|  |
| --- |
| Name: Pathirathnage Attygala |
| Student Reference Number: 10899179 |



|  |  |  |
| --- | --- | --- |
| Module Code: PUSL3189 | | Module Name: Natural Language Processing |
| Coursework Title: Individual Coursework | | |
| Deadline Date: 02nd of January 2025 | Member of staff responsible for coursework: Nethmi Weerasingha | |
| Program: Bsc (Hons) in Data Science | | |
| Please note that University Academic Regulations are available under Rules and Regulations on the University website [www.plymouth.ac.uk/studenthandbook](http://www.plymouth.ac.uk/studenthandbook). | | |
| Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.  ***We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.***  Signed on behalf of the group: | | |
| Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***  Signed: | | |
| Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.  I \*have used/not used translation software.  If used, please state name of software………………………………………………………………… | | |
| **Overall mark \_\_\_\_\_% Assessors Initials \_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_** | | |

\*Please delete as appropriateSci/ps/d:/students/cwkfrontcover/2013/14

Table of Contents

[Task 1: Introduction to NLP and Data Collection 3](#_Toc185805668)

[1.1. Introduction to NLP 3](#_Toc185805669)

[1.2. Its significance in real-world applications 3](#_Toc185805670)

[1.3. Description of the data source 4](#_Toc185805671)

[Task 2: Text Preprocessing and Tokenization 4](#_Toc185805672)

[2.1. Tokenization 4](#_Toc185805673)

[2.2. Removing Stop Words 5](#_Toc185805674)

[2.2.1. Removing Punctuation 5](#_Toc185805675)

[2.3. Lemmatization and Stemming 5](#_Toc185805676)

[2.4. N-grams 5](#_Toc185805677)

[Task 3: POS Tagging and Named Entity Recognition (NER) 5](#_Toc185805678)

[3.1. Part-of-Speech (POS) Tagging 5](#_Toc185805679)

[3.2. Named Entity Recognition (NER) 6](#_Toc185805680)

[Task 4: Sentiment Analysis 6](#_Toc185805681)

[Task 5: Topic Modeling 7](#_Toc185805682)

[5.1. Creative Technique: TF-IDF Removal for Topic Refinement 7](#_Toc185805683)

[5.1.1. The Process 8](#_Toc185805684)

[5.1.2. Rationale and Benefits 8](#_Toc185805685)

[Task 6: Stylometric Analysis and Visualization 9](#_Toc185805686)

[Task 7: Document Clustering with Word2Vec or Doc2Vec 12](#_Toc185805687)

[Task 8: Dependency Parsing and Advanced Structures 13](#_Toc185805688)

[Task 9: Insights and Real-World Application 13](#_Toc185805689)

[Bonus Task: Implement an Advanced NLP Technique 15](#_Toc185805690)

[Text Summarization Using BART 15](#_Toc185805691)

[References 15](#_Toc185805692)

[Appendix 16](#_Toc185805693)

# Task 1: Introduction to NLP and Data Collection

## Introduction to NLP

Natural Language Processing (NLP) is a branch of artificial intelligence (AI) that bridges the gap between human communication and computer systems. At its core, NLP enables machines to understand, interpret, and generate human language in a way that feels natural. From text to speech and everything in between, NLP powers many of the tools we interact with daily, such as virtual assistants, chatbots, search engines, and translation apps.

The complexity of human language filled with nuances, context, slang, and emotion makes NLP a challenging yet fascinating field. It draws upon techniques from linguistics, computer science, and machine learning to tackle problems like sentiment analysis, machine translation, text summarization, and question answering. By teaching machines to "read" and "write" like humans, NLP helps unlock new ways of automating tasks, enhancing communication, and gaining insights from vast amounts of textual data.

As NLP continues to advance, its applications are becoming more diverse, from personalized recommendations and real-time language translation to medical diagnostics and legal document analysis. With its growing importance in both everyday tools and specialized systems, NLP is not just about processing language but also about fostering better interaction between humans and machines.

## Its significance in real-world applications

Natural Language Processing (NLP) plays a transformative role in the modern world, driving numerous applications that make our lives more efficient, informed, and connected. Its significance lies in its ability to process and analyze vast amounts of human generated language, enabling machines to provide insights, automate tasks and facilitate seamless interaction with technology.

