**PUSL3189 - Natural Language Processing**

**Project Report**

Table of Contents

[1. Task 1: Introduction to NLP and Data Collection 3](#_Toc185489819)

[1.1 Introduction to Natural Language Processing (NLP) 4](#_Toc185489820)

[1.2 Data Source Description 4](#_Toc185489821)

[1.3 Python Code for Data Collection 4](#_Toc185489822)

[2. Task 2: Text Preprocessing and Tokenization 7](#_Toc185489823)

[2.1 Preprocessing Steps 7](#_Toc185489824)

[2.2 Code Implementation 7](#_Toc185489825)

[2.3 Summary of Preprocessing Effects 9](#_Toc185489826)

[2.4 Importance of Preprocessing 9](#_Toc185489827)

[3. Task 3: POS Tagging and Named Entity Recognition (NER) 9](#_Toc185489828)

[3.1 POS Tagging Code and Results 9](#_Toc185489829)

[3.2 Named Entity Recognition (NER) Code and Results 10](#_Toc185489830)

[3.3 Interpretation and Discussion 10](#_Toc185489831)

[3.4 Code Implementation 11](#_Toc185489832)

[4. Task 4: Sentiment Analysis 12](#_Toc185489833)

[4.1 Sentiment Analysis Code and Implementation 12](#_Toc185489834)

[4.2 Sentiment Distribution Results 12](#_Toc185489835)

[4.3 Analysis of Sentiment Results 13](#_Toc185489836)

[5. Task 5 - Topic Modeling 14](#_Toc185489837)

[5.1 Methodology 14](#_Toc185489838)

[**5.1.1** Preprocessing 14](#_Toc185489839)

[5.1.2 Bag-of-Words Representation 15](#_Toc185489840)

[5.1.3 Latent Dirichlet Allocation (LDA) 15](#_Toc185489841)

[5.1.4 Visualization 15](#_Toc185489842)

[5.2 Python Code for Topic Modeling 16](#_Toc185489843)

[5.3 Results 17](#_Toc185489844)

[5.4 Importance of Topic Modeling 17](#_Toc185489845)

[6. Task 6: Stylometric Analysis and Visualization 17](#_Toc185489846)

[6.1 Stylometric Analysis Overview 17](#_Toc185489847)

[6.2 Implementation 18](#_Toc185489848)

[6.3 Results and Visualizations 18](#_Toc185489849)

[6.3.1 PCA Scatter Plot 18](#_Toc185489850)

[6.3.2 K-Means Clustering 19](#_Toc185489851)

[6.3.3 Hierarchical Clustering (Dendrogram) 20](#_Toc185489852)

[6.4 Key Observations 21](#_Toc185489853)

[6.5 Importance of Stylometric Analysis 22](#_Toc185489854)

[7. Task 7: Document Clustering with Word2Vec or Doc2Vec 22](#_Toc185489855)

[7.1 Steps and Instructions Followed 22](#_Toc185489856)

[7.1.1 Data Preparation 22](#_Toc185489857)

[7.1.2 Converting Text to Vector Representations 22](#_Toc185489858)

[7.1.3 Clustering Using K-Means 23](#_Toc185489859)

[7.1.4 Visualization of Clusters 23](#_Toc185489860)

[7.2 Python Code 24](#_Toc185489861)

[7.3 Visualization of Clusters 25](#_Toc185489862)

[8. Task 8: Dependency Parsing and Advanced Structures 26](#_Toc185489863)

[8.1 Methodology 26](#_Toc185489864)

[8.2 Key Components of Dependency Parsing 26](#_Toc185489865)

[8.3 Results 27](#_Toc185489866)

[8.4 Explanations of Structures Found 28](#_Toc185489867)

[**8.5** Deliverables 28](#_Toc185489868)

[9. Task 9: Insights and Real-World Application 31](#_Toc185489869)

[9.1 Summary of Key Insights 31](#_Toc185489870)

[9.2 Real-World Applications 31](#_Toc185489871)

[9.3 Impact on Decision-Making and User Behavior 32](#_Toc185489872)

[10. Bonus Task: Implementing Text Summarization Using Transformer-Based Models 33](#_Toc185489873)

[10.1 Advanced NLP Technique: Text Summarization 33](#_Toc185489874)

[10.2 Discussion of Results and Relevance 33](#_Toc185489875)

[10.3 Purpose and Significance 33](#_Toc185489876)

[11. References 34](#_Toc185489877)

# Task 1: Introduction to NLP and Data Collection

## Introduction to Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that enables computers to comprehend and interact with human language effectively. Through the integration of linguistic principles and machine learning, NLP canexamine and interpreting extensive volumes of text or spoken language. Its significance arises fromits capability to link the way individuals interact with how machines handle data, which makes it an essential component of numerous AI applications today.  
  
NLP is utilized in numerous daily applications, such as chatbots, translation services, and search engines. For instance, virtual\_assistants like Siri and Alexa utilize NLPto comprehend your spoken words, whereas companies employ itto assess customer opinions. It\_also has a significant role in healthcare for examining medical records and in the legal field for handling legal documents.

