' s terrible, and I saw you talking about it the other night, Paris, on something else that was really—you did a fantastic job the other night on a very unrelated show.I' m ready to do my part, and I will say this: We' re gonna work together. This is a great group, this is a group that' s been so special to me. You really helped me a lot. If you remember I wasn' t going to do well with the African-American community, and after they heard me speaking and talking about the inner city and lots of other things, we ended up getting—and I won' t go into details—but we ended up getting substantially more than other candidates who had run in the past years. And now we' re gonna take that to new levels. I want to thank my television star over here— Omarosa' s actually a very nice person, nobody knows that. I don' t want to destroy her reputation but she' s a very good person, and she' s been helpful right from the beginning of the campaign, and I appreciate it. I really do. Very special.So I want to thank everybody for being here.



"Text mining is an umbrella term for different types of analysis that we perform over text"

What is Text mining?

- Text mining, text data mining (TDM), or text analytics is the process of deriving high-quality information from text.
- It involves "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. Written resources may include websites, books, emails, reviews, and articles.
- High-quality information is typically obtained by devising patterns and trends by means such as statistical pattern learning.
- According to Hotho et al., we can distinguish between three different perspectives of text mining: information extraction, data mining, and a knowledge discovery in databases (KDD) process.
- Text mining usually involves the process of structuring the input text, deriving patterns within the structured data, and finally evaluation and interpretation of the output.

https://en.wikipedia.org/wiki/Text_mining

What is Text mining?

In short, Text mining is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights. Text is one of the most common data types within databases. Depending on the database, this data can be organized as:

1. **Structured data:** This data is standardized into a tabular format with numerous rows and columns, making it easier to store and process for analysis and machine learning algorithms.
2. **Unstructured data:** This data does not have a predefined data format.

3. **Semi-structured data:** As the name suggests, this data is a blend between structured and unstructured data formats. While it has some organization, it doesn't have enough structure to meet the requirements of a relational database.

KS

what is Text mining?

Text mining, also known as text data mining or text analytics, is the process of extracting meaningful information and knowledge from large volumes of unstructured text. Unstructured text data refers to information that is not organized in a predefined manner, such as emails, social media posts, articles, books, and more. Text mining involves the application of various techniques from natural language processing, machine learning, and computational linguistics to analyze and interpret textual data.

Key components of text mining include:

1. **Text Preprocessing**: This involves cleaning and preparing the raw text data for analysis. Steps may include removing stop words (common words that do not carry much meaning), stemming (reducing words to their root form), and handling other noise or irrelevant information.
2. **Tokenization**: Breaking down the text into smaller units, often words or phrases, referred to as tokens. This process is a fundamental step for further analysis.
3. **Entity Recognition**: Identifying and classifying entities such as names, locations, dates, and other specific information within the text.

4. **Sentiment Analysis**: Determining the sentiment expressed in the text, whether it is positive, negative, or neutral. This is commonly used in social media monitoring, customer reviews, and other applications to gauge public opinion.
5. **Topic Modeling**: Identifying topics or themes present in a collection of documents. Techniques like Latent Dirichlet Allocation (LDA) are often used for this purpose.
6. **Text Classification**: Categorizing documents into predefined categories or classes based on their content. This is frequently used for tasks like spam detection, news categorization, and sentiment analysis.
7. **Information Extraction**: Extracting structured information from unstructured text, such as identifying relationships between entities or events.
8. **Clustering**: Grouping similar documents or pieces of text together based on their content, without predefined categories. This can reveal underlying patterns and structures in the data.

Text mining has numerous practical applications across various domains, including business, healthcare, finance, social media analysis, and scientific research. It allows organizations to derive valuable insights, automate information retrieval, and make informed decisions based on the analysis of large volumes of textual data.

Example for each type of data

1. **Structured Data:** names, addresses, and phone numbers.

2. **Unstructured Data:** social media or product reviews, or rich media formats like, video and audio files.

3. **Semi-structured Data:** semi-structured data include XML, JSON and HTML files.

Why is Text mining?

- Text is all around us, Twitter, books, or reviews.
- Text data grows dramatically, and huge amounts of data are generated each passing day.
- We can analyze data to conduct sentiment analysis, topic modeling, understanding, identifying, finding, classifying, and extracting information.

https://www.hubscience.com/post/what-is-text-mining-and-why-is-it-important

KS

Why do we need Text Mining?

Text mining is valuable for several reasons, and its applications span various industries. Here are some key reasons why text mining is essential:

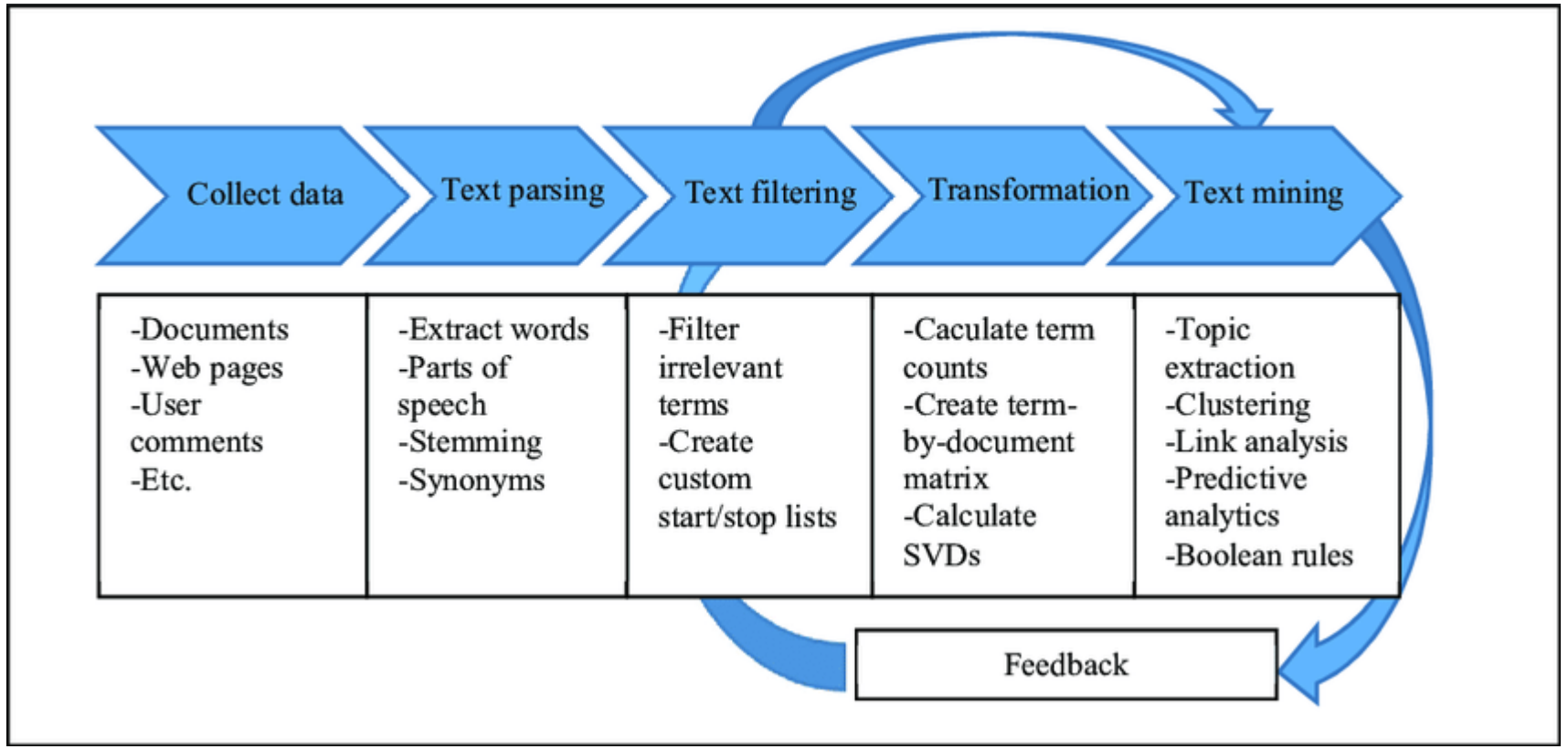
1. **Information Extraction and Knowledge Discovery**: Text mining helps uncover hidden patterns, trends, and valuable knowledge within large volumes of unstructured text. It allows organizations to extract meaningful insights from sources such as articles, documents, social media, and more.
2. **Automated Information Retrieval**: With the ever-increasing amount of textual data available, manual extraction of information is impractical. Text mining automates the process of sorting through vast datasets, making it possible to retrieve relevant information efficiently.
3. **Decision-Making Support**: Text mining provides decision-makers with valuable information for making informed decisions. By analyzing textual data, organizations can gain insights into customer feedback, market trends, and competitive intelligence, aiding strategic decision-making.
4. **Customer Feedback and Sentiment Analysis**: Businesses can use text mining to analyze customer reviews, comments, and feedback on products and services. Sentiment analysis helps gauge public opinion, identify customer satisfaction or dissatisfaction, and make improvements accordingly.

Regenerate

5. **Content Categorization and Organization**: Text mining enables the categorization and organization of large amounts of unstructured text into meaningful groups or topics. This is particularly useful for news articles, academic papers, and other content where automated categorization can save time and improve accessibility.
6. **Search and Information Retrieval**: Text mining techniques enhance search capabilities by improving the relevance and accuracy of search results. This is crucial for search engines, document management systems, and information retrieval applications.
7. **Healthcare and Biomedical Research**: In the healthcare industry, text mining is used to extract information from medical records, research articles, and clinical notes. It aids in identifying patterns, predicting disease outcomes, and advancing biomedical research.
8. **Fraud Detection and Security**: In finance and cybersecurity, text mining is employed to analyze patterns in text data that may indicate fraudulent activities or security threats. This helps in early detection and prevention of potential risks.
9. **Human Resources and Recruitment**: Text mining assists in analyzing resumes, job descriptions, and applicant profiles to match candidates with job requirements. It streamlines the recruitment process by automating the screening of large volumes of text data.
10. **Scientific Research**: In fields such as genomics, proteomics, and materials science, text mining is used to analyze research papers and extract valuable information. This accelerates the pace of scientific discovery and facilitates collaboration among researchers.

Overall, text mining is a powerful tool for converting unstructured textual data into actionable insights, improving decision-making processes, and enhancing the efficiency of various tasks across different domains. Its applications continue to expand as technology advances and the volume of digital text data grows.

Workflow of Text Mining



https://www.researchgate.net/figure/Text-mining-process-Source-Chakraborty-Pagolu-and-Garla-2013_fig1_262413948

KS

Workflow of Text Mining

The workflow of text mining typically involves several stages, each designed to process and extract valuable information from unstructured text data. Here is a generalized workflow for text mining:

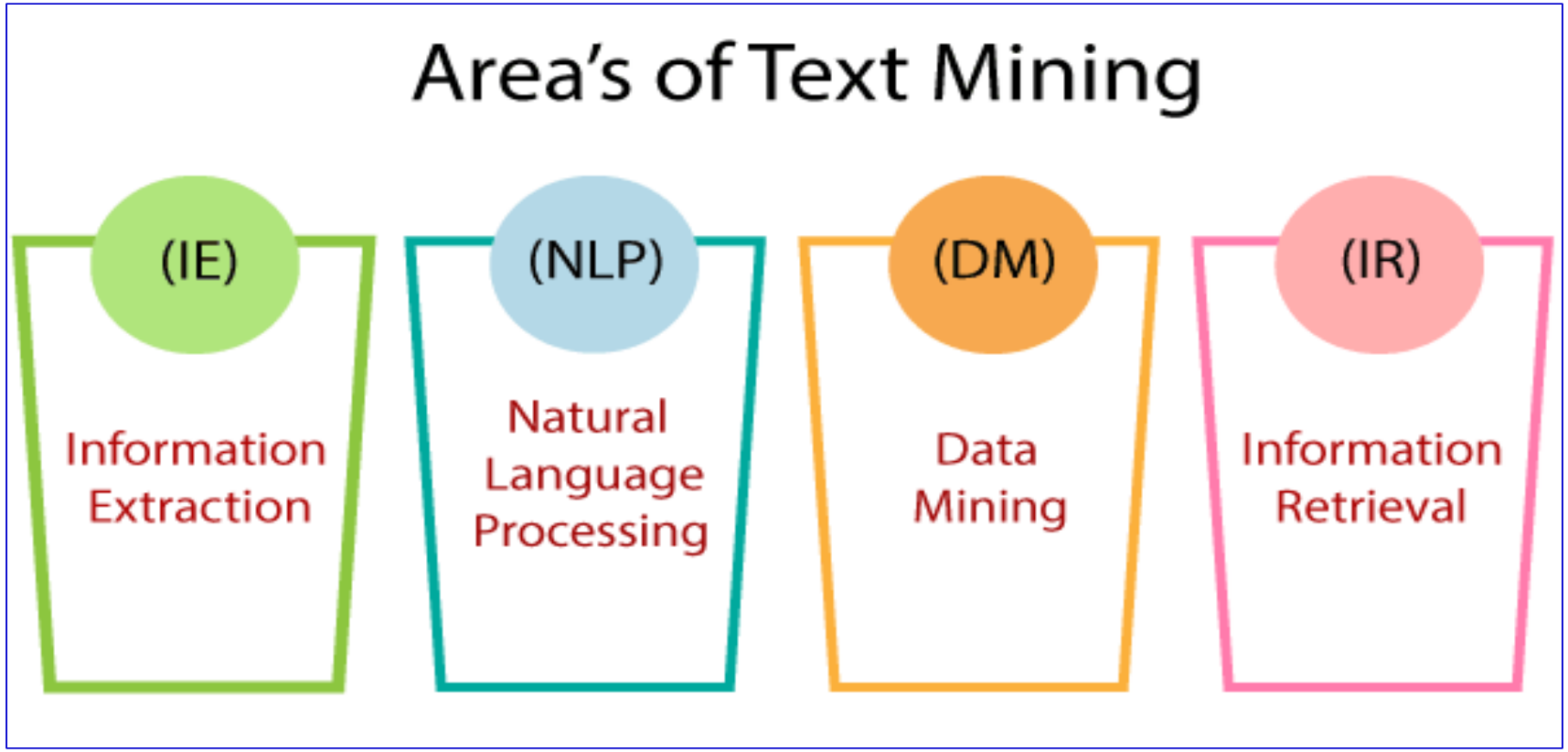
1. **Define Objectives and Scope**:
   - Clearly define the goals and objectives of the text mining project.
   - Determine the scope of the analysis, including the types of documents or texts to be analyzed.
2. **Data Collection**:
   - Gather the relevant text data from various sources, such as documents, websites, social media, emails, or other textual content.
   - Ensure the data is representative of the problem domain and aligned with the project objectives.
3. **Text Preprocessing**:
   - Clean the raw text data to remove noise and irrelevant information.
   - Tokenize the text into words or phrases.
   - Remove stop words (common words with little meaning), perform stemming, and handle other text preprocessing tasks.
4. **Exploratory Data Analysis (EDA)**:
   - Conduct exploratory data analysis to understand the characteristics of the text data.
   - Explore word frequencies, document lengths, and other relevant statistics.