One of the most visible applications of NLP is in customer service, where chatbots and virtual assistants use it to understand and respond to user queries in real time. This not only enhances user experience but also reduces the need for human intervention. Similarly, search engines like Google rely on NLP to understand user intent delivering relevant results even when queries are vague or complex.

In healthcare, NLP helps analyze clinical notes, patient records, and medical literature to assist in diagnosis, treatment planning, and drug discovery. Legal and financial sectors leverage NLP for document summarization, contract analysis and fraud detection, saving time and reducing errors in critical processes.

NLP enables tools like real-time language translation, voice activated assistants (e.g. Siri, Alexa), and predictive text, making technology more accessible and personalized. In social media and marketing, sentiment analysis allows businesses to gauge public opinion, tailor campaigns and even predict trends.

NLP also supports inclusivity and accessibility, for example, by powering speech-to-text and text-to-speech systems for individuals with disabilities. Additionally, its use in content moderation ensures safer online environments by detecting and filtering harmful or inappropriate content.

## Description of the data source

I used the Kaggle API to extract data from Kaggle, a platform that hosts a vast collection of datasets and competitions for data science and machine learning. The API provides seamless access to these resources, allowing users to authenticate using a ‘kaggle.json’ file that contains their API credentials. By configuring the API and ensuring proper permissions, I could explore and download datasets directly into my working environment. In this instance, I downloaded the "Fake and Real News Dataset," which contains labeled articles aimed at supporting research in detecting fake news. The API simplifies the process of accessing high-quality datasets for analysis, experimentation, and developing data-driven solutions.

# Task 2: Text Preprocessing and Tokenization

The text data undergoes a series of preprocessing steps to prepare it for analysis. Initially, tokenization with ‘word\_tokenize’ and ‘sent\_tokenize’ splits the text into individual words and sentences (Webster and Kit, 1992). Subsequently, approximately 40% of the data consisting of common English stop words is removed using ‘nltk.corpus.stopwords’. The process further cleans the text by eliminating punctuation marks with ‘string.punctuation’, ensuring smoother analysis. Both stemming, utilizing NLTK's ‘PorterStemmer’, and lemmatization, powered by spaCy, are then applied to reduce words to their root or base forms, potentially improving accuracy by grouping similar words. Finally, ‘CountVectorizer’ from sklearn is employed to generate bigrams, capturing relationships between adjacent words and adding context to the analysis (Cavnar and Trenkle, 2001). These preprocessing techniques result in a refined dataset that's tailored for effective NLP modeling.

## 2.1. Tokenization

Dividing the text into individual words or sentences (tokens) using functions like ‘word\_tokenize’ and ‘sent\_tokenize’ from the nltk library.

Importance: Breaks down text into manageable units for easier processing and analysis. By using these functions, tokens of the text can be generated, which will be used in further steps like POS tagging and NER.

## 2.2. Removing Stop Words

Eliminating commonly occurring words (e.g., "the," "a," "is") that are usually not significant for analysis. This is achieved using the ‘nltk.corpus.stopwords’ list and filtering out these words from the text.

Importance: Reduces noise and data size, allowing models to focus on meaningful words. It has been observed that 40% of the data was stop words, which highlights how important it is to remove them.

### 2.2.1. Removing Punctuation

Removing punctuation marks (e.g., commas, periods, question marks) using the string module's punctuation attribute.

Importance: Eliminates non-alphanumeric characters that can interfere with text processing and analysis.

## 2.3. Lemmatization and Stemming

Reducing words to their base or root form (e.g., "running" becomes "run"). Lemmatization leverages vocabulary and morphological analysis of words to obtain the base form and is implemented using spaCy. Stemming simply chops off the end of the word to obtain the base form and is implemented using ‘PorterStemmer’.

Importance: Groups of similar words, improving accuracy and efficiency of analysis. This process can help extract the topic from the data or to extract features in a machine learning model.

## 2.4. N-grams

Creating combinations of adjacent words (e.g., "New York," "social media"). The code uses ‘CountVectorizer’ with ‘ngram\_range=(2, 2)’ to generate bigrams.