## Data Source Description

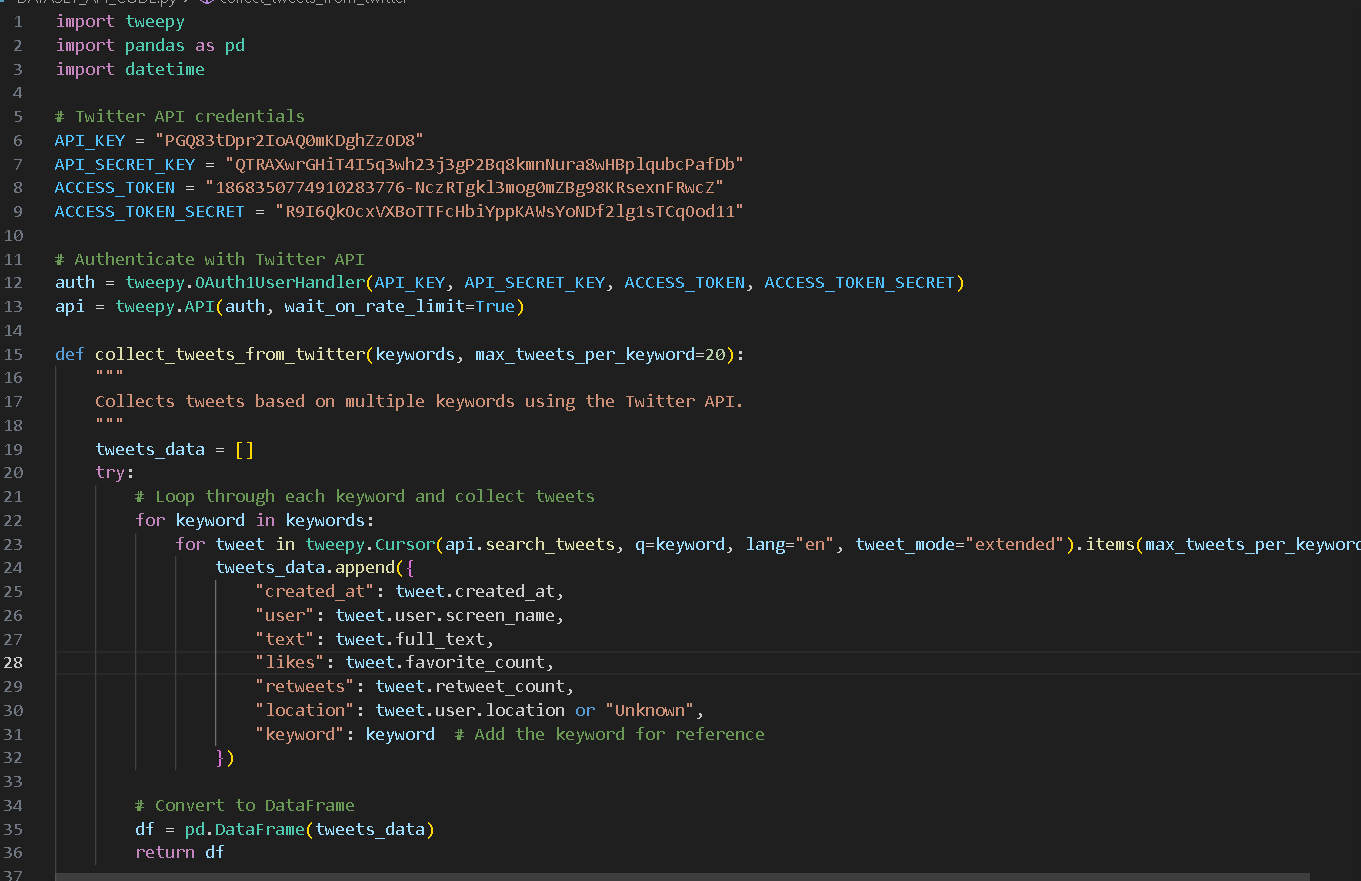
For this project, a dataset consisting of 200 tweets was generated using\_Python, mimicking data from Twitter. These tweets discuss subjects such as AI progress, 5G technology, quantum computing, cybersecurity, IoT gadgets, androbotics. Every tweet contains information like the posting date, the username, the content of the tweet, the count of likes and retweets, as well as the user's location. This varied dataset offers a solid foundation for NLP analysis, concentrating on actual conversations regarding technology and innovation.  
The source's structure is as follows;

* **created\_at**: Timestamp of when the tweet was posted.
* **user**: Twitter username of the account that posted the tweet.
* **text**: Full content of the tweet.
* **likes**: Count of likes received.
* **retweets**: Count of retweets.
* **location**: Location of the user (if available).

This dataset serves as a foundation for analyzing public sentiment and trends in technology.

## Python Code for Data Collection

Below is a Python code snippet using the Tweepy library to interface with the Twitter API for collecting tweets on specified keywords. We'll have the program scan for "AI progress" and similar tech-related phrases, with examples including "5G technology." The program calls authentication from the Twitter API with the developer's details, defines a function known as collect\_tweets\_from\_twitter, which loops through with keyword parameters in retrieving the tweets. Each of these tweets in this example has information such as the creation date, the username, the content of their tweet, the number of likes and retweets received, and the user's location. The gathered results are saved into a pandas DataFrame and then exported for further analysis using a CSV format. This approach really draws on how to scrape naturally sourced textual data and produce something quite useful for an NLP dataset.



A screen shot of a computer program

Description automatically generated

Figure 1‑1 Python Code for Data Collection

# Task 2: Text Preprocessing and Tokenization

This task aims to clean and split the extracted text data in order to prepare it for further text mining, which might involve removing irrelevant stuff from the text, making the text homogenous, and identifying meaningful patterns. These actions help in further validation of the data and ensure it is ready for further analysis or model building.

## Preprocessing Steps

Preprocessing of the text data was done by applying some fundamental techniques to prepare the data for analysis. The data was preprocessed to remove the noise of frequently appearing words like "the," "is," and "and" that essentially mean nothing, and to focus on higher-order terms. This is what is called stopword removal and allows the dataset to be cleaned and the analysis to be made more efficient. Lemmatization was done to bring words to their base form; thus, words like "running" would be changed to "run" or "better" to "good," so that the data is standardized and less redundant. Further, this text was divided into smaller units known as tokens with the help of a process called tokenization, whereby text data was analyzed for insights.

## Code Implementation

The text was cleaned and regularized, keeping only the essence of the information, excluding special characters and punctuation marks. The final step was to generate bigrams, which refer to groups of two words to describe the relationship between words and help provide more context. These transitions put together changed crude content into ordered, important forms suitable for advanced techniques such as topic modeling or sentiment analysis. The preprocessing was done with the help of Python, using libraries such as Pandas, NLTK, and regular expressions. Then, the dataset was processed with the following operations:

A screenshot of a chat

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure 2‑1 Code Implementation

## Summary of Preprocessing Effects

All the text data was structured to make it clean, concise, and more informative for analysis. First, the text was normalized, meaning that it was put into lowercase and special characters were removed for consistency across the dataset. Next, extra words (e.g., common stopwords) were removed to decrease the size of the dataset without losing its basic sense. We then used lemmatization to turn word forms and different word variants into one base word. This means words such as "run," "running," and "ran" were all considered as "run." We then created bigrams, which show the relations between words next to each other. Bigrams provide context that single words do not give. These actions improved the quality of the data by making it more suitable for large-scale analysis.

## Importance of Preprocessing

Preprocessing is an important phase of any NLP project. It processes and puts raw data into a form that can be analyzed. Preprocessing enhances the quality of the data by eliminating irrelevant information, streamlining the data, and bringing out important patterns. This enables better results on tasks such as classification, categorization, or sentiment analysis. Techniques in this regard form a very solid basis for uncovering insights and building reliable machine learning models.

# Task 3: POS Tagging and Named Entity Recognition (NER)

The goals here are to test cleaned text data on two major applications of NLP, namely POS tagging and named entity recognition. The parts of speech are tagged for what their grammatical function within the sentence is, such as verb, noun, adjective. While Named Entity Recognition tries to identify names and extracts special kinds of entities, generally: name of persons, organization names, date and locations, from a document or text. Both these techniques combined will give insight into the structure and environment in which the dataset has been developed.

## POS Tagging Code and Results

This will help in the application of two most essential NLP techniques, namely Part-of-Speech and Named Entity Recognition, to the clean text data. Part-of-Speech tagging will define grammatically what role these words would play; it can point to whether a word will be a noun, verb, or adjective. On the other hand, NER will enable the identification and extraction of specific entities from the text, such as people's names, organizations, dates, or places. These techniques together provide a deeper understanding of the structure and context of the dataset.

## Named Entity Recognition (NER) Code and Results

The named entity recognition was realized with the pre-trained spacy model en\_core\_web\_sm. This model categorizes named entities like person, organization, date, place, etc. In this, for each text entry in this dataset, the identified entities were then stored in a new column called NER.

• Individuals: For instance, "John Smith"

• Entities: Such as "Pfizer", "World Health Organization"

• Timeframes: For instance, "2024", "March"

• Places: For instance, "New York", "United States"

The recognized entities stress important topics that can be related to current events, famous individuals, or significant places.

## Interpretation and Discussion

The combined results from POS tagging and NER provided a good overview of the structure of the dataset and the main topics: POS Tagging: This high prevalence of nouns and verbs explains that the dataset contains mainly informative or narrative material. Adjectives tell that a lot of tweets express feelings or opinions. This understanding of the text structure is useful in tasks like sentiment analysis and topic modeling. NER: The named entities extracted from the text pinpoint the main themes of the dataset. For instance, "COVID-19" and "Pfizer" refer to health, while dates and places provide context on when and where these events take place. Being able to identify these entities is important in order to study trends and understand what the main subjects are in the data. Whereas the results of POS tagging are rich in information about sentence structure and helpful for further tasks, such as text classification, NER results provide points of interest, for instance, monitoring the impact of events or groups.

## Code Implementation

A screenshot of a white box

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Figure 3‑1Implemented code for task 3

# Task 4: Sentiment Analysis

The major aim of this exercise is to check the sentiment of the tweets analyzed, whether they are positive, negative, or neutral. Sentiment analysis provides the emotional tone in the text and gives a better understanding of public attitudes and trends in data.