5. **Text Representation**:
   - Convert the text data into a format suitable for analysis. This often involves creating a numerical representation of the text, such as a bag-of-words model or word embeddings.
6. **Feature Extraction**:
   - Extract relevant features from the text data, which could include n-grams, term frequency-inverse document frequency (TF-IDF) scores, or other features based on the analysis goals.
7. **Model Development**:
   - Choose appropriate text mining techniques and algorithms based on the objectives.
   - For tasks like sentiment analysis, topic modeling, or classification, select and train machine learning models or use deep learning techniques.
8. **Model Evaluation**:
   - Evaluate the performance of the text mining model using appropriate metrics.
   - Fine-tune the model parameters if necessary to improve performance.
9. **Results Interpretation**:
   - Interpret the results of the text mining analysis in the context of the project goals.
   - Extract meaningful insights and knowledge from the model's predictions or classifications.

10. **Visualization and Reporting**:
    - Create visualizations to represent the findings, such as word clouds, charts, or graphs.
    - Prepare a comprehensive report or presentation summarizing the key results and insights.
11. **Iterative Refinement**:
    - Based on feedback and additional insights, iterate on the text mining process to refine the models, features, or preprocessing steps.
12. **Deployment (Optional)**:
    - If the text mining model is part of a larger system or application, deploy the model for real-time or batch processing.

Throughout the workflow, it's important to iterate and refine the process based on the feedback received and the insights gained from the analysis. The specific steps and tools used may vary depending on the nature of the text mining task and the available resources.

Text mining techniques



https://www.javatpoint.com/text-data-mining

KS

what are the Text mining techniques?

Text mining techniques encompass a variety of methods and approaches to extract meaningful information from unstructured text data. Here are some key text mining techniques:

1. **Tokenization**:
   - **Description**: Breaking down the text into individual units, such as words or phrases (tokens).
   - **Purpose**: Facilitates further analysis by providing a basis for counting and processing individual elements.
2. **Stemming and Lemmatization**:
   - **Description**: Reducing words to their root form (stemming) or base dictionary form (lemmatization).
   - **Purpose**: Helps in normalizing variations of words, reducing inflected words to a common base.
3. **Stopword Removal**:
   - **Description**: Removing common words (stop words) that do not carry significant meaning, such as "the," "and," or "is."
   - **Purpose**: Reduces noise in the data and focuses on more meaningful terms.
4. **Text Cleaning**:
   - **Description**: Removing irrelevant characters, symbols, or formatting from the text.
   - **Purpose**: Improves the quality of the text data by eliminating unnecessary elements.

5. **Bag-of-Words (BoW) Model**:
    - **Description**: Representing a document as an unordered collection of words, disregarding grammar and word order.
    - **Purpose**: Enables the quantitative analysis of text by creating a matrix of word frequencies.
6. **Term Frequency-Inverse Document Frequency (TF-IDF)**:
    - **Description**: Assigning weights to words based on their frequency in a document and across a collection of documents.
    - **Purpose**: Highlights important terms that are distinctive to a particular document.
7. **N-grams**:
    - **Description**: Extracting contiguous sequences of n items (words or characters) from a given text.
    - **Purpose**: Captures local patterns and relationships within the text.
8. **Named Entity Recognition (NER)**:
    - **Description**: Identifying and classifying entities such as names, locations, organizations, and dates in text.
    - **Purpose**: Extracts structured information from unstructured text.
9. **Sentiment Analysis**:
    - **Description**: Determining the sentiment expressed in a piece of text, whether it is positive, negative, or neutral.
    - **Purpose**: Gauges public opinion, customer sentiment, or emotional tone.

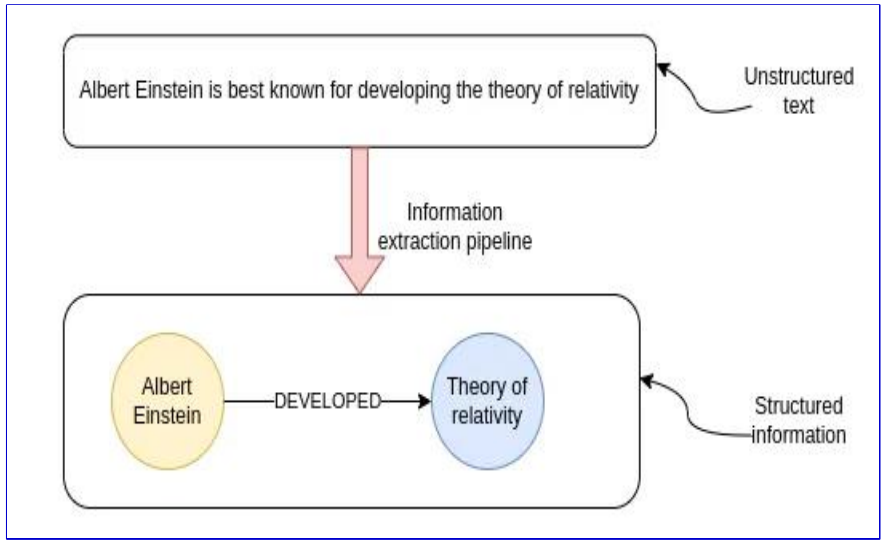
10. **Topic Modeling**:
    - **Description**: Identifying topics present in a collection of documents.
    - **Purpose**: Reveals the main themes or subjects within a set of documents.
11. **Clustering**:
    - **Description**: Grouping similar documents together based on their content.
    - **Purpose**: Identifies patterns and relationships within the text data.
12. **Machine Learning Models**:
    - **Description**: Using supervised or unsupervised machine learning algorithms for tasks like text classification, regression, or clustering.
    - **Purpose**: Automates the analysis of textual data based on labeled or unlabeled examples.
13. **Deep Learning Models**:
    - **Description**: Leveraging neural networks with multiple layers for tasks such as sequence-to-sequence learning, sentiment analysis, and language modeling.
    - **Purpose**: Handles complex relationships and dependencies within text data.

These techniques can be applied individually or in combination, depending on the specific goals of a text mining project. The choice of technique often depends on the nature of the text data and the insights sought from the analysis.

Information Extraction

Information extraction (IE) surfaces the relevant pieces of data when searching various documents. It also focuses on extracting structured information from free text and storing these entities, attributes, and related information in a database.

In short, we can say that the goal of information extraction pipeline is to extract structured information from unstructured text.



https://towardsdatascience.com/extract-knowledge-from-text-end-to-end-information-extraction-pipeline-with-spacy-and-neo4j-502b2b1e0754

Information Extraction

- **Feature selection (attribute selection):** The process of selecting the important features (dimensions) to contribute the most to the output of a predictive analytics model.
- **Feature extraction:** the process of selecting a subset of features to improve the accuracy of a classification task. This is particularly important for dimensionality reduction.
- **Named-entity recognition (NER):** Also known as entity identification or entity extraction, aims to find and categorize specific entities in text, such as names or locations. For example, NER identifies “California” as a location and “Mary” as a woman’s name.

Figure 1: An example of NER application on an example text

An Example of NER Application on an example text.

[Natural Language Processing] #3 Name Entity Recognition Theory Design | by Roye Qiu | Roye’s ResearchCraft | Medium

KS

Information Extraction In Text Mining

Information extraction (IE) is a crucial aspect of text mining that involves automatically extracting structured information from unstructured text. The goal is to identify and capture specific pieces of information, such as entities, relationships, and events, from a large body of text. Here are key components of information extraction in text mining:

1. **Named Entity Recognition (NER)**:
   - **Description**: Identifying and classifying entities, such as names of people, organizations, locations, dates, and more, within the text.
   - **Purpose**: Extracts structured information to understand the "who," "what," "where," and "when" aspects of the text.
2. **Relation Extraction**:
   - **Description**: Identifying and extracting relationships between entities mentioned in the text.
   - **Purpose**: Reveals connections and associations between different entities, providing context and understanding.
3. **Event Extraction**:
   - **Description**: Identifying events or occurrences described in the text, including the involved entities and the relationships between them.
   - **Purpose**: Captures information about actions, processes, or incidents mentioned in the text.

4. **Coreference Resolution**:
    - **Description**: Resolving references to the same entity when mentioned with different expressions or pronouns.
    - **Purpose**: Ensures that information about a specific entity is properly linked and not fragmented across the text.
5. **Template-Based Extraction**:
    - **Description**: Defining templates or patterns for specific types of information and extracting instances that match these templates.
    - **Purpose**: Allows for the extraction of structured information based on predefined patterns.
6. **Rule-Based Extraction**:
    - **Description**: Applying predefined rules or patterns to identify and extract specific information from the text.
    - **Purpose**: Offers a more flexible approach than template-based extraction, allowing for rule customization.
7. **Machine Learning-Based Extraction**:
    - **Description**: Training machine learning models to automatically identify and extract information based on labeled examples.
    - **Purpose**: Provides adaptability and scalability for information extraction tasks.
8. **Dependency Parsing**:
    - **Description**: Analyzing the grammatical structure of sentences to identify relationships between words and their syntactic dependencies.
    - **Purpose**: Aids in understanding the semantic structure of the text, contributing to information extraction.

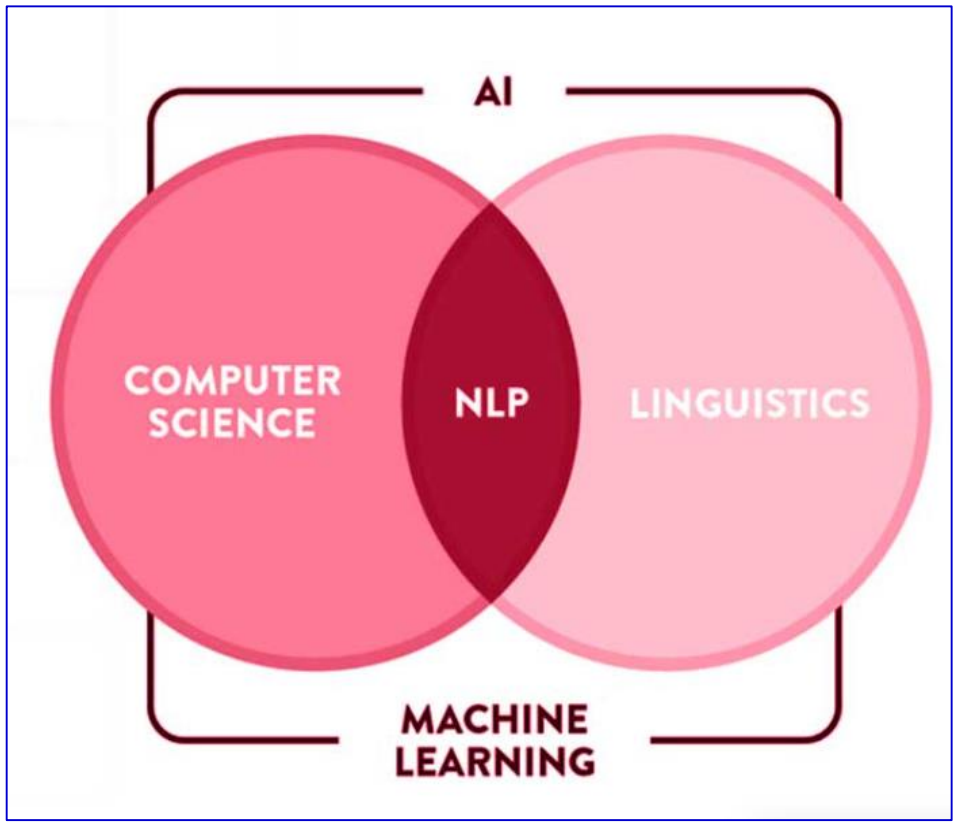
9. **Semantic Role Labeling (SRL)**:
    - **Description**: Identifying the roles that different entities play in relation to an action or event described in the text.
    - **Purpose**: Enhances the understanding of the roles and relationships between entities in a given context.
10. **Open Information Extraction**:
    - **Description**: Extracting information without relying on predefined templates or patterns, allowing for the discovery of novel relationships.
    - **Purpose**: Useful when dealing with diverse and evolving information that may not fit into predefined structures.

Information extraction is particularly valuable for converting unstructured text data into structured formats, enabling further analysis and knowledge discovery. It plays a crucial role in applications such as data integration, knowledge base construction, and building structured databases from textual sources.

Natural language processing (NLP)

- Natural language processing (NLP) refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

- NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models.
- Together, these technologies enable computers to process human language in the form of text or voice data and to 'understand' its full meaning, complete with the speaker or writer's intent and sentiment.



https://www.engati.com/glossary/natural-language-processing

https://www.ibm.com/topics/natural-language-processing#:~:text=Natural%20language%20processing%20(NLP)%20refers,same%20way%20human%20beings%20can.

KS

What is Natural language processing (NLP)?

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that focuses on the interaction between computers and humans through natural language. The goal of NLP is to enable computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant. NLP involves a variety of techniques and methods for processing and analyzing textual data, and it plays a key role in applications such as language translation, sentiment analysis, chatbots, and information retrieval.

Key components of Natural Language Processing include:

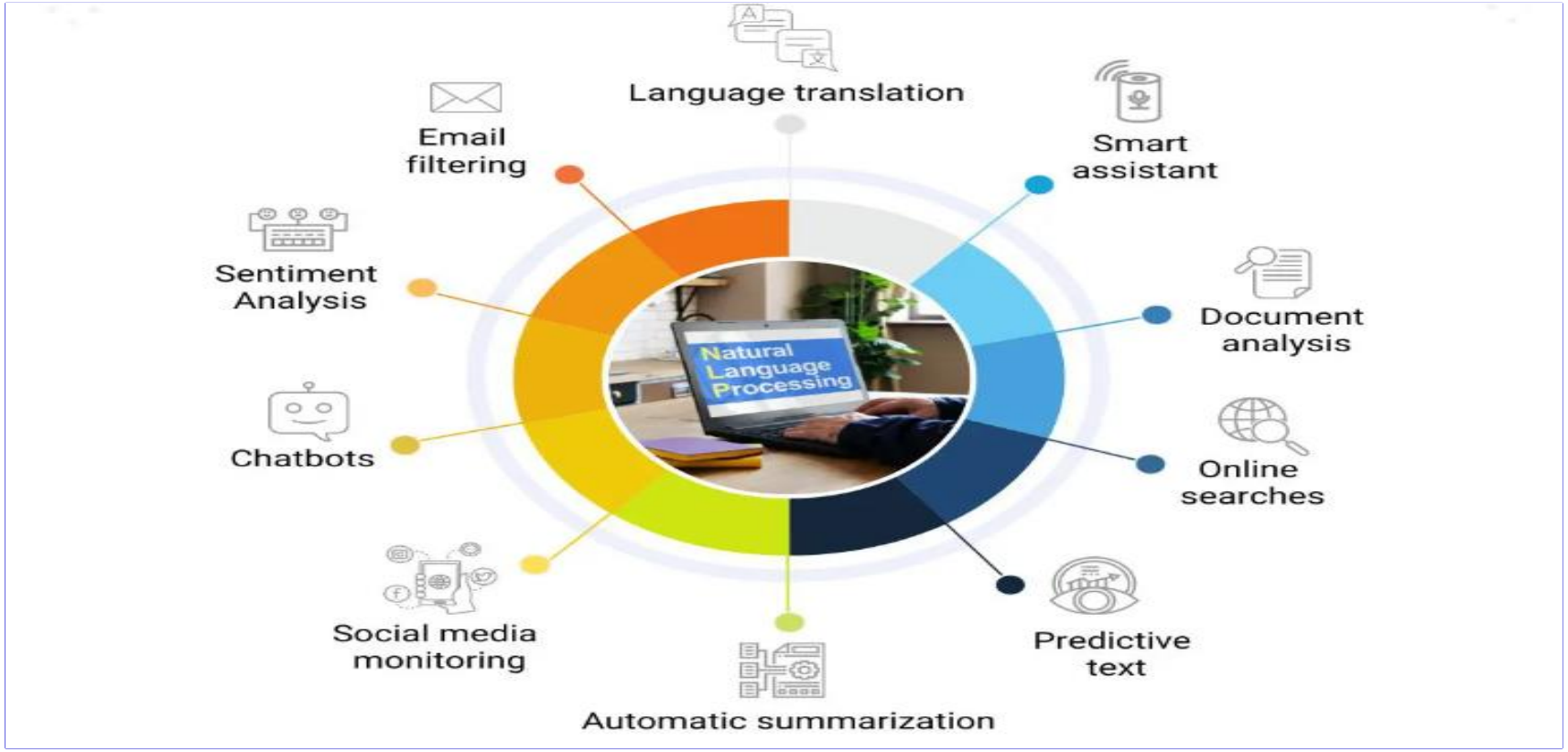
1. **Tokenization**:
   - **Description**: Breaking down text into individual units, such as words or phrases (tokens).
   - **Purpose**: Provides a foundation for further analysis by dividing text into manageable units.
2. **Part-of-Speech Tagging**:
   - **Description**: Assigning grammatical categories (e.g., nouns, verbs, adjectives) to each word in a sentence.
   - **Purpose**: Helps understand the syntactic structure of a sentence and the roles of individual words.

3. **Named Entity Recognition (NER)**:
    - **Description**: Identifying and classifying entities, such as names of people, organizations, locations, dates, etc., in a text.
    - **Purpose**: Extracts structured information and identifies key entities within the text.
4. **Syntax and Grammar Parsing**:
    - **Description**: Analyzing the grammatical structure of sentences to understand the relationships between words.
    - **Purpose**: Facilitates understanding of sentence structure and meaning.
5. **Semantic Analysis**:
    - **Description**: Analyzing the meaning of words, phrases, and sentences in the context of the entire text.
    - **Purpose**: Goes beyond syntax to understand the intended meaning of language.
6. **Sentiment Analysis**:
    - **Description**: Determining the sentiment expressed in a piece of text, such as positive, negative, or neutral.
    - **Purpose**: Helps gauge opinions and emotions expressed in text data.
7. **Machine Translation**:
    - **Description**: Automatically translating text from one language to another.
    - **Purpose**: Facilitates communication between speakers of different languages.

7. **Machine Translation**:
    - **Description**: Automatically translating text from one language to another.
    - **Purpose**: Facilitates communication between speakers of different languages.
8. **Information Retrieval**:
    - **Description**: Extracting relevant information from large datasets or documents.
    - **Purpose**: Aids in finding and retrieving specific information based on user queries.
9. **Question Answering**:
    - **Description**: Developing systems that can understand and answer questions posed in natural language.
    - **Purpose**: Enhances human-computer interaction and information retrieval.
10. **Speech Recognition**:
    - **Description**: Transcribing spoken words into written text.
    - **Purpose**: Enables hands-free interaction with devices and facilitates voice commands.
11. **Chatbots and Virtual Assistants**:
    - **Description**: Creating computer programs that can engage in natural language conversations with users.
    - **Purpose**: Enhances user experience by providing information or assistance in a conversational manner.

NLP draws on various linguistic, statistical, and machine learning techniques to achieve its objectives. The field continues to evolve with advancements in deep learning and other AI technologies, making it a crucial component in many applications that involve human-computer interaction.

Application of NLP

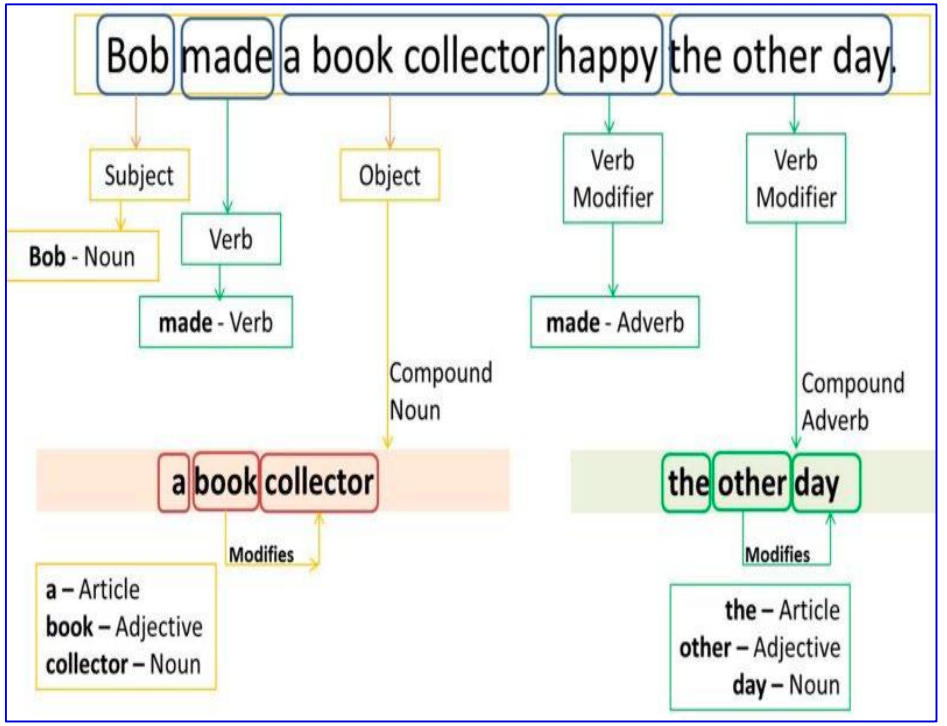


https://datasciencedojo.com/blog/natural-language-processing-applications/

Natural language processing (NLP)

Summarization: This technique provides a synopsis of long pieces of text to create a concise, coherent summary of a document's main points.

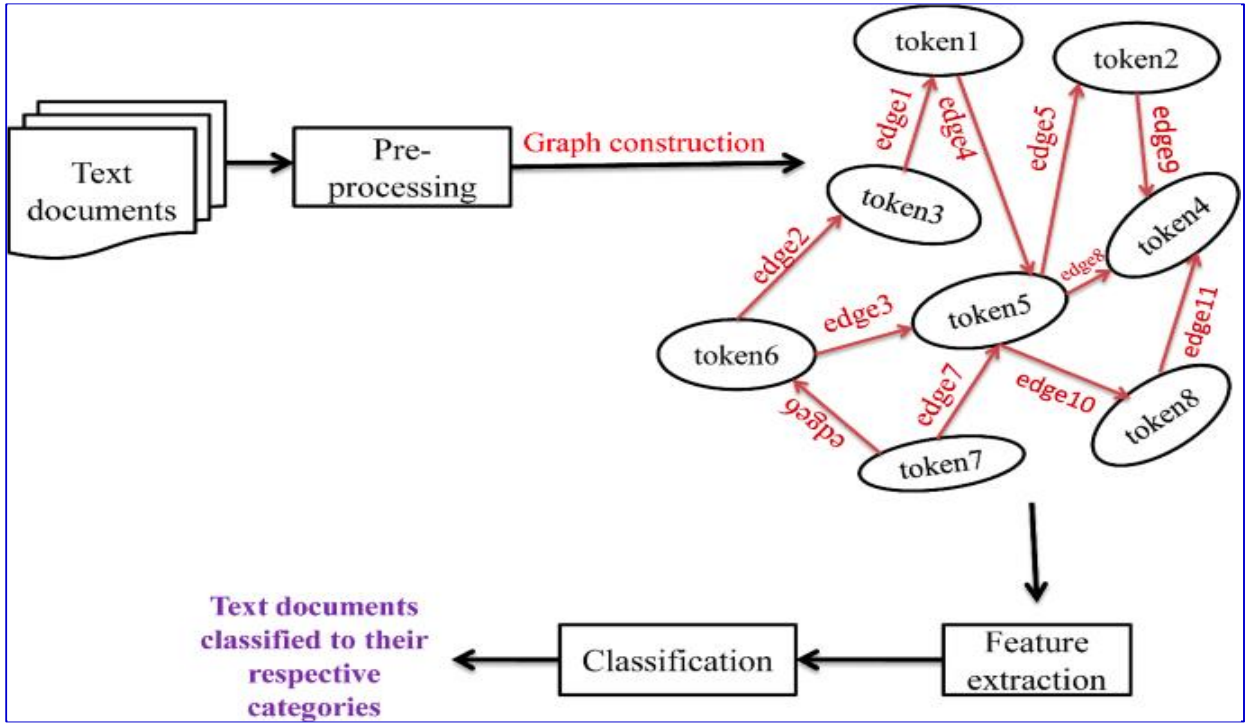
Part-of-Speech (PoS) tagging: This technique assigns a tag to every token in a document based on its part of speech—i.e. denoting nouns, verbs, adjectives, etc. This step enables semantic analysis on unstructured text.



https://www.freecodecamp.org/news/an-introduction-to-part-of-speech-tagging-and-the-hidden-markov-model-953d45338f24/

Natural language processing (NLP)

Text categorization: Responsible for analyzing text documents and classifying them based on predefined topics or categories. This sub-task is particularly helpful when categorizing synonyms and abbreviations

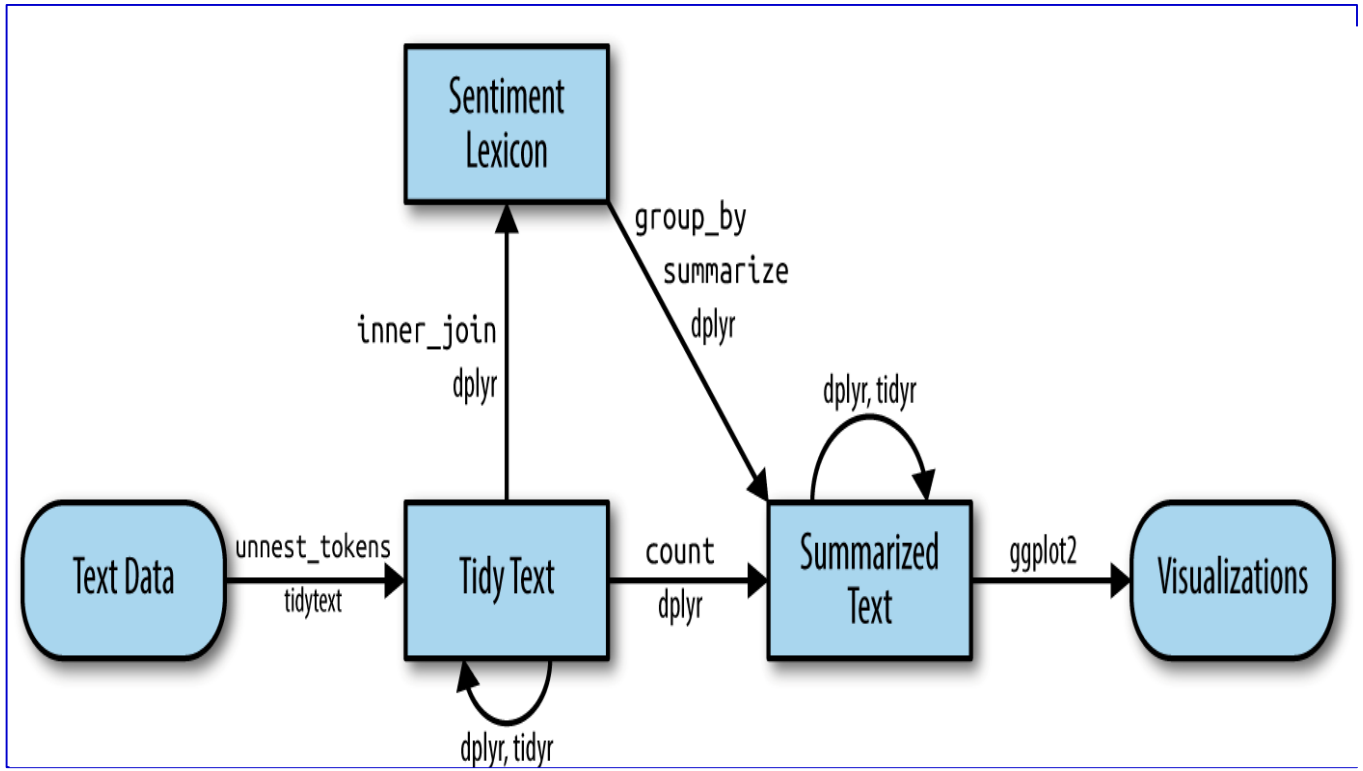


https://link.springer.com/article/10.1007/s10462-020-09919-1

Natural language processing (NLP)

Sentiment analysis

- This task detects positive or negative sentiment from internal or external data sources, allowing you to track changes in customer attitudes over time.
- It is commonly used to provide information about perceptions of brands, products, and services. These insights can propel businesses to connect with customers and improve processes and user experiences.



https://www.tidytextmining.com/sentiment.html

Data Mining

- Data mining is the process of identifying patterns and extracting useful insights from big data sets.

- This practice evaluates both structured and unstructured data to identify new information, and it is commonly utilized to analyze consumer behaviors within marketing and sales.

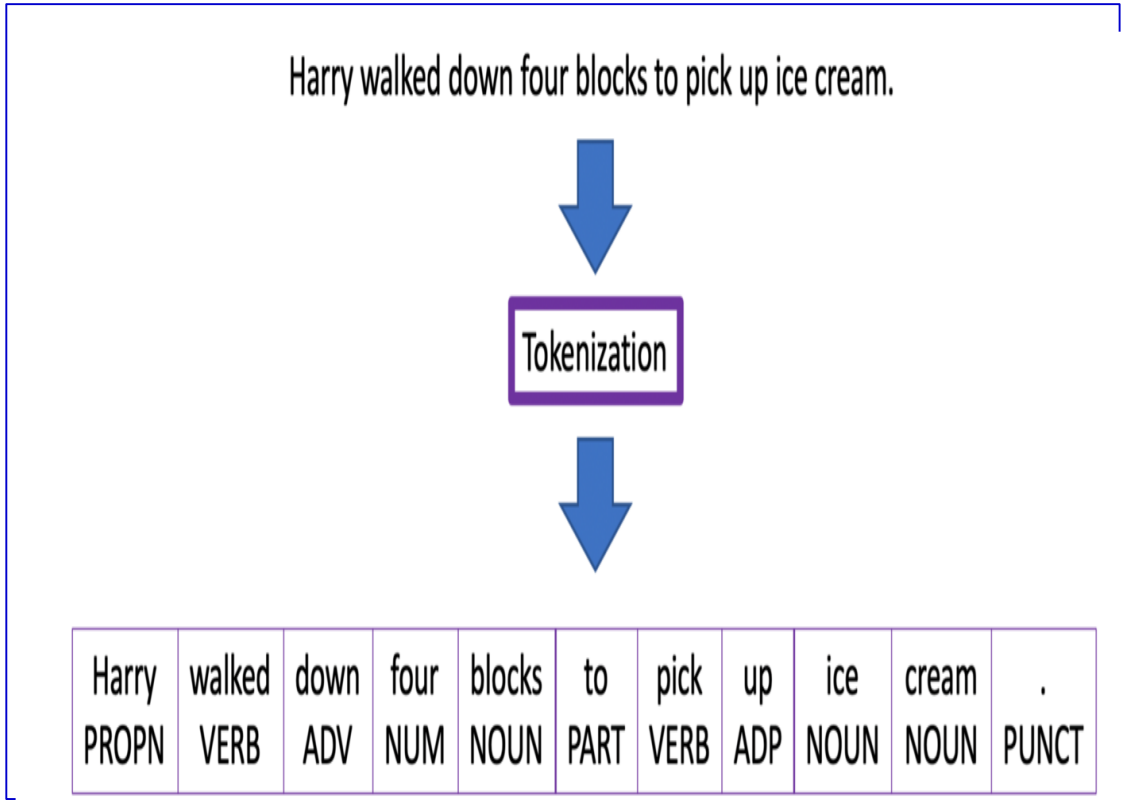
- Text mining is essentially a sub-field of data mining as it focuses on bringing structure to unstructured data and analyzing it to generate novel insights.

Information Retrieval

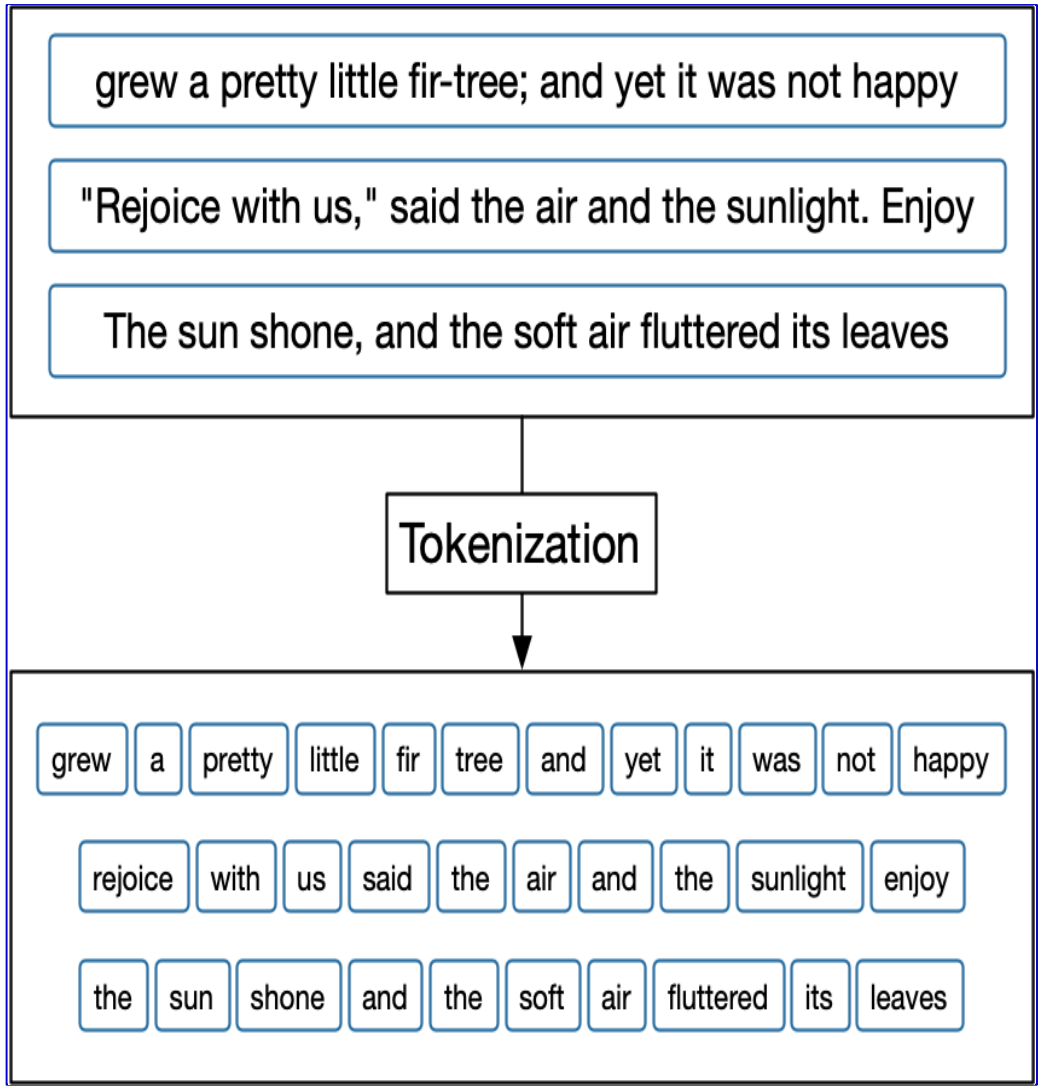
- The automatic extraction of structured data such as entities, entities relationships, and attributes describing entities from an unstructured source is called **information extraction**.
- Information retrieval is commonly used with library catalog systems and popular search engines such as Google. Some common IR subtasks include:

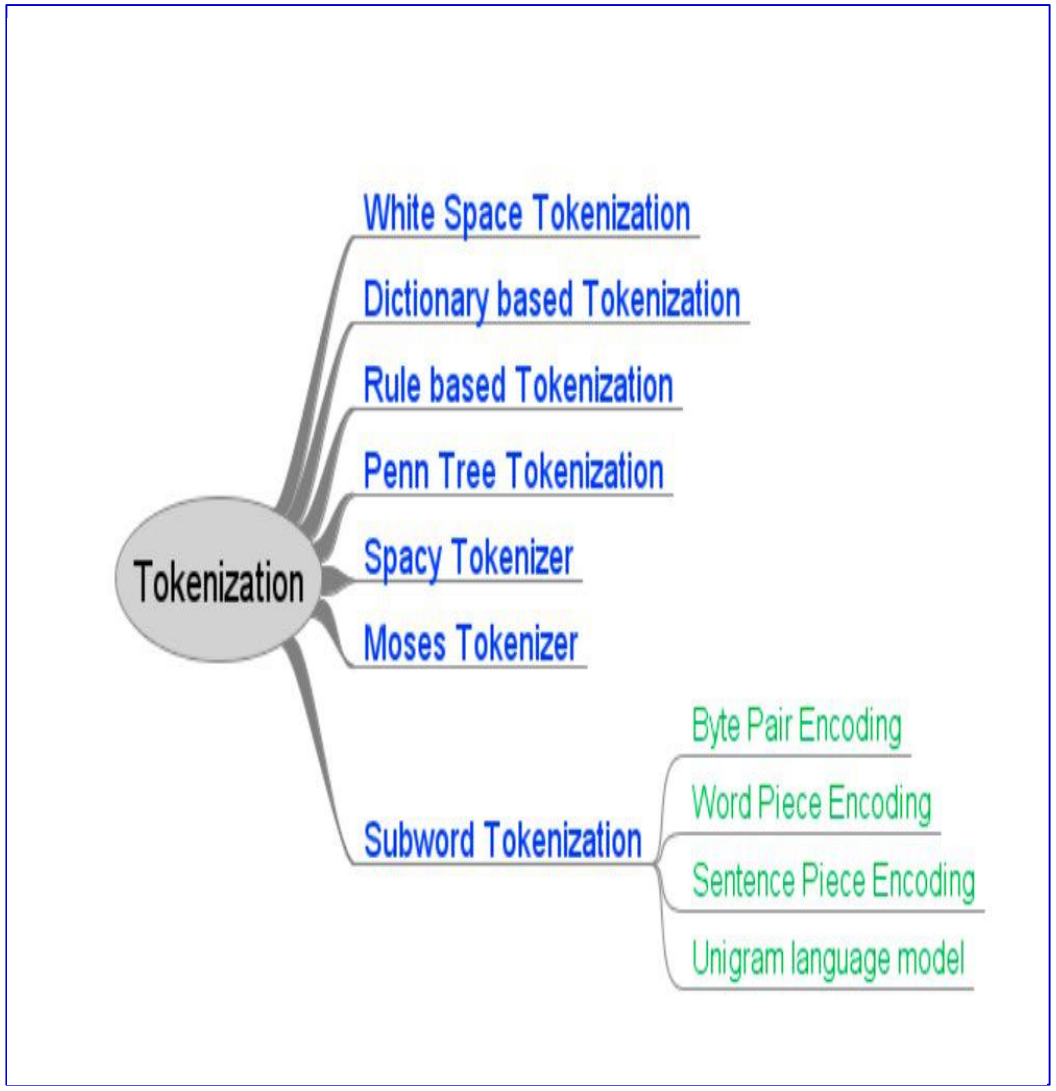
Tokenization: This is the process of breaking out long-form text into sentences and words called “tokens”. These are, then, used in the models, like bag-of-words, for text clustering and document matching tasks.

Stemming: This refers to the process of separating the prefixes and suffixes from words to derive the root word form and meaning. This technique improves information retrieval by reducing the size of indexing files.



Information Retrieval





https://smltar.com/tokenization.html
https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4

Information Retrieval

Stemming: This refers to the process of separating the prefixes and suffixes from words to derive the root word form and meaning. This technique improves information retrieval by reducing the size of indexing files.

Lemmatization:

Lemmatization is a method responsible for grouping different inflected forms of words into the root form, having the same meaning.

S.No	Stemming	Lemmatization
1	Stemming is faster because it chops words without knowing the context of the word in given sentences.	Lemmatization is slower as compared to stemming but it knows the context of the word before proceeding.
2	It is a rule-based approach.	It is a dictionary-based approach.
3	Accuracy is less.	Accuracy is more as compared to Stemming.
4	When we convert any word into root-form then stemming may create the non-existence meaning of a word.	Lemmatization always gives the dictionary meaning word while converting into root-form.
5	Stemming is preferred when the meaning of the word is not important for analysis. Example: Spam Detection	Lemmatization would be recommended when the meaning of the word is important for analysis. Example: Question Answer
6	For Example: "Studies" => "Studi"	For Example: "Studies" => "Study"

https://www.analyticssteps.com/blogs/what-stemming-and-lemmatization-nlp

Text Mining Applications

Customer service

- There are various ways in which we solicit customer feedback from our users.
- When combined with text analytics tools, feedback systems, such as chatbots, customer surveys, NPS (net-promoter scores) , online reviews, support tickets, and social media profiles, enable companies to improve their customer experience with speed.
- Text mining and sentiment analysis can provide a mechanism for companies to prioritize key pain points for their customers, allowing businesses to respond to urgent issues in real-time and increase customer satisfaction

Text Mining Applications

Risk management

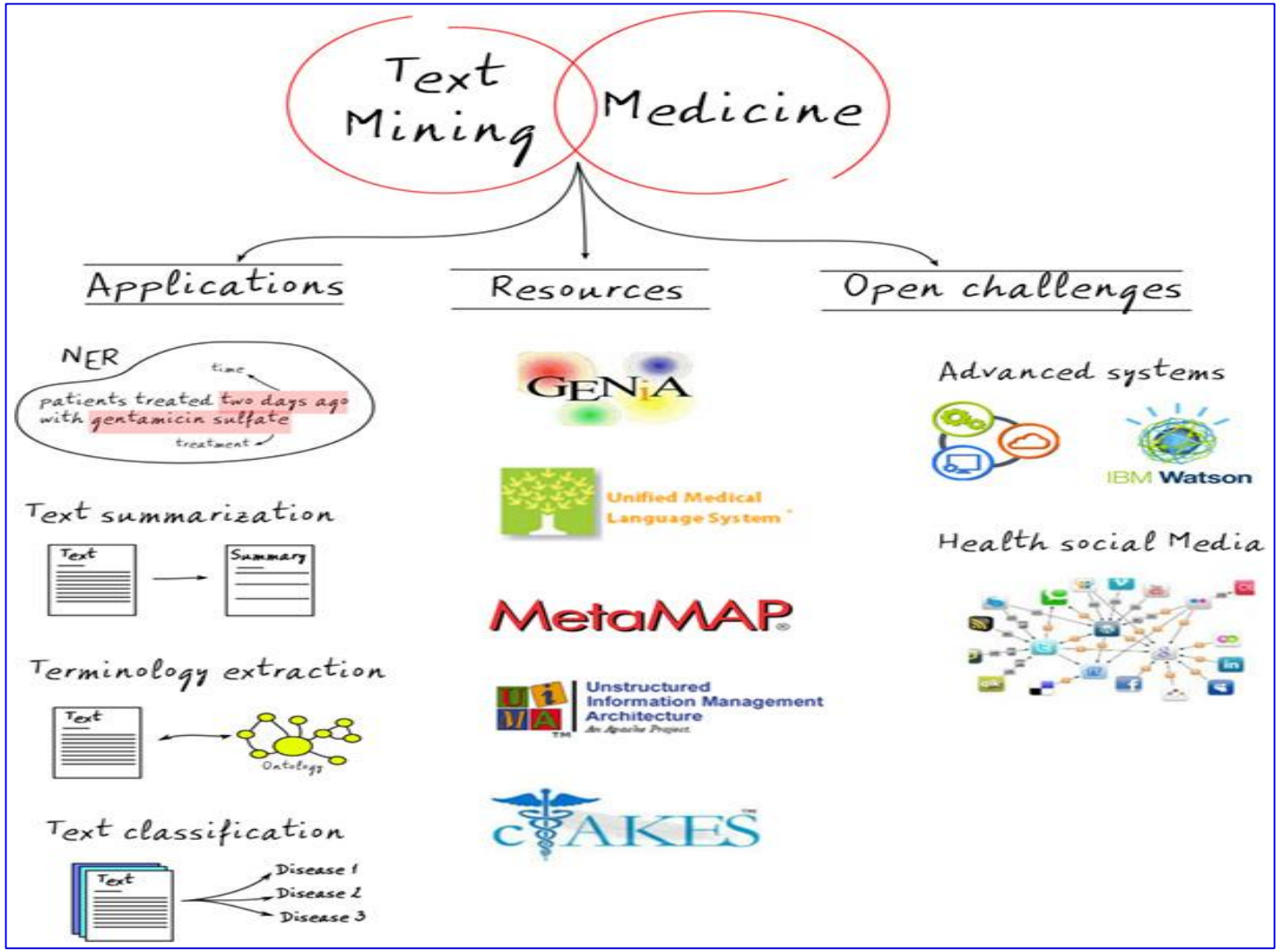
- Text mining can provide insights into industry trends and financial markets by monitoring shifts in sentiment and by extracting information from analyst reports and whitepapers.

- This is particularly valuable to banking institutions as this data provides more confidence when considering business investments across various sectors

Text Mining Applications

Healthcare:

- Text mining techniques have been increasingly valuable to researchers in the biomedical field, particularly for clustering information.
- Manual investigation of medical research can be costly and time-consuming; text mining provides an automated method for extracting valuable information from medical literature.
- Telemedicine.

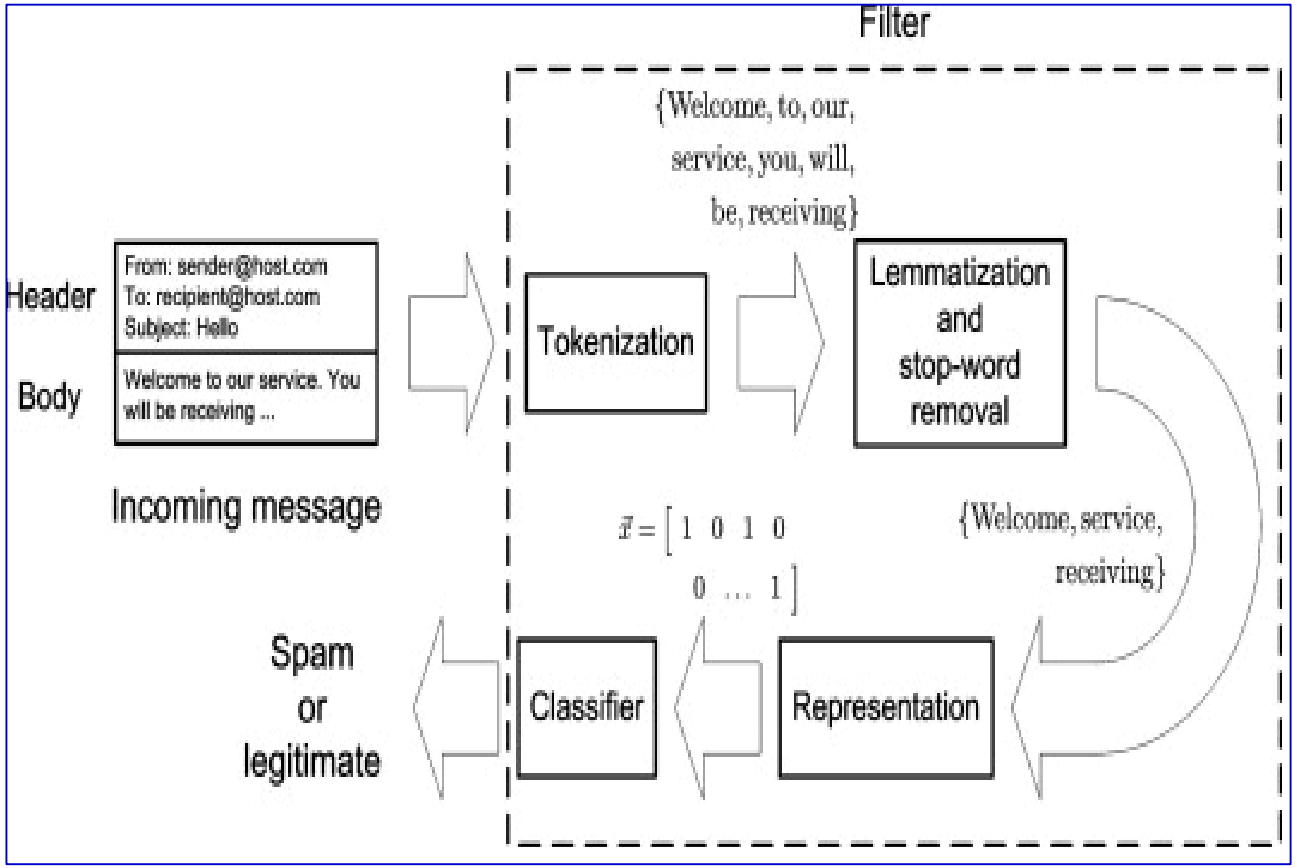


https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1302

Text Mining Applications

Spam filtering

- Spam frequently serves as an entry point for hackers to infect computer systems with malware.
- Text mining can provide a method to filter and exclude these emails from inboxes, improving the overall user experience and minimizing the risk of cyber-attacks to end users.



Text Mining with R

Data and packages prepare

Code:

```
Needed <- c("tm", "SnowballCC", "RColorBrewer", "ggplot2", "wordcloud", "biclust", "cluster", "igraph", "fpc")
install.packages(Needed, dependencies = TRUE)
install.packages("Rcampdf", repos = "http://datacube.wu.ac.at/", type = "source")
```

Data:

Trump speech:D:\TA Introducing the data Science\2023 notes\Topic-8

Please put the data you want analysis into one folder and set the work path

EX: cname = file.path("speech:D:\TA Introducing the data Science\2023 notes\Topic-8")

Data summary

Code:

```
library(tm)
docs <- VCorpus(DirSource(cname))
summary(docs)
```

File list :

```
                                                Length Class               Mode
Trump Black History Month Speech.txt            2      PlainTextDocument list
Trump CIA Speech.txt                            2      PlainTextDocument list
Trump Congressional Address.txt                 2      PlainTextDocument list
Trump CPAC Speech.txt                           2      PlainTextDocument list
Trump Florida Rally 2-18-17.txt                 2      PlainTextDocument list
Trump Immigration Speech 8-31-16.txt            2      PlainTextDocument list
Trump Inauguration Speech.txt                   2      PlainTextDocument list
Trump National Prayer Breakfast.txt             2      PlainTextDocument list
Trump Nomination Speech.txt                     2      PlainTextDocument list
Trump Police Chiefs Speech.txt                  2      PlainTextDocument list
Trump Response to Healthcare Bill Failure.txt 2      PlainTextDocument list
```

Data summary

Code:

```
inspect(docs[1])
writeLines(as.character(docs[1]))
```

Firs file word:

```
list(list(content = c("Well, the election, it came out really well. Next time we<e2>\u0080l triple the nu
mber or quadruple it. We want to get it over 51, right? At least 51.", "", "Well this is Black History Mon
th, so this is our little breakfast, our little get-together. Hi Lynn, how are you? Just a few notes. Duri
ng this month, we honor the tremendous history of African-Americans throughout our country. Throughout the
 world, if you really think about it, right? And their story is one of unimaginable sacrifice, hard work,
and faith in America. I<e2>\u0080e gotten a real glimpse<e2>\u0080uring the campaign, I<e2>\u0080 go ar
ound with Ben to a lot of different places I wasn<e2>\u0080 so familiar with. They<e2>\u0080e incredible
 people. And I want to thank Ben Carson, who<e2>\u0080 gonna be heading up HUD. That<e2>\u0080 a big job
. That<e2>\u0080 a job that<e2>\u0080 not only housing, but it<e2>\u0080 mind and spirit. Right, Ben? A
nd you understand, nobody<e2>\u0080 gonna be better than Ben.",
"", "Last month, we celebrated the life of Reverend Martin Luther King, Jr., whose incredible example is u
nique in American history. You read all about Dr. Martin Luther King a week ago when somebody said I took
the statue out of my office. It turned out that that was fake news. Fake news. The statue is cherished, it
```

Data preprocessing

Remove Punctuation

```
docs <- tm_map(docs,removePunctuation)
writeLines(as.character(docs[1])) # Check to see if it worked.
```

Output:

```
list(list(content = c("Well the election it came out really well Next time wee2\u0080l triple the number
or quadruple it We want to get it over 51 right At least 51", "", "Well this is Black History Month so thi
s is our little breakfast our little gettogether Hi Lynn how are you Just a few notes During this month we
 honor the tremendous history of AfricanAmericans throughout our country Throughout the world if you reall
y think about it right And their story is one of unimaginable sacrifice hard work and faith in America Ie2
\u0080e gotten a real glimpsee2\u0080uring the campaign Ie2\u0080 go around with Ben to a lot of differ
ent places I wasne2\u0080 so familiar with Theye2\u0080e incredible people And I want to thank Ben Carso
n whoe2\u0080 gonna be heading up HUD Thate2\u0080 a big job Thate2\u0080 a job thate2\u0080 not only
housing but ite2\u0080 mind and spirit Right Ben And you understand nobodye2\u0080 gonna be better than
Ben",
```

Data preprocessing

Remove ascii character that did not translate, so it had to be removed

```
for (j in seq(docs)){
docs[[j]] <- gsub("/", " ", docs[[j]])
docs[[j]] <- gsub("@", " ", docs[[j]])
docs[[j]] <- gsub("\\|", " ", docs[[j]])
docs[[j]] <- gsub('e2\u0080e ', " ", docs[[j]])
docs[[j]] <- gsub('e2\u0080', " ", docs[[j]])
docs[[j]] <- gsub("e2\u2028e", " ", docs[[j]])
docs[[j]] <- gsub("e2\u2028", " ", docs[[j]])
#docs[[j]] <- gsub("\u0080 ", " ", docs[[j]])# This is an ascii character that did not translate, so it had to be removed.  }
writeLines(as.character(docs[1])) # You can check a document (in this case # the first) to see if it worked.
```

Output

```
list(c("Well the election it came out really well Next time we l triple the number or quadruple it We wan
t to get it over 51 right At least 51", "", "Well this is Black History Month so this is our little breakf
ast our little gettogether Hi Lynn how are you Just a few notes During this month we honor the tremendous
history of AfricanAmericans throughout our country Throughout the world if you really think about it right
 And their story is one of unimaginable sacrifice hard work and faith in America I e gotten a real glimps
e uring the campaign I  go around with Ben to a lot of different places I wasn  so familiar with They
e incredible people And I want to thank Ben Carson who  gonna be heading up HUD That  a big job That
a job that  not only housing but it  mind and spirit Right Ben And you understand nobody  gonna be bett
er than Ben",
```

Data preprocessing

Remove number and Converting to lowercase

```
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, tolower)
docs <- tm_map(docs, PlainTextDocument)
DocsCopy <- docs
writeLines(as.character(docs[1])) # Check to see if it worked.
```

Output

```
list(list(content = c("well the election it came out really well next time we l triple the number or quad
ruple it we want to get it over  right at least ", "", "well this is black history month so this is our li
ttle breakfast our little gettogether hi lynn how are you just a few notes during this month we honor the
tremendous history of africanamericans throughout our country throughout the world if you really think abo
ut it right and their story is one of unimaginable sacrifice hard work and faith in america i e gotten a
real glimpse uring the campaign i  go around with ben to a lot of different places i wasn  so familiar
with they e incredible people and i want to thank ben carson who  gonna be heading up hud that  a big j
ob that  a job that  not only housing but it  mind and spirit right ben and you understand nobody  gon
na be better than ben",
"", "last month we celebrated the life of reverend martin luther king jr whose incredible example is uniqu
e in american history you read all about dr martin luther king a week ago when somebody said i took the st
atue out of my office it turned out that that was fake news fake news the statue is cherished it  one of
```

Data preprocessing

Remove stop words

```
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, PlainTextDocument)
writeLines(as.character(docs[1])) # Check to see if it worked.
docs <- tm_map(docs, removeWords, c("syllogism", "tautology"))
```

Output

```
list(list(content = c("well  election  came  really well next time  l triple  number  quadruple  want  g
et  right  least ", "", "well  black history month  little breakfast  little gettogether hi lynn
just  notes  month  honor  tremendous history  africanamericans throughout  country throughout  world
really think  right  story  one  unimaginable sacrifice hard work  faith  america  e gotten  real glimp
se uring  campaign  go around  ben  lot  different places  wasn  familiar  e incredible people  w
ant  thank ben carson  gonna  heading  hud  big job  job  housing  mind  spirit right ben
 understand nobody  gonna  better  ben",
"", "last month  celebrated  life  reverend martin luther king jr whose incredible example  unique  americ
an history  read  dr martin luther king  week ago  somebody said  took  statue  office  turned  fake
 news fake news  statue  cherished  one  favorite things  nd  good ones  lincoln  jefferson  d
r martin luther king  said  statue  bust  martin luther king  taken  office  never even touched  thi
nk  disgrace  way  press  unfortunate", "", "  proud now  museum  national mall  people can le
arn  reverend king  many  things frederick douglass  example  somebody  done  amazing job  recognized
 noticed harriet tubman rosa parks  millions  black americans  made america  today big impact",
```

Data preprocessing

Combining words that should stay together

```
for (j in seq(docs))
{
 docs[[j]] <- gsub("fake news", "fake_news", docs[[j]])
 docs[[j]] <- gsub("inner city", "inner-city", docs[[j]])
 docs[[j]] <- gsub("politically correct", "politically_correct", docs[[j]])
}
docs <- tm_map(docs, PlainTextDocument)
writeLines(as.character(docs[1])) # Check to see if it worked.
```

Output

```
list(list(content = c("well  election  came  really well next time  l triple  number  quadruple  want  g
et  right  least ", "", "well  black history month  little breakfast  little gettogether hi lynn
just  notes  month  honor  tremendous history  africanamericans throughout  country throughout  world
really think  right  story  one  unimaginable sacrifice hard work  faith  america  e gotten  real glimp
se uring  campaign  go around  ben  lot  different places  wasn  familiar  e incredible people  w
ant  thank ben carson  gonna  heading  hud  big job  job  housing  mind  spirit right ben
 understand nobody  gonna  better  ben",
"", "last month  celebrated  life  reverend martin luther king jr whose incredible example  unique  americ
an history  read  dr martin luther king  week ago  somebody said  took  statue  office  turned  fake
_news fake_news  statue  cherished  one  favorite things  nd  good ones  lincoln  jefferson  d
r martin luther king  said  statue  bust  martin luther king  taken  office  never even touched  thi
nk  disgrace  way  press  unfortunate", "", "  proud now  museum  national mall  people can le
arn  reverend king  many  things frederick douglass  example  somebody  done  amazing job  recognized
 noticed harriet tubman rosa parks  millions  black americans  made america  today big impact",
"", "  proud  honor  heritage  will  honoring  folks  table  almost  cases  great friends  supporte
rs darrell  met darrell  defending  television  people  side  argument didn  chance right  pa
```

Data preprocessing

Removing stem word (e.g., “ing” , “es” , “s”)

```
docs_st <- tm_map(docs, stemDocument)
docs_st <- tm_map(docs_st, PlainTextDocument)
writeLines(as.character(docs_st[1])) # Check to see if it worked.
```

Output

```
", "", "well black histori month littl breakfast littl gettogeth hi lynn just note month honor tremend his
tori africanamerican throughout countri throughout world realli think right stori one unimagin sacrific ha
rd work faith america e gotten real glimps ure campaign  go around ben lot differ place wasn  familiar
 e incred peopl want thank ben carson  gonna head hud  big job  job  hous  mind spirit right ben und
erstand nobodi  gonna better ben",
"", "last month celebr life reverend martin luther king jr whose incred exampl uniqu american histori read
 dr martin luther king week ago somebodi said took statu offic turn fake_new fake_new statu cherish  one
favorit thing nd good one lincoln jefferson dr martin luther king said statu bust martin luther king take
n offic never even touch think disgrac  way press unfortun", "", "proud now museum nation mall peopl can
learn reverend king mani thing frederick douglass exampl somebodi  done amaz job recogn notic harriet tub
man rosa park million black american made america today big impact",
"", " proud honor heritag will honor folk tabl almost case great friend support darrel  met darrel defen
d televis peopl side argument didn  chanc right pari done amaz job hostil cnn communiti  l seven peopl
pari l take pari seven don  watch cnn don  get see much use don  like watch fake_new fox treat nice wh
erev fox thank", "", "e gonna need better school need soon need job need better wage lot better wage e g
```

Data preprocessing

Stripping unnecessary whitespace

```
docs <- tm_map(docs, stripWhitespace)
writeLines(as.character(docs[1])) # Check to see if it worked.
```

Output

```
list(list(content = c("well election came really well next time 1 triple number quadruple want get right
least ", "", "well black history month little breakfast little gettogether hi lynn just notes month honor
tremendous history africanamericans throughout country throughout world really think right story one unima
ginable sacrifice hard work faith america e gotten real glimpse uring campaign  go around ben lot diffe
rent places wasn  familiar e incredible people want thank ben carson  gonna heading hud  big job  job
  housing  mind spirit right ben understand nobody  gonna better ben",
"", "last month celebrated life reverend martin luther king jr whose incredible example unique american hi
story read dr martin luther king week ago somebody said took statue office turned fake_news fake_news stat
ue cherished  one favorite things nd good ones lincoln jefferson dr martin luther king said statue bust
martin luther king taken office never even touched think disgrace  way press unfortunate", "", " proud no
w museum national mall people can learn reverend king many things frederick douglass example somebody  do
ne amazing job recognized noticed harriet tubman rosa parks millions black americans made america today bi
g impact"
```

Data preprocessing

Finish preprocessing

```
docs <- tm_map(docs, PlainTextDocument)
writeLines(as.character(docs[1])) # Check to see if it worked.
```

Output

```
list(list(content = c("well election came really well next time l triple number quadruple want get
least ", "", "well black history month little breakfast little gettogether hi lynn just notes month
tremendous history africanamericans throughout country throughout world really think right story on
ginable sacrifice hard work faith america e gotten real glimpse uring campaign  go around ben lot d
rent places wasn  familiar e incredible people want thank ben carson  gonna heading hud  big job  j
  housing  mind spirit right ben understand nobody  gonna better ben",
"", "last month celebrated life reverend martin luther king jr whose incredible example unique amer
story read dr martin luther king week ago somebody said took statue office turned fake_news fake_ne
ue cherished  one favorite things nd good ones lincoln jefferson dr martin luther king said statue
martin luther king taken office never even touched think disgrace  way press unfortunate", "", " pr
```