Importance: Captures word relationships and context, enhancing the understanding of text meaning and can be helpful in improving model performance.

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

## 3.1. Part-of-Speech (POS) Tagging

Interpretation: POS tagging assigns grammatical tags to each word in a sentence, identifying its role (e.g., noun, verb, adjective). The code uses spaCy's nlp object to perform POS tagging, providing detailed information about each token.

Significance: POS tagging is crucial for understanding the structure of sentences and the relationship between words. It is a fundamental step in many NLP tasks, such as text parsing, dependency analysis, and feature extraction for machine learning models.

## 3.2. Named Entity Recognition (NER)

Interpretation: NER identifies and classifies named entities in text, such as people, organizations, locations, dates, and more. The code utilizes spaCy's ‘doc.ents’ attribute to extract named entities and their labels, providing insights into the key elements within the text.

Significance: NER helps extract important information from text and is used in applications like information retrieval, knowledge base creation, and question answering. By automatically recognizing entities, it enables text summarization, content understanding, and search engine optimization.

The code also utilizes ‘displacy.render()’ from spaCy to create visually appealing representations of the POS tags and NER results. By using the style="dep" or style="ent" within ‘displacy.render()’, users can display a dependency parse tree or visualize the named entities in a document, respectively.

# Task 4: Sentiment Analysis

This task focuses on determining the overall sentiment expressed in the text data, typically categorized as positive, negative, or neutral. The code uses the TextBlob library to perform sentiment analysis. Here's a breakdown of the analysis:

**Sentiment Polarity:**

Calculation: TextBlob's ‘sentiment.polarity’ attribute provides a score ranging from -1 (negative) to 1 (positive), indicating the sentiment's intensity.

Interpretation: Higher polarity scores suggest positive sentiment, while lower scores indicate negative sentiment. Scores close to 0 represent neutral sentiment.

**Sentiment Classification:**

Categorization: Based on the polarity scores, the code classifies sentiments into three categories: "Positive," "Negative," or "Neutral" using the ‘classify\_sentiment’ function.

Interpretation: This classification helps understand the overall sentiment expressed in the text, providing a general overview of the data's emotional tone.

**Sentiment Distribution:**

Analysis: The code calculates the distribution of sentiments (positive, negative, and neutral) across the entire dataset using ‘value\_counts’(normalize=True).

Interpretation: This distribution provides insights into the prevalence of each sentiment category within the data. It helps identify overall trends and patterns related to the emotional tone of the text.

# Task 5: Topic Modeling

Topic modeling, specifically using Latent Dirichlet Allocation (LDA), plays a crucial role in uncovering hidden thematic structures within the text data (Jelodar *et al.*, 2019). By automatically grouping documents based on shared topics, it provides valuable insights into the main themes and subjects discussed across the corpus. This allows for a deeper understanding of the content beyond simple keyword analysis, revealing underlying patterns and relationships that might not be immediately apparent. Topic modeling helps organize and summarize large collections of text, making it easier to navigate and interpret vast amounts of information. In this analysis, it assists in identifying prominent themes such as political discourse, international relations, economic matters, and potential social issues, thereby revealing the core areas of focus within the dataset. By providing a thematic framework, topic modeling enhances content analysis, enabling researchers to explore trends, identify clusters of similar documents, and ultimately gain a more comprehensive understanding of the text data's key areas of focus (Kherwa and Bansal, 2020). It serves as a valuable tool for uncovering hidden patterns and generating insightful summaries, making it an essential component of this analysis.

LDA topic modeling to reveal underlying themes within the text data. While the code itself doesn't explicitly label or interpret the topics, by examining the most frequent words associated with each topic using ‘pyLDAvis’, we can make inferences. One prominent topic revolves around political figures and events, featuring words like "Trump," "president," "election," and "republican," suggesting discussions about political discourse and news. Another topic focuses on international relations and conflicts, highlighted by terms such as "Russia," "China," "Syria," and "war," indicating coverage of global affairs. Additionally, a topic related to economic matters emerges, including words like "market," "economy," "trade," and "financial," implying discussions of economic trends and policies. Further analysis with ‘pyLDAvis’ reveals potential topics concerning social issues, healthcare, and legal matters, though these require further exploration and confirmation. While interpretation is subjective and influenced by the dataset and specific LDA implementation, these initial observations provide a glimpse into the diverse themes captured by the discovered topics.

