File Similarity Detection

Introduction:

In this task, we aim to analyze and categorize documents from various domains, such as education, health, and entertainment, by leveraging natural language processing (NLP) techniques and machine learning algorithms. We start by extracting and preprocessing the content from PDF, DOCX, and TXT files, followed by cleaning the text data through tokenization, stopword removal, and lemmatization. Using Word2Vec, we generate word embeddings to create sentence vectors, and employ Latent Dirichlet Allocation (LDA) for topic modelling. We then combine text, topic, and metadata similarities to cluster the documents. The clustering quality is evaluated using the silhouette score. Additionally, we perform supervised classification using logistic regression, decision trees, and random forests to determine the best performing model for classifying the documents based on their content. The results demonstrate the effectiveness of various classification algorithms, highlighting the superior performance of the Random Forest classifier.

Step-by-step Approach:

Library Installation:

- Install necessary libraries such as PyPDF2, python-docx, contractions, and unidecode.

File Reading and Data Extraction:

- Define functions to extract text from PDF and DOCX files.
- Loop through the files in the specified directory and read the content based on file type (PDF, DOCX, TXT).
- Store the extracted content along with file metadata in a DataFrame.

Text Preprocessing:

- Download required NLTK resources like stopwords and wordnet.
- Define a function to clean the text by removing contractions, converting to lowercase, removing non-alphabetical characters, decoding, tokenizing, removing stopwords, and lemmatizing the tokens.
- Apply the text cleaning function to the content in the DataFrame.

Label Encoding:

- Create a mapping for labels (education, health, entertainment) and encode the labels numerically.
- Split the data into training and testing sets.

Word2Vec Vectorization:

- Train a Word2Vec model on the cleaned text data to create word embeddings.
- Define a function to generate sentence vectors by averaging the word vectors in each sentence.

- Create sentence vectors for the training, testing, and all data sets.

Metadata Extraction:

- Define a function to extract metadata from the file names.
- Encode the file names to numerical values and scale the metadata features using StandardScaler.

Topic Modeling with LDA:

- