

Task 1: Use NLP techniques to analyze a collection of texts

Submitted to: Pro. Dr. Frank Passing

Submitted by: Dost Muhammad

Matriculation no: 92128703

Concept phase

GitHub: <https://github.com/Dost-DS/project-DA>

Concept Phase: NLP analysis of citizen complaints

Objective and data source

The objective of this project is to analyze a large collection of unstructured textual complaints in order to identify the most prevalent topics discussed by citizens. Such insights can help decision-makers understand recurring concerns that are difficult to detect through manual inspection. As a representative dataset, a publicly available collection of Consumer Financial Protection Bureau (CFPB) complaint narratives is used. Although originating in the financial domain, the dataset closely resembles municipal complaint data in structure and content: it consists of large volumes of free-text complaints submitted by individuals, making it suitable for applying Natural Language Processing (NLP) techniques to real-world, unstructured data.

Data preprocessing

Before analysis, the text data is transformed into a clean and consistent format. The preprocessing strategy includes converting all text to lowercase, removing URLs, email addresses, numbers, and special characters, and normalizing whitespace. The cleaned text is then tokenized into individual words. Stopwords (both general English stopwords and domain-specific terms) are removed to reduce noise. Finally, lemmatization is applied to reduce words to their base form, ensuring that grammatical variations of the same concept are treated consistently. Empty or invalid texts are filtered out after preprocessing.

Text vectorization

Since most NLP algorithms require numerical input, the cleaned texts are converted into numeric representations using multiple vectorization approaches. First, a CountVectorizer is used to represent documents as word-frequency matrices. This representation is well-suited for probabilistic topic models. Second, TF-IDF (Term Frequency–Inverse Document Frequency) is applied to emphasize words that are distinctive within documents, which is particularly useful for matrix decomposition techniques. Third, dense document embeddings generated with spaCy are used to capture semantic similarities between texts.

Topic extraction techniques

To extract prevalent topics, multiple complementary techniques are applied. Latent Dirichlet Allocation (LDA) is used on the Count Vectorizer output to discover probabilistic topic structures based on word frequencies. Non-negative Matrix Factorization (NMF) is applied to TF-IDF vectors to produce highly interpretable and distinct topics. In addition, K-means clustering is applied to

spaCy embeddings to group semantically similar complaints. Using multiple methods allows for comparison and increases the robustness of the results.

Tools and libraries

The analysis is implemented in Python using widely adopted libraries: pandas and NumPy for data handling; NLTK and spaCy for preprocessing and linguistic analysis; scikit-learn for vectorization and topic modeling; and matplotlib, seaborn, and wordcloud for visualization. All configuration parameters are centralized to ensure reproducibility and transparency.