## 5.1. Creative Technique: TF-IDF Removal for Topic Refinement

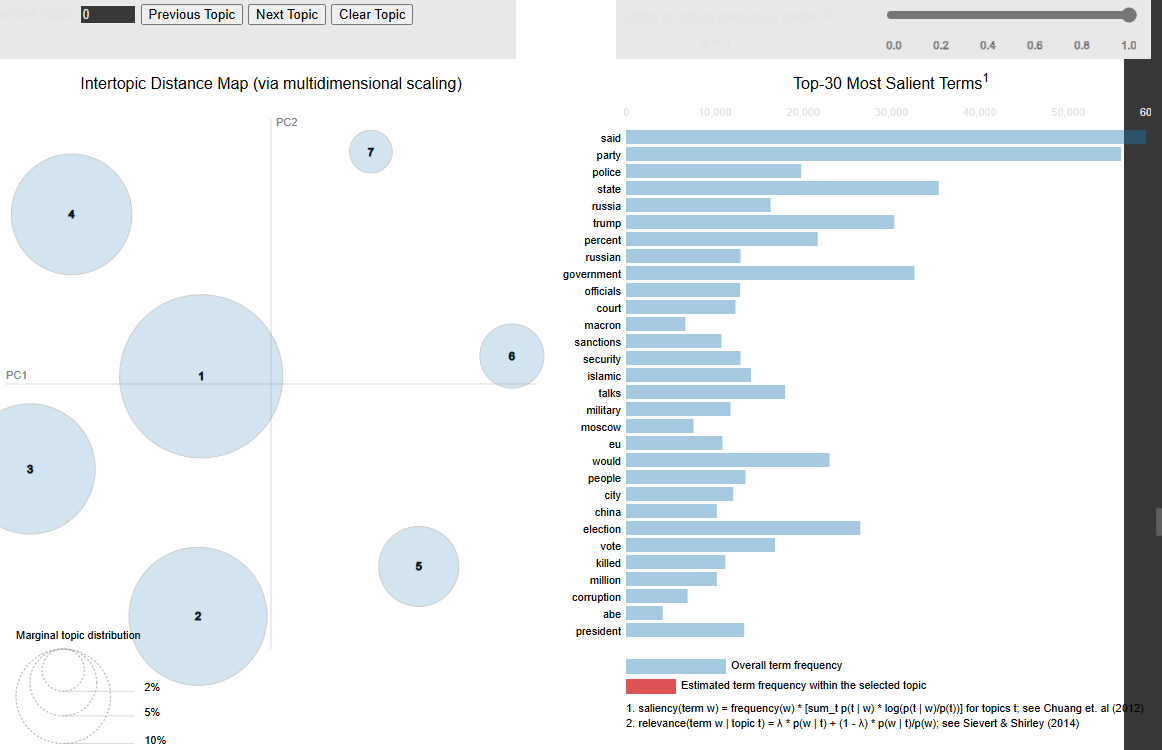
This employs a creative approach to refine topics generated by Latent Dirichlet Allocation (LDA) using TF-IDF (Term Frequency-Inverse Document Frequency) removal. This technique aims to enhance topic quality by filtering out common or less informative words, ultimately leading to more distinctive and coherent topic representations. TF-IDF, a statistical measure, reflects a word's importance within a document and across a collection of documents. By selectively removing terms with low TF-IDF scores, the analysis prioritizes words that are more relevant and discriminative for topic modeling.

### 5.1.1. The Process

The technique involves several steps: First, the TF-IDF scores are calculated for every term in the corpus using the gensim library's ‘TfidfModel’. Next, a threshold is defined to identify low-value terms. Terms with scores below this threshold are considered less important for representing topics. The code then iterates through the corpus, locating these low-value terms and removing them from the document representation. Finally, the LDA model is trained on this refined corpus, where less informative words have been filtered out, leading to potentially more meaningful and distinguishable topics.

### 5.1.2. Rationale and Benefits

This creative approach seeks to improve the quality of topic modeling by focusing on terms that truly distinguish topics within the dataset. By removing common or less informative words, the LDA model can better identify underlying themes and patterns, leading to more interpretable topics. This technique offers several benefits: enhanced topic coherence, as terms within a topic become more closely related; improved topic distinctiveness, as common terms that might blur topic boundaries are removed; and reduced noise, leading to a clearer representation of underlying themes within the text data. The creative use of TF-IDF removal for topic refinement contributes to a more insightful and meaningful analysis of the text data, enhancing the overall understanding of its thematic structures.

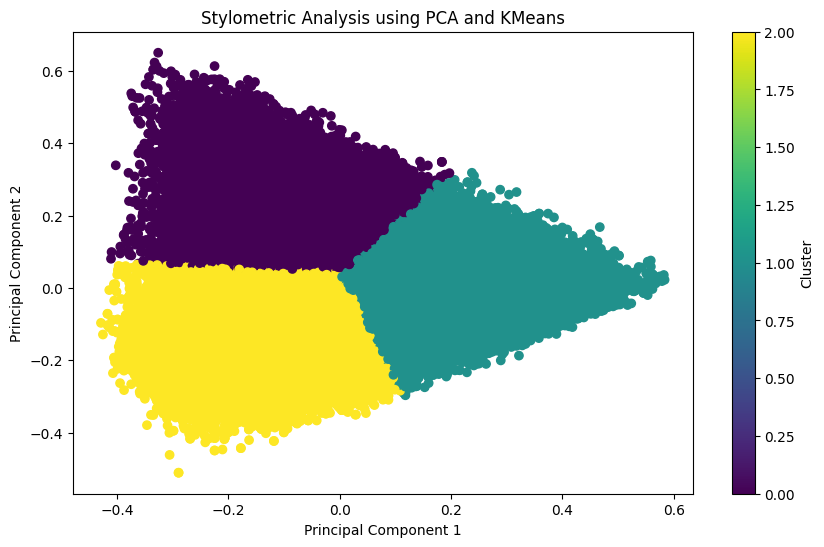


Each circle represents a topic, with size indicating prevalence and proximity suggesting semantic similarity. Clusters of circles indicate related topic areas. Selecting a topic reveals its most relevant terms in a bar chart, aiding interpretation. Overlapping circles may suggest closely related topics but could also indicate a need for model refinement. The overall distribution of circles provides insights into the topic landscape of the corpus, revealing patterns in information organization. By analyzing the size, proximity, and overlap of circles, alongside the relevant terms for each topic, researchers can gain a comprehensive understanding of the topics discussed in their text data and their relationships. This understanding is crucial for extracting meaningful insights and patterns from textual data.

# Task 6: Stylometric Analysis and Visualization

In stylometry, Principal Component Analysis (PCA) serves as a crucial technique for reducing the dimensionality of text data and uncovering underlying patterns in writing style (Craig, 2024). By transforming high-dimensional feature vectors, such as word frequencies or sentence lengths, into a lower-dimensional space, PCA captures the most significant variations in writing style. This dimensionality reduction allows for easier visualization and analysis of stylometric features, aiding in tasks like authorship attribution, genre classification, and text similarity detection. Essentially, PCA helps distill the essence of writing style into a few principal components, enabling researchers to identify key stylometric markers and distinguish between different authors or writing styles more effectively.

In stylometry, KMeans clustering serves as a powerful tool for grouping similar text documents or sections based on their writing style characteristics. By analyzing various features like word frequencies, sentence structures, and punctuation usage, KMeans aims to partition the data into clusters where texts within the same cluster exhibit greater stylistic similarities compared to those in other clusters (Eder, 2017). This approach aids in tasks like authorship attribution, where texts potentially written by the same author are grouped together or genre classification, where texts belonging to similar genres are clustered. KMeans clustering, therefore, provides a valuable method for identifying and visualizing stylistic relationships within a collection of texts, facilitating deeper understanding of writing patterns and potential authorship.