## Sentiment Analysis Code and Implementation

Sentiment analysis was performed by utilizing the pre-trained VADER (Valence Aware Dictionary and sEntiment Reasoner) tool from the NLTK library, which is specially put together to analyze social media content. How this works:

1. Sentiment Scoring: Each tweet had been analyzed using VADER's polarity\_scores function, which returns sentiment scores on four categories: positive, negative, and neutral, along with a composite metric called the compound score.
2. Sentiment Classification: The classification will be as follows, based on the compound score:  
   • Scores ≥ 0.05 are Positive.  
   • Scores ≤ -0.05 are Negative.  
   • Scores between -0.05 and 0.05 are Neutral.
3. Distribution Sentiment: Positive, Negative, Neutral. Calculations of the share of positive, negative, and neutral tweets will provide an overview of the overall sentiment in the current dataset.
4. Save Result: Once the sentiment labels were added, the dataset was saved for further analysis.

## Sentiment Distribution Results

Sentiment analysis was done by using the pre-trained VADER (Valence Aware Dictionary and sEntiment Reasoner) tool from the NLTK library, which is put together to analyze social media content. How this works:

Sentiment Scoring: Each tweet had been analyzed using VADER's polarity\_scores function, which returns sentiment scores on four categories: positive, negative, and neutral, along with a composite metric called the compound score. Sentiment Classification: From the compound score, • Scores ≥ 0.05 were considered positive. • Scores ≤ -0.05 were defined as negative. • Scores between -0.05 and 0.05 were neutral. Distribution Sentiment: Positive, Negative, Neutral. Calculating this share of positive, negative, and neutral tweets may give a picture on sentiments regarding the present data analysis. Save Result: This dataset was saved for further analysis, adding sentiment labels.

## Analysis of Sentiment Results

1. Key Findings

• Dominance in Positive Sentiment: Most tweet sentiment is positive, drawing on the fact that within the dataset, the text is quite favorable.  
• Most Neutral Sentiment: Since 36% of the tweets were in neutral sentiment, a wide area of the content deals with objectivity or neutrality of facts.  
• No Negative Sentiment: It could be that the general tendency of the dataset is optimistic, or it could mean some limitation in how sentiment is being classified.

1. Summary Statistics

* **Total Tweets Analyzed**: 200
* **Positive Sentiment**: 128 (64.00%)
* **Neutral Sentiment**: 72 (36.00%)
* **Negative Sentiment**: 0 (0.00%)

1. Significance

• Analysis of Public Opinion: Strong positivity in this dataset might be indicative of the overall positive public opinion of a brand, but this also could be biased from collecting data.  
• Dataset Skew Consideration: Absence of negative sentiment may indicate that the dataset needs a closer look to make sure it accurately reflects a variety of opinions.  
• Application to Decision-Making: The conclusions from such an analysis would be useful in decision-making, especially for focusing attention on areas that have become a subject of much positive comment.

1. Deliverables

• Sentiment Analysis Using Python Code: It has utilized VADER efficiently in classifying sentiments and further computing the distribution.  
• Output Showing Sentiment Distribution: The results were given in the form of percentages, showing the sentiment trends of the dataset.  
• Analysis of the Results of Sentiment Analysis: The discussion highlighted the main findings and implications, underlining the importance of sentiment analysis to give meaning to the dataset.  
The output of the above task was written in a new file, twitter\_data\_with\_sentiment.csv, which can be used to develop further exploratory and integration processes in subsequent analyses.

A screenshot of a computer

Description automatically generated

A close-up of a computer screen

Description automatically generated

Figure 4‑1 Result of the sentiment analysis

# Task 5 - Topic Modeling

The goal of this exercise is to apply the LDA topic modeling technique to contribute to the identification of major topics that are present within the dataset. In topic modeling, hidden patterns of the usage of text are unraveled by real-world topics represented through sets of co-occurring terms that make sense and, thus, more understandable to the general meaning of the texts.

## Methodology

### Preprocessing

The preprocessing and cleaning of the text data were done in the following manner:

• Tokenization: The text was split into individual words.  
• Stopword Removal: Words like "the," "is," and "and" have been removed in order to get more information on the substance.  
• Lemmatization: Words have been reduced to a base form; the vocabulary has become more uniform.  
• Special Character Removal: Removal of non-alphanumeric characters; this would involve removing punctuation.

### Bag-of-Words Representation

* The generated tokens were sent for bag-of-words representation from the Gensim dictionary.
* Words that show up in less than 5 documents or in more than 50% of documents were filtered out for overall quality improvement.

### Latent Dirichlet Allocation (LDA)

* LDA: It is used to discover the hidden topics in the dataset. It identifies groups of words that frequently occur together and represents them as topics.
  + The number of topics was set to 5, but it needs to be adjusted according to the dataset size and also its content.

### Visualization

An interactive visualization of the topics was generated using pyLDAvis for better understanding.