Stage the Data

Create a document term matrix

```
dtm <- DocumentTermMatrix(docs)
dtm
tdm <- TermDocumentMatrix(docs)
Tdm
```

Output

```
> dtm
<<DocumentTermMatrix (documents: 11, terms: 3624)>>
Non-/sparse entries: 8317/31547
Sparsity           : 79%
Maximal term length: 19
Weighting          : term frequency (tf)
> tdm <- TermDocumentMatrix(docs)
> tdm
<<TermDocumentMatrix (terms: 3624, documents: 11)>>
Non-/sparse entries: 8317/31547
Sparsity           : 79%
Maximal term length: 19
Weighting          : term frequency (tf)
```

Explore data

Terms frequency

```
freq <- colSums(as.matrix(dtm))
length(freq)
```

Output

```
> length(freq)
[1] 3624
```

as.matrix(dtm)

	abandon	abandoned	abandonment	abc	abe	abilities	ability	able	abolish	abolishing	abraham
character.0.	0	0	0	0	0	0	0	0	0	0	0
character.0..1	0	0	0	0	0	1	0	0	0	0	0
character.0..2	0	0	1	0	0	0	1	4	0	0	1
character.0..3	0	0	0	1	0	0	0	1	0	0	0
character.0..4	0	0	0	0	1	0	0	1	0	0	1
character.0..5	0	0	0	0	0	0	1	2	0	1	0
character.0..6	0	0	0	0	0	0	0	0	0	0	0
character.0..7	0	0	0	0	0	0	0	0	0	0	0
character.0..8	1	1	0	0	0	0	0	2	1	0	0
character.0..9	0	0	0	0	0	0	0	2	0	0	0
character.0..10	0	0	0	0	0	0	1	0	0	0	0

```
> dim(as.matrix(dtm))
[1]   11 3624
```

Explore data

Start by removing sparse terms

```
dtms <- removeSparseTerms(dtm, 0.2) # This makes a matrix that is 20% empty space, maximum.
Dtms
```

Output

```
<<DocumentTermMatrix (documents: 11, terms: 87)>>
Non-/sparse entries: 848/109
Sparsity           : 11%
Maximal term length: 11
Weighting          : term frequency (tf)
```

Definition

Term frequency (TF)

how many times a term occurs in a document

Word Frequency (WF)

how many times a word occurs in all document

Frequency-inverse document frequency (TF-IDF)

all words in the corpus are not equally important

IDF: Log(N/d),

A corpus has N documents, and a term appears in d of them

A terms most of the documents will have, the IDF value will be low. Otherwise, will be large

"abandon" IDF= log(11/1) "able" IDF= log(11/6)

Word Frequency

Most and least frequently occurring words

```
head(table(freq), 20) # The ", 20" indicates that we only want the first 20 frequencies. Feel free to change that number.
tail(table(freq), 20) # The ", 20" indicates that we only want the last 20 frequencies. Feel free to change that number, as needed.
freq <- colSums(as.matrix(dtms))
freq
```

Output

```
> freq
      also     always    america   american    another       back        bad    believe
        54         24        128        107         22         75         35         60
       big       came        can       care       come    country        day  different
        45         20        107         37         55        174         36         16
      done enforcement       even       ever      every        get    getting       give
        24         43         55         42         49         79         24         25
     going       good      great      group     happen        job       just       know
       265         58        163         20         36         39         88        127
      last        law        let       life       like     little       long       look
        44         59         45         27         79         24         36         52
       lot       love       made       many       much       must     nation       need
        44         45         32        101         68         53         50         32
     never        new        now     office        one     people  president        put
        83         69        111         24        139        279         48         35
```

Word Frequency

Most and least frequently occurring words (TOP 14)

```
freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
head(freq, 14)
findFreqTerms(dtm, lowfreq=50) # Change "50" to whatever is most appropriate for your text data.
wf <- data.frame(word=names(freq), freq=freq)
head(wf)
```

Output

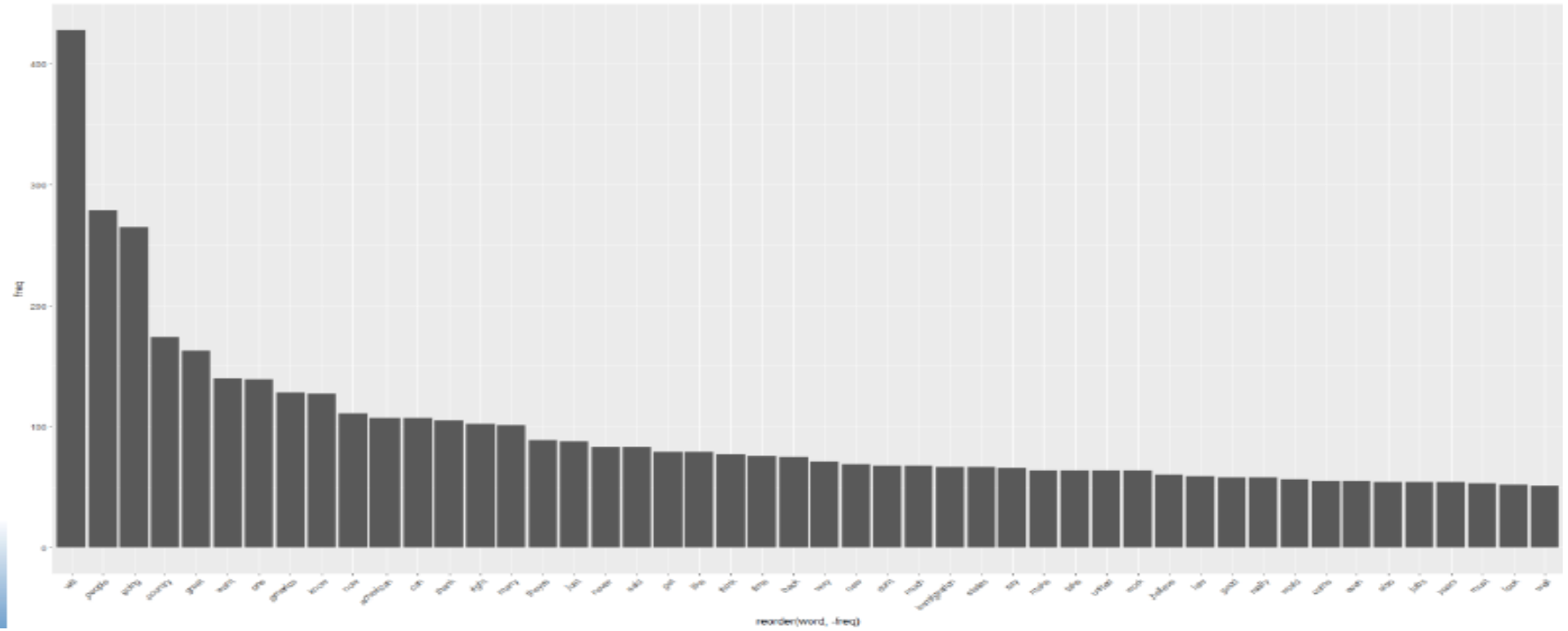
```
 will   people    going  country    great     want      one  america     know      now american
  428      279      265      174      163      140      139      128      127      111      107
  can    thank    right
  107      105      102
> head(wf)
             word freq
will         will  428
people     people  279
going       going  265
country   country  174
great       great  163
want         want  140
```

Plot Word Frequencies

Plot words that appear at least 50 times

```
library(ggplot2)
p <- ggplot(subset(wf, freq>50), aes(x = reorder(word, -freq), y = freq)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x=element_text(angle=45, hjust=1))
p
```

Output



Relationships Between Terms

Term Correlations

```
findAssocs(dtms, "think", corlimit=0.70) # specifying a correlation limit of 0.7
```

Output

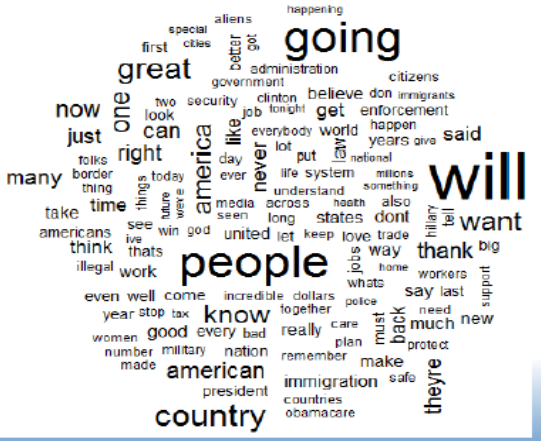
```
$think
  well   care realli  thing
  0.87   0.86   0.86   0.79
```

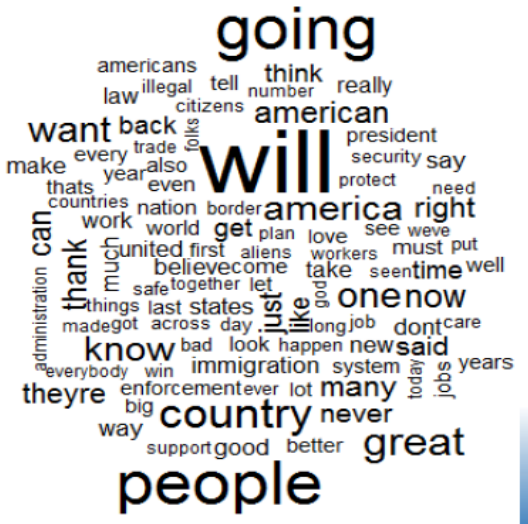
Relationships Between Terms

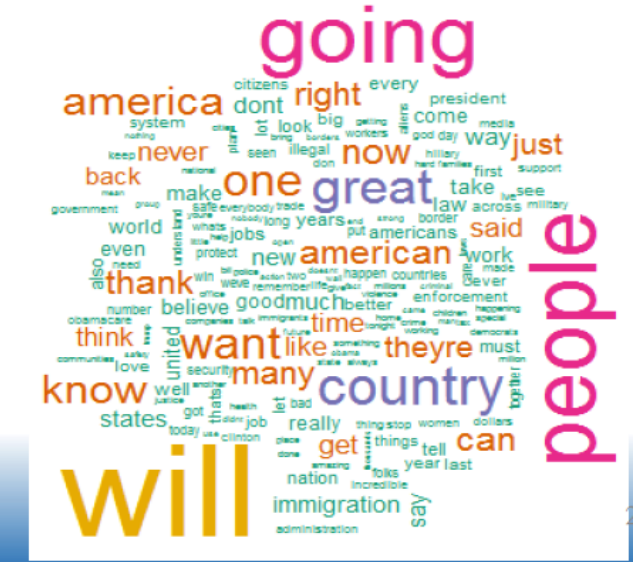
Word Cloud

```
library(wordcloud)
set.seed(142)
wordcloud(names(freq), freq, min.freq=25)
set.seed(142)
wordcloud(names(freq), freq, max.words=100)
set.seed(142)
wordcloud(names(freq), freq, min.freq=20, scale=c(5, .1), colors=brewer.pal(6, "Dark2"))
```

Output







Clustering by Term Similarity

Remove uninteresting or infrequent words

```
dtmss <- removeSparseTerms(dtm, 0.15) # This makes a matrix that is only 15% empty space, maximum.
Dtmss
```

Output

```
<<DocumentTermMatrix (documents: 11, terms: 43)>>
Non-/sparse entries: 452/21
Sparsity           : 4%
Maximal term length: 9
Weighting          : term frequency (tf)
```

Clustering by Term Similarity

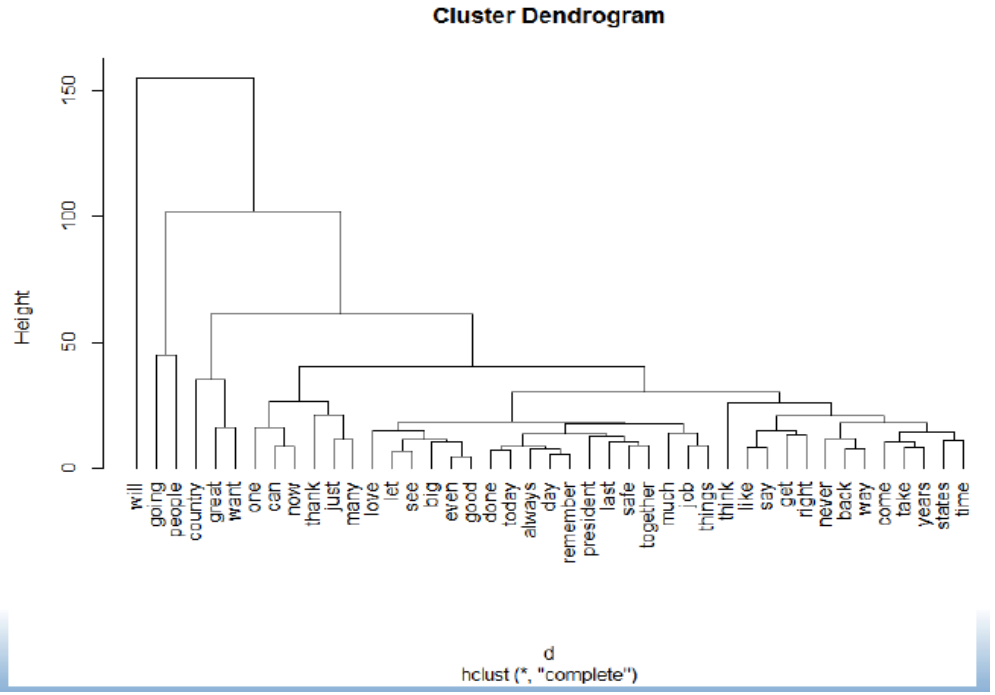
Hierarchal Clustering

```
library(cluster)
d <- dist(t(dtmss), method="euclidian")
fit <- hclust(d=d, method="complete") # for a different look try substituting: method="ward.D"
fit
plot(fit, hang=-1)
```