The scatterplot, generated using Principal Component Analysis (PCA) and KMeans clustering, visually represents stylometric similarities between different text sections. Each point on the plot corresponds to a text section, and its position is determined by the first two principal components derived from the text's features. Points clustered closely together indicate text sections with similar writing styles or characteristics, suggesting they might originate from the same author or share common themes. Conversely, points further apart represent text sections with greater stylometric differences. The color-coding, if present, further distinguishes clusters based on KMeans, highlighting potential groupings of similar texts. Overall, this visualization provides a concise overview of the relationships between text sections based on their stylometric properties, enabling identification of distinct writing styles and potential authorship patterns within the dataset.

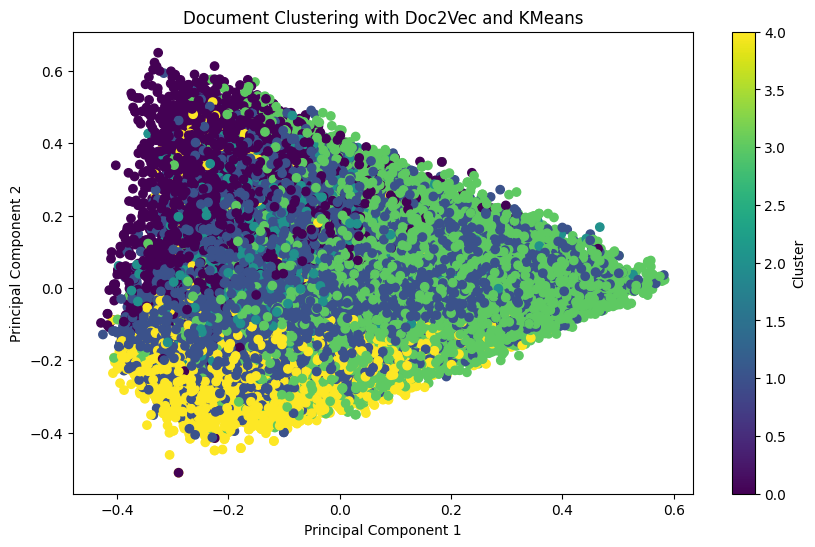
A diagram of a diagram

Description automatically generated with medium confidence

The dendrogram visually represents the hierarchical relationships between different text sections based on their stylometric features. Each branch in the dendrogram corresponds to a cluster of text sections, and the height of the branch indicates the dissimilarity between the clusters it connects. The lower the branch, the more similar the connected clusters. The dendrogram allows us to see how text sections are grouped together based on their stylometric properties. By analyzing the branching patterns and the distances between clusters, we can infer the degree of similarity between different text sections and potentially identify distinct writing styles or authorial patterns within the dataset. The vertical lines represent the clusters and the horizontal lines represent the distance between clusters. The longest vertical line represents the main clusters and as the lines become shorter, it represents sub-clusters. The height of the connection (i.e., the y-coordinate) indicates the distance between two clusters. So, for example, the clusters joined first are more similar to one another than the clusters joined last. This allows us to quickly interpret the hierarchy of the clusters.

# Task 7: Document Clustering with Word2Vec or Doc2Vec

This employs Word2Vec and Doc2Vec embeddings followed by KMeans clustering to group similar documents based on semantic content. Each document is assigned to a cluster, with documents within the same cluster sharing common themes or topics (Dai, Olah and Le, 2015). Visualizing the clusters with PCA helps in understanding their relationships and separation in a reduced dimensional space. Qualitative analysis, involving examining the content of documents within each cluster, aids in identifying characteristic themes and labeling the clusters. Identifying frequent keywords or topics within clusters further clarifies their semantic content. Comparing results obtained using Word2Vec and Doc2Vec reveals potential differences in how these embeddings capture document semantics and influence cluster formation. Ultimately, this clustering analysis reveals the underlying semantic structure of the text data, enabling efficient organization, understanding, and downstream applications like document summarization and information retrieval. By grouping documents based on their meaning, it provides a powerful tool for exploring and interpreting large text collections.



The scatterplot visualizes the relationships between documents based on their semantic similarities, as captured by Doc2Vec and clustered using KMeans. Each point on the plot represents a document, and its position is determined by the first two principal components derived from the Doc2Vec document embeddings. Points clustered closely together indicate documents with similar semantic content or themes, suggesting they might belong to the same topic or category. Conversely, points further apart represent documents with greater semantic differences. The color-coding further distinguishes clusters based on KMeans, highlighting potential groupings of semantically related documents. Overall, this visualization provides a concise overview of the semantic landscape within the dataset, enabling identification of distinct document clusters and potential topic areas within the collection.