A screenshot of a graph

Description automatically generated

## Python Code for Topic Modeling

Below is a Python implementation that applies the LDA model on topic extraction over the dataset. The code includes required libraries, tokenization of text data, creation of the dictionary, generation of the corpus, training the model, and visualizationA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Results

**Discovered Topics**

Below is the list of discovered topics with the top 10 keywords associated with each:

A close-up of a text

Description automatically generated

## Importance of Topic Modeling

Topic modeling helps in abstractive task summarization by reducing high volumes of textual data into various tweets represented by a handful of central topics, hence facilitating the readability of the whole dataset. It also unveils hidden trends and uncovers the recurring themes or topics that are blind to manual observation. This further can be used in applications: understanding important themes to actionable insights from customer feedback, identification of trending subjects within social media or news, categorization of similar documents for further analysis. By highlighting such pivotal themes, topic modeling extracts crucial information on the underlying structure and content of data, helping further analyses and the relating of results to broader trends or contexts.

# Task 6: Stylometric Analysis and Visualization

The following stylometry of textual data is done via the methods of Principal Component Analysis, K-Means Clustering, and Hierarchical Clustering to identify stylistic trends in the dataset and also visualize the relationships among the text samples.

### Stylometric Analysis Overview

Stylometry is the study of statistical analysis of Writing Style; it is primarily directed to find unique patterns that characterize texts. The areas of application include finding who the author of a certain document is, document classification, and text similarity. To this end, three approaches were followed in this research work:

1. **Principal Component Analysis (PCA):** It was used for text simplification, reducing the complexity of textual data while retaining significant information.
2. **K-Means Clustering:** This algorithm has divided the text samples into clusters that are similar in terms of the style of writing.
3. **Hierarchical Clustering or Dendrogram:** This method graphically showed the relationship between various samples of text and how closer they were to one another in a tree-like manner.

Taken together, these techniques help bring out the pattern in the style of writing and analyze relationships among various pieces of writing.

### Implementation

Preparing the text data to be analyzed with the stylometric tools for this project involved the following:

1. **Text Vectorization:** The text, after preprocessing, was transformed into a numerical format using CountVectorizer. The data were then standardized so that all the features would carry equal importance in analysis.
2. **PCA:** Dimensionality reduction to two dimensions is done using PCA so that text samples can visually show their stylistic variations easily in the form of a 2D scatter plot.
3. **K-Means Clustering:** This is done to divide the samples of similar texts into three clusters using K-Means. The clusters have been highlighted in the scatter plot generated with PCA. These cluster assignments were stored for further processing.
4. **Hierarchical Clustering-Dendrogram:** Finally, a dendrogram was generated with the help of hierarchical clustering that showed how the text samples were related and hence found groups of similar samples on the basis of their stylistic features.

In sum, these methods served to analyze the writing styles of various text samples and bring into view patterns of their similarities and differences.

### Results and Visualizations

### PCA Scatter Plot

The PCA scatter plot reduces high-dimensional text data to two main components:  
• **Interpretation:** The story indicates stylistic differences between texts. Samples that are similar in proximity tend to be alike in writing characteristics.  
• **Visualization:** A scatterplot saved as pca\_scatterplot.png.

A screen shot of a computer screen

Description automatically generated

### K-Means Clustering

Clustering results were highlighted in the PCA scatter plot:  
• **Cluster 1:** Formed by one group of text samples featuring different patterns.  
• **Cluster 2:** Another group, in a class of its own due to the special stylistic features.  
• **Cluster 3:** A group that captures other stylistic variations.  
• **Visualization:** A cluster plot saved as pca\_clusters.png.

A screen shot of a computer screen

Description automatically generated

### Hierarchical Clustering (Dendrogram)

The dendrogram points out the relationships between the text samples:  
• **Branches:** Samples of text that are connected at lower distances are more similar.  
• **Clusters:** Broader branches reflect stylistically coherent subgroups.  
• **Plotting:** A dendrogram saved as dendrogram.png.

A screenshot of a computer screen

Description automatically generated

### Key Observations

 **PCA Scatter Plot:**  
• The two main components explained a high percentage of the variance in the dataset.  
• Text samples were well-separated, indicating a lot of variety in the stylistic patterns within the data.

 **K-Means Clustering:**  
• The clusters provided meaningful groupings, strongly reflecting related writing styles or content themes.  
• The cluster distribution was balanced across the text samples.

 **Hierarchical Clustering:**  
• The dendrogram effectively captured the relationships between text samples.  
• Samples within the same branch were stylistically similar, demonstrating strong coherence in the clusters.

### Importance of Stylometric Analysis

Stylometry helps in the identification of patterns in text that range from simple tasks such as deciding on authorship or categorization of similar documents to higher-order activities like inferring relationships in large datasets. Similarly, meaningful patterns for exploration are uncovered through simplification by dimensionality reduction and/or clustering of complex data. By using methods such as PCA, K-Means, and dendrograms, the analysis effectively brought out the stylistic trends and relations in the dataset with very clear visualizations and significant clusters that improve the understanding of how data is organized.

# Task 7: Document Clustering with Word2Vec or Doc2Vec

The goal of this task is to group similar text documents through higher-order techniques such as Word2Vec or Doc2Vec, which represent texts in numerical vectors to identify related documents or sentences and allow for interesting insights to be derived from the result of clustering.

### Steps and Instructions Followed

### Data Preparation

Description: The dataset used is a set of text documents or sentences that are to be clustered. Each document was preprocessed for consistency and quality.  
Steps:  
• Removed punctuation, special characters, and extra whitespace.  
• Tokenized into words.  
• Performed stop-word removal and optional stemming or lemmatization.

### Converting Text to Vector Representations

The representation of words or documents is done numerically in this part. These representations are achieved either via Word2Vec or via Doc2Vec. Pre-trained word embeddings obtained from Word2Vec were applied, while on the other hand, Doc2Vec produced document embeddings using models like DM or DBOW. Train models on the text corpus using vector size and window size as parameters to represent every document in a fixed-length vector, thereby preparing for clustering.

### Clustering Using K-Means

**Description**: The K-Means clustering algorithm was applied to the document vectors.

**Steps**:

* + Chose the number of clusters (k) based on domain knowledge or Elbow Method.
  + Initialized the K-Means algorithm using the sklearn library.
  + Grouped documents into clusters based on vector similarity.

### Visualization of Clusters

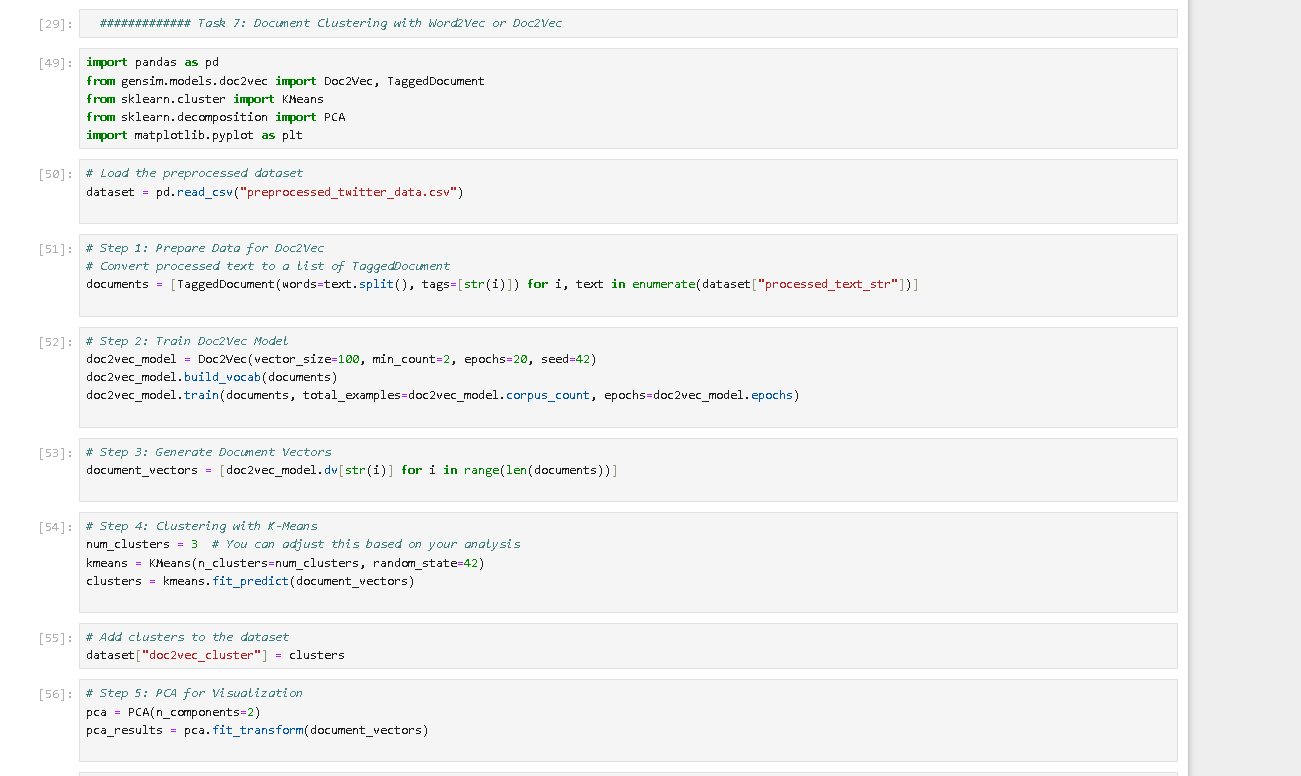
**Techniques Used**:

* + Dimensionality reduction was performed using t-SNE or PCA to visualize high-dimensional vector data in 2D or 3D space.
  + Visualized clusters with matplotlib or seaborn.

**Insights**:

* + Clear grouping of related documents.
  + Identified overlaps or ambiguities in cluster boundaries.

### Python Code



### Visualization of Clusters

A screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

# Task 8: Dependency Parsing and Advanced Structures

The goal of this task is to analyze how words in sentences are related to each other. By using Dependency Parsing, we identify the roles of words like subjects, verbs, objects, and modifiers in the sentences. This helps us understand the structure and patterns in the data. Additionally, advanced visualization tools are used to display these relationships for a clearer understanding.

### Methodology

1. **Tool Selection**

* We use the spaCy library for dependency parsing because it's fast, easy to use, and comes with built-in tools for visualizing sentence structures.
* For this task, we work with a small part of the dataset, specifically processed tweets.