Output

```
Call:
hclust(d = d, method = "complete")

Cluster method   : complete
Distance         : euclidean
Number of objects: 43
```

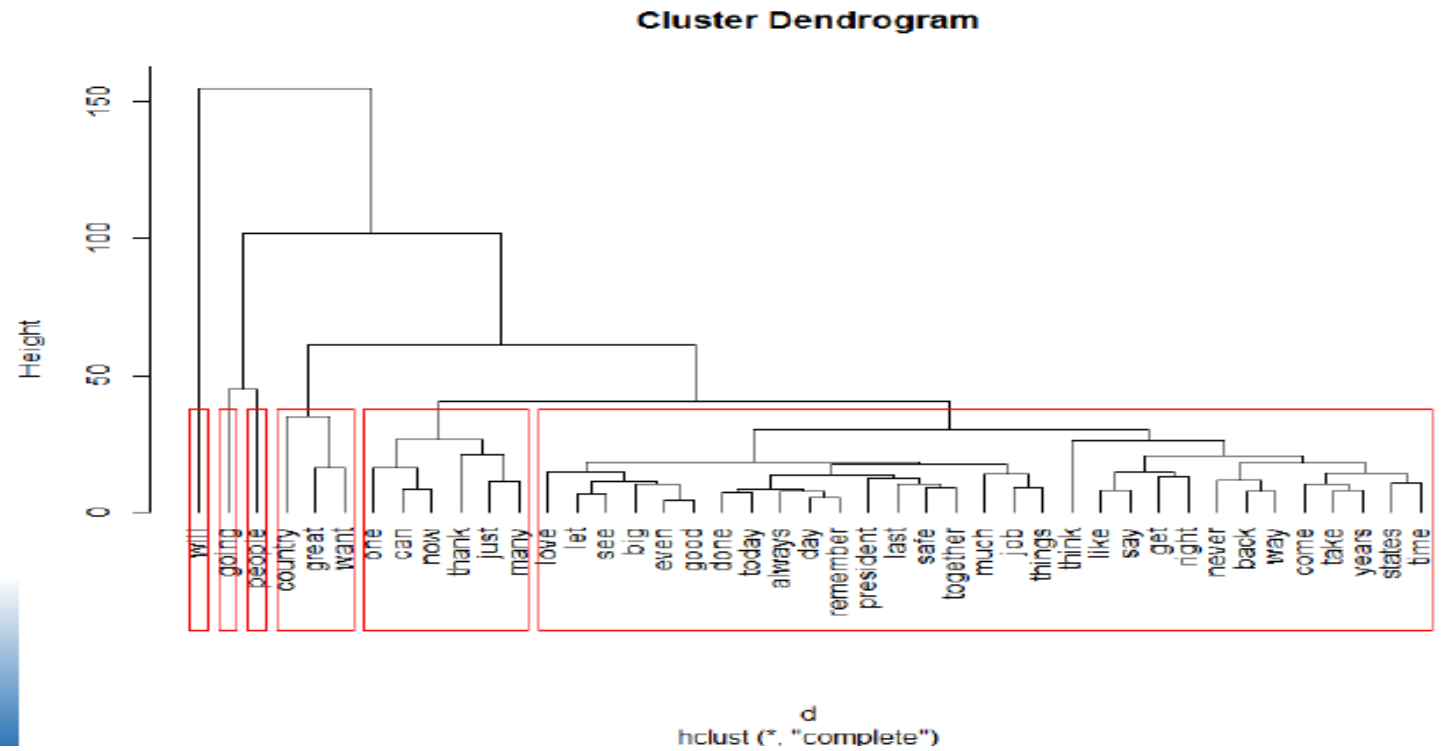


Clustering by Term Similarity

Helping to Read a Dendrogram

```
plot.new()
plot(fit, hang=-1)
groups <- cutree(fit, k=6) # "k=" defines the number of clusters you are using
rect.hclust(fit, k=6, border="red") # draw dendogram with red borders around the 6 clusters
```

Output

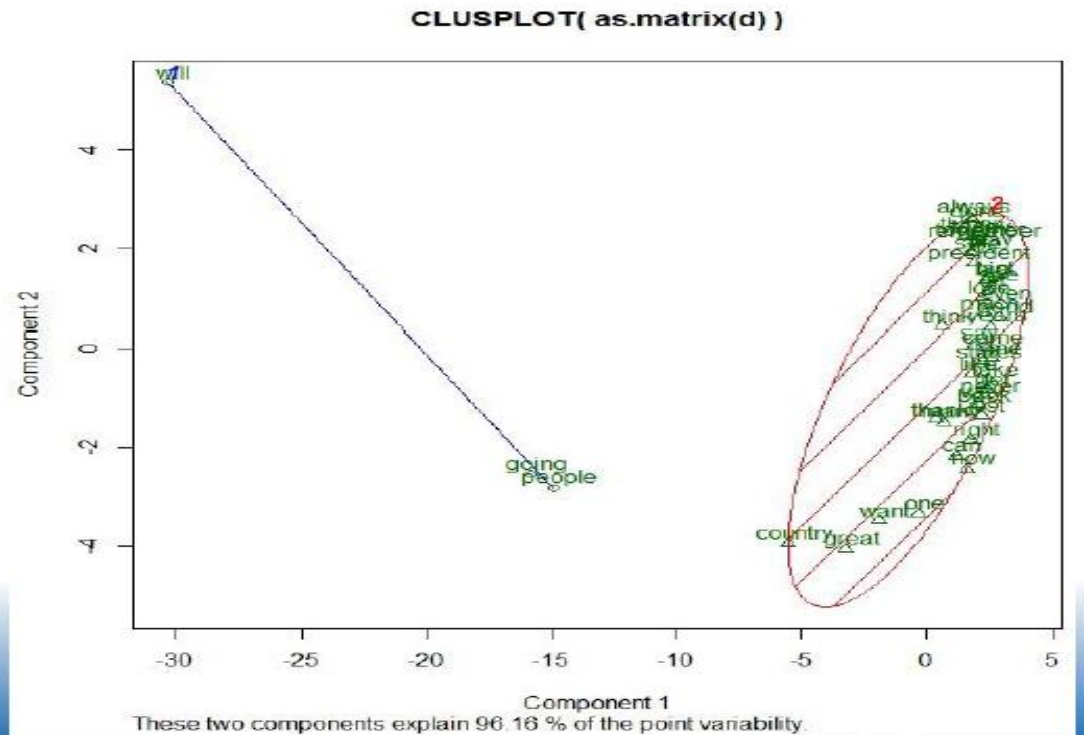


Clustering by Term Similarity

K-means clustering

```
library(fpc)
d <- dist(t(dtmss), method="euclidian")
kfit <- kmeans(d, 2)
clusplot(as.matrix(d), kfit$cluster, color=T, shade=T, labels=2, lines=0)
```

Output



Classification data

Movie comments
1000 positive and 1000 negative processed reviews. Introduced in Pang/Lee ACL 2004. Released June 2004.
http://www.cs.cornell.edu/people/pabo/movie-review-data/

TF-IDF Weighting Function

Term frequency (TF)

- how many times a term occurs in a document

Frequency-inverse document frequency (TF-IDF)

- all words in the corpus are not equally important
- IDF: Log(N/d),
  - A corpus has N documents, and a term appears in d of them
  - A terms most of the documents will have, the IDF value will be low. Otherwise, will be large

TF-IDF Weighting Function

Code:

```
tdm.pos3 = TermDocumentMatrix(corpus.pos[1:3], control = list(weighting = weightTfIdf, stopwords = T, removePunctuation = T
stemming = T))
inspect(tdm.pos3)
```

Output:

	Docs		
Terms	cv000_29590.txt	cv001_18431.txt	cv002_15918.txt
broderick	0	0.02264232	0.00000000
elect	0	0.04528464	0.00000000
fox	0	0.00000000	0.02264232
manipul	0	0.00000000	0.02264232
rushmor	0	0.02717079	0.00000000
ryan	0	0.00000000	0.02264232
school	0	0.02264232	0.00000000
shop	0	0.00000000	0.02264232
student	0	0.02264232	0.00000000
witherspoon	0	0.02264232	0.00000000

Classification

Preprocessing:

```
source = DirSource( "work_path",recursive=T)
corpus = Corpus(source)
dtm = DocumentTermMatrix(corpus,control = list(weighting = weightTfIdf,
                                stopwords = T, removePunctuation = T,
                                stemming = T))
x.df = as.data.frame(as.matrix(dtm))
dim(x.df)
x.df$class_label = c(rep(0,1000),rep(1,1000))
index = sample(1:1000,300)

train = x.df[-c(index,index+1000),]
test = x.df[c(index,index+1000),]
table(train$class_label)
s = findFreqTerms(dtm,4)#the lower bound value of IDF is 4
s = c(s, "class_label")
```

Classification

Logistic Regression:

```
model.glm = glm(class_label ~ .,
train[,s],family=' binomial' )
coef(model.glm)
```

Output:

```
> coef(model.glm)
  (Intercept)           act        action         actor        actual
-8.290933e-02 -1.766547e+01 -9.532001e+00 -2.175362e+01 -2.635650e+01
        alien          also      although      american          anim
 3.597223e+00  1.112705e+02  2.490209e+01  3.489623e+01  3.763168e+00
        anoth       audienc          back           bad         becom
-2.417609e+01 -2.421298e+01  8.131895e+01 -1.569754e+02 -3.976716e+01
         best        better           big       brother           can
 3.345572e+01 -4.905861e+01 -3.733192e+01 -1.236567e+01 -5.080901e+01
         cast       charact          come        comedi           day
 8.234391e-03 -6.839820e+00 -3.756932e+01 -8.845282e+00  1.613605e+01
       direct      director         doesn           don        effect
 2.508545e+01 -2.254644e+01 -1.640883e+01 -2.382740e+01  4.716120e+00
          end        enough          even          everi        famili
 7.671934e+00 -2.801125e+01 -9.795454e+01 -7.182451e+00  2.156777e+01
         feel          film         final          find         first
 3.603780e+01 -2.603441e+01  3.604573e+01  2.805977e+01  3.070028e+01
       friend         funni           get          girl         give
-2.508410e+00 -2.690193e+01  3.131949e+01 -1.175206e+01  2.247310e+00
         good         great           guy        happen         hard
 3.372868e+01  1.287330e+02  3.185141e+01  5.776861e-01 -5.528525e+01
```

Classification

Verification:

```
p = ifelse(predict(model.glm, newdata = test) < 0,0,1)
table(p , test$class_label)
```

Output:

```
p      0   1
  0  229  94
  1   71 206
```

Classification

Support Vector Machine:

```
gamma_list <- 2 ^ seq(-20, 20, 4)
cost_list <- 2 ^ seq(-10, 10, 4)
model.svm = tune.svm(class_label ~., data = train[,s],kernel = "radial",gamma = gamma_list, cost=cost_list)
model.svm
```

Output:

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
      gamma cost
 0.00390625 0.25

- best performance: 0.1885059
```

Classification

Support Vector Machine:

```
model_svm = svm(class_label ~., data = train[,s],kernel = "radial",gamma = 0.00390625,cost = 0.25)
model_svm
```

Output:

```
Call:
svm(formula = class_label ~ ., data = train[, s], kernel = "radial",
    gamma = 0.00390625, cost = 0.25)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  0.25
      gamma:  0.00390625
    epsilon:  0.1


Number of Support Vectors:  1332
```

Classification

Support Vector Machine:

```
p.svm = ifelse(predict(model_svm,newdata = test) < 0.5,0,1)
table(p.svm , test$class_label)
```

Output:

```
p.svm   0   1
    0 237  92
    1  63 208
```

Summary

R packages: tm

- load the text dataset
- Remove the noise from the data

TF and TF-IDF

- Frequency of occurence of the term in the document
- The importance of terms based on how infrequently the term occurs in the corpus

Frequency analysis

Visualization

Clustering

Classification

Homework 8 (submitted to e3.nycu.edu.tw before Nov 29, 2023)

Basic parts:

- Find datasets and conduct text mining with data preprocessing and frequency analysis.

Advanced parts:

- Perform more analysis, like more classification and/or other analysis, with your text data.

Possible sources of open datasets:

- UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets.php)
- Kaggle Datasets (https://www.kaggle.com/datasets)
- World Health Organization Datasets (https://www.who.int/)