# Task 8: Dependency Parsing and Advanced Structures

Dependency parsing analyzing the grammatical structure of sentences by identifying relationships between words (Milano and Schilder, 2009). This process involves generating dependency parse trees, where words are represented as nodes connected by edges labeled with dependency relations, indicating their grammatical roles (e.g., subject, object). SpaCy provides access to token attributes like ‘token.dep\_’ and ‘token.head.text’ for understanding individual word roles and relationships. Visualization with ‘displacy.render(style="dep")’ further aids in comprehending sentence structure and word hierarchies. By examining dependency relations and head words, key sentence elements and their contributions to overall meaning are revealed. Dependency parsing also enables the identification of advanced grammatical structures like noun phrases and clauses, offering a deeper understanding of sentence organization. Ultimately, this analysis provides valuable insights into how language is structured and sentences are formed, facilitating various NLP tasks such as text understanding and information extraction by revealing the grammatical connections between words.

This utilizes spaCy's displacy library to visualize and explain dependency structures within sentences. The primary visualization is a dependency tree, where words are nodes connected by edges labeled with dependency relations, revealing grammatical relationships like subject, object, and modifier. Understanding common relations such as nsubj, dobj, and amod is crucial for interpreting the tree. By examining the visualization, key structures like noun phrases, verb phrases, and clauses can be identified, providing insights into sentence meaning and organization. The hierarchical structure of the tree showcases word dependencies and contributes to a deeper understanding of sentence construction. Dependency parsing also allows for exploring advanced structures in complex sentences with multiple clauses and nested components. Overall, these visualizations and explanations offer a powerful tool for understanding grammatical relationships and sentence organization, enhancing our ability to analyze and interpret language effectively for various NLP tasks.

# Task 9: Insights and Real-World Application

Through a comprehensive NLP analysis encompassing various techniques, several key insights emerged regarding the text data. Preprocessing revealed the significance of noise reduction and text standardization for effective analysis, while POS tagging and NER illuminated the grammatical roles and key entities within sentences. Sentiment analysis provided an understanding of the prevailing emotional tone across documents, identifying predominantly neutral sentiment with smaller proportions of positive and negative sentiments. Topic modeling uncovered prominent themes including political discourse, international relations, and economic matters, showcasing the diverse subject matter within the corpus. Stylometric analysis, using PCA and KMeans, revealed clusters of documents with similar writing styles, suggesting potential distinctions in authorship or content focus. Furthermore, dependency parsing unveiled the grammatical relationships between words, facilitating a deeper comprehension of sentence structure and meaning. Overall, this NLP analysis provided a multifaceted view of the text data, revealing its thematic structures, stylistic variations, emotional tone, and underlying grammatical relationships, ultimately enabling a richer understanding of the content and its potential implications.

These findings offer significant potential for real-world applications in business analytics, sentiment analysis can be employed to gauge customer feedback, analyze market trends, and inform product development. Topic modeling can assist in understanding customer preferences and tailoring marketing campaigns to target specific groups. Social media analysis can benefit from sentiment analysis to assess brand reputation and identify emerging trends. Named entity recognition enables tracking mentions of key individuals or organizations across platforms. Researchers can leverage topic modeling to identify research gaps and explore emerging areas of interest within a body of literature. Stylometric analysis can be applied to identify authorship or analyze writing styles in historical documents. Furthermore, dependency parsing aids in developing more accurate and sophisticated language models for machine translation and text summarization, benefiting fields like research, content creation, and information dissemination. The tools and techniques used in this project empower businesses, researchers, and social media analysts to extract valuable insights from text data, aiding in strategic decision-making, enhancing customer engagement, and advancing knowledge discovery.