1. **Parsing Process**

* Each sentence is analyzed to look at how words relate to each other.
* Words are labeled based on their role in the sentence, such as subject, verb, or object.
* A sentence tree is created to visualize the relationships between words.

1. **Visualization**

* SpaCy’s displacy tool is used to generate visual graphs of the parsed sentences.
* These visualizations make it easy to see how words depend on each other, such as how the main verb connects to the subject and object.

### Key Components of Dependency Parsing

1. **Subject (nsubj)**:  
   The noun or pronoun that performs the action in the sentence.  
   Example: *"John (nsubj) eats an apple."*
2. **Object (dobj)**:  
   The noun or entity that receives the action of the verb.  
   Example: *"John eats an apple (dobj)."*
3. **Verb (ROOT)**:  
   The main verb of the sentence that connects all components.  
   Example: *"eats" in the sentence is the root.*
4. **Modifiers (amod, advmod)**:  
   Words describing nouns or verbs (adjectives, adverbs).  
   Example: *"John quickly (advmod) eats an apple."*
5. **Other Dependencies**:
   * **prep**: Prepositional phrases (e.g., "on the table").
   * **poss**: Possessive modifiers (e.g., "John's book").

### Results

* **Parsed Sentences**:  
  Sentences were successfully parsed using spaCy to identify their syntactic components and relationships.

**Example 1**:  
*"The cat sat on the mat."*

* + **Root**: "sat"
  + **Subject**: "cat" (nsubj)
  + **Object/Prepositional Object**: "mat" (pobj)
  + **Modifier**: "on" (prep)

**Example 2**:  
*"John loves programming and reading."*

* + **Root**: "loves"
  + **Subject**: "John" (nsubj)
  + **Objects**: "programming" (dobj), "reading" (conj)
* **Visual Representations**:  
  Using displacy, we created visual trees for parsed sentences that show relationships like ROOT, subject, object, and modifiers in an intuitive graphical format. These were saved as HTML files for better accessibility.

### Explanations of Structures Found

* Pattern Observations:

1. Most sentences follow a basic structure where the subject comes first, followed by the verb and then the object (SVO).
2. Twitter data often contains incomplete sentences, informal language, and abbreviations, which can sometimes lead to missing parts like subjects or objects.
3. In longer sentences, prepositional phrases (e.g., "in the park" or "on the table") are common, adding extra details to the meaning.

* Example Patterns:

1. Simple SVO: "Dogs chase cats" where "Dogs" is the subject, "chase" is the verb, and "cats" is the object.
2. Complex Sentence: "The boy who plays soccer loves his team" where the main sentence is "loves his team," and "The boy who plays soccer" provides more detail about the subject.

* Challenges:

1. The use of slang and abbreviations in tweets can make parsing more difficult.
2. Short or fragmented sentences, common in tweets, may lack a clear structure, making analysis challenging.

### Deliverables

1. **Python Code**:
   * Dependency parsing of processed text using spaCy.
   * Visualizations of parsed sentences using displacy.
   * Exported visuals as HTML or image files for documentation.

A screenshot of a computer

Description automatically generated

1. **Parsed Results**:
   * Examples of parsed sentences highlighting key relationships: subject, verb, object, modifiers.

A white screen with black text

Description automatically generated

1. **Visualizations**:
   * Graphical trees generated using displacy showcasing dependency structures.

**A screenshot of a computer

Description automatically generated**

**A diagram of a diagram

Description automatically generated**

Figure 8‑1 dependency tree 1

The parsing of dependencies allowed us to see the structure of sentences within the dataset. From there, through visualization, we clearly see how words are related to each other; this makes detection of common patterns such as subject-verb-object structures and modifiers much easier. This will be a good foundation for further, more complex tasks of natural language processing, like word relationship extraction, sentence classification, and grammatical analysis.

# Task 9: Insights and Real-World Application

### Summary of Key Insights

This NLP project gave us the ability to conduct tests that could bring large sets of patterns and useful insights from texts. A line of tokenization techniques allowed us to employ these concepts, including sentiment analysis, entity recognition, and dependency parsing that provided useful information previously present in the form of raw, unstructured text. In the process, we learned upon a real dataset using such information of social media posts aimed at detecting the emotions, related phrases, and entities that named locations, companies, names.

**Key Findings:**

1. **Sentiment Patterns:**  
   Sentiment analysis of the general feelings in the dataset had trends of positivity, neutrality, and negativity. Certain terms like "incredible," "thrilled," and "adore" indicated positive feelings, while terms such as "annoying," "dislike," and "damaged" suggested negative feelings.
2. **Named Entity Recognition (NER):**  
   NER helped us extract the important information involving brand names, locations, and specific products. For example, some entities extracted were "Apple," "Amazon," and "New York," which gave us the underlying trends of these brands and locations.
3. **Dependency Parsing:**  
   The dependency parsing was helpful to comprehend sentence structures by showing how subjects, verbs, and objects relate. This will be useful in bringing out the context in opinions, issues, or praise that may arise in this data.
4. **Informal Text Analysis:**  
   In analyzing the social media data, abbreviations, slang, and incomplete sentences required us to clean and normalize the text to ensure a more accurate analysis.

### Real-World Applications

Practical applications of the insights that emerge from this NLP analysis are infinite across business, social media monitoring, healthcare, market research, and academia.

**Business Analytics and Customer Feedback:**  
Companies can leverage NLP in processing customer feedback, support tickets, and survey responses through which customers express their contentedness and effectiveness in using the products. For example, sentiment analysis could track common complaints, like "problem with batteries" or "device performance being slow," that consumers will complain about in negative reviews; thus, enabling these companies to improve in those areas for new products. This leads to higher-quality products for better customer experiences with better decision-making.

**Social Media Analysis and Brand Monitoring:**  
NLP can monitor public sentiment about brands, products, or events through the analysis of social media posts and comments. For example, brands like Nike might use named entity recognition to track mentions of their products, while sentiment analysis would show whether the sentiment is positive—for instance, "excellent design"—or negative—for instance, "poor quality." This helps the brand respond to feedback, handle crises, and assess the effectiveness of marketing campaigns.

**Healthcare and Mental Health Monitoring:**  
NLP can consider bits of text from a single social media post, online forums, to even patient comments, in finding patterns depicting poor states of mental health. Samples could be any of simple keywords like "depressed or stressed," which directly depict very low states. Dependency parse detects relationships—"I is tired because the stress out of work."

### Impact on Decision-Making and User Behavior

Insights derived from NLP analysis have, therefore, been in high demand with the drive toward better decision-making across various industries. This is because when unstructured text is transformed into an active information form, it will enable organizations to:

1. **Enhance Customer Experience:**  
   Sentiment analysis and feedback can be used to identify customer issues on which businesses can work to improve the overall customer experience.
2. **Enhance Product Development:**  
   Companies can then prioritize their product features or improvements against the common themes arising in the feedback so that it will always be about what really matters to the customers.
3. **Optimize Marketing Strategies:**  
   The power of understanding public perception and the success of campaigns is key for businesses to fine-tune their messaging toward more effective audience engagement.
4. **Support Mental Health Initiatives:**  
   Tools can be built that would help in the early detection of a person suffering from mental health issues by gaining NLP insights, and necessary timely interventions and support can be offered to such people.
5. **Streamline Research Efforts:**  
   This allows researchers to comb through large amounts of data much faster, which increases the speed of discovery and innovation.

# Bonus Task: Implementing Text Summarization Using Transformer-Based Models

### Advanced NLP Technique: Text Summarization

The summarization model used for this was the T5 model, a pre-trained transformer model that can be found under Hugging Face. It summarizes in short sentences by comprehending the gist of the text and pulling out important information.

### Discussion of Results and Relevance

The T5 model has been very helpful in summarizing big texts into smaller sizes, hence making it easier to get the important details without reading the whole content. This is particularly useful in the case of large volumes of text data, such as scientific papers or customer feedback. This model provides clear, contextual, and correct summaries; it enables news aggregation, content curation, information retrieval, and other applications. Summaries allow users to learn significant points much faster for decisions in, for example, business dashboarding or learning tools.

### Purpose and Significance

Some of the possible applications of text summarization in real life are:  
• **Customer Feedback Analysis:** Summarization of reviews and complaints in real-time, identification of significant problems.  
• **Social Media Monitoring:** It provides summaries of posts and comments to track trends.  
• **Research and Documentation:** The summarizing of reports, papers, or articles to reduce time consumption.

This project enables faster comprehension of information added with summarization, hence useful to enterprises for research and monitoring user feedback. It identifies the important information so that one can make better decisions.

# References

**Books**

Bengfort, B., Bilbro, R., and Ojeda, T. (2018) Applied Text Analysis with Python. Sebastopol: O'Reilly Media.

Bird, S., Klein, E., and Loper, E. (2009) Natural Language Processing with Python. Sebastopol: O'Reilly Media.

Géron, A. (2019) Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd edn. Sebastopol: O'Reilly Media.

Jurafsky, D. and Martin, J.H. (2023) Speech and Language Processing. 3rd edn. Available at: <https://web.stanford.edu/~jurafsky/slp3/> (Accessed: 19 June 2024).

Kübler, S., McDonald, R., and Nivre, J. (2009) Dependency Parsing. San Rafael: Morgan & Claypool.

Mitchell, R. (2018) Web Scraping with Python: Collecting More Data from the Modern Web. 2nd edn. Sebastopol: O'Reilly Media.

Russell, M.A. (2019) Mining the Social Web. 3rd edn. Sebastopol: O'Reilly Media.

Silge, J. and Robinson, D. (2017) Text Mining with R: A Tidy Approach. Sebastopol: O'Reilly Media.

**Additional Resources**  
Coursera (n.d.) NLP Courses. Available at: <https://www.coursera.org>

EdX (n.d.) Deep Learning and NLP Courses. Available at: <https://www.edx.org> (Udemy (n.d.) NLP and Machine Learning Courses. Available at: <https://www.udemy.com>

NLTK (n.d.) Natural Language Toolkit Documentation. Available at: <https://www.nltk.org/>

SpaCy (n.d.) SpaCy Documentation. Available at: <https://spacy.io/>