The insights derived from this NLP analysis can significantly impact decision-making and user behavior across various domains. By understanding the prevalent sentiment expressed in text data, businesses can tailor their products, services, and marketing strategies to better align with customer preferences and address concerns. For instance, negative sentiment surrounding a product feature could prompt a redesign, while positive feedback could highlight successful elements to emphasize. Identifying key topics through topic modeling empowers organizations to focus their resources and prioritize content that resonates with target audiences. Stylometric analysis can inform authorship attribution, potentially influencing content moderation or fraud detection strategies. Understanding dependency structures and sentence meaning through parsing enables development of more nuanced and effective chatbots or virtual assistants, leading to enhanced user experiences and more natural interactions. In research, topic modeling assists in identifying knowledge gaps and guiding future research directions, ultimately influencing academic discourse and knowledge discovery. Overall, the insights gleaned from this NLP analysis offer valuable information that can drive informed decision-making, improve product development, enhance customer engagement, and advance various fields by providing a deeper understanding of language and its impact on human behavior.

# Bonus Task: Implement an Advanced NLP Technique

## Text Summarization Using BART

This task focused on text summarization using the BART model, demonstrating its ability to condense lengthy texts while preserving key information. Applied to the news dataset, BART effectively generated concise summaries of news articles, capturing the main events and perspectives conveyed in the original text (Dharrao *et al.*, 2024). This functionality aligns well with the insights gained from previous tasks, particularly topic modeling and sentiment analysis. Topic modeling identified prominent themes within the news articles, while sentiment analysis gauged the overall emotional tone. By incorporating BART-generated summaries, users can quickly grasp the core message of each article, facilitating efficient content exploration and understanding. This integration enhances the value of previous analysis by providing readily accessible summaries aligned with identified topics and sentiments. Moreover, the bonus task highlights the practical applicability of advanced NLP techniques for real-world scenarios like news aggregation and content curation. By automatically summarizing large volumes of text, BART empowers users to efficiently navigate information-rich environments, aiding in informed decision-making and knowledge acquisition. Ultimately, the bonus task's results further demonstrate the potential of NLP to unlock valuable insights from text data and enhance human understanding of complex textual information.

# References

Cavnar, W.B. and Trenkle, J.M. (2001) ‘N-Gram-Based Text Categorization N-Gram-Based Text Categorization’, *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, (May), pp. 1–14.

Craig, H. (2024) ‘Principal components analysis in stylometry’, *Digital Scholarship in the Humanities*, 39(1), pp. 97–108. Available at: https://doi.org/10.1093/llc/fqad083.

Dai, A.M., Olah, C. and Le, Q. V. (2015) ‘Document Embedding with Paragraph Vectors’, pp. 1–8. Available at: http://arxiv.org/abs/1507.07998.

Dharrao, D. *et al.* (2024) ‘Summarizing Business News: Evaluating BART, T5, and PEGASUS for Effective Information Extraction’, *Revue d’Intelligence Artificielle*, 38(3), pp. 847–855. Available at: https://doi.org/10.18280/ria.380311.

Eder, M. (2017) ‘Visualization in stylometry: Cluster analysis using networks’, *Digital Scholarship in the Humanities*, 32(1), pp. 50–64. Available at: https://doi.org/10.1093/llc/fqv061.

Jelodar, H. *et al.* (2019) *Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey*, *Multimedia Tools and Applications*. Available at: https://doi.org/10.1007/s11042-018-6894-4.

Kherwa, P. and Bansal, P. (2020) ‘Topic Modeling: A Comprehensive Review’, *EAI Endorsed Transactions on Scalable Information Systems*, 7(24), pp. 1–16. Available at: https://doi.org/10.4108/eai.13-7-2018.159623.

Milano, P. and Schilder, F. (2009) ‘Book Reviews’. Available at: https://doi.org/10.1007/978-1-84800-078-0.

Webster, J.J. and Kit, C. (1992) ‘Tokenization as the initial phase in NLP’, (May), p. 1106. Available at: https://doi.org/10.3115/992424.992434.

# Appendix

OneDrive Link: [NLP Final Coursework](https://liveplymouthac-my.sharepoint.com/:f:/g/personal/10899179_students_plymouth_ac_uk/EnCvY-a175NOhd3BYc8nM_0BVd7tVdn-vwKQwmyil4kBng?e=ExjpVB)

GitHub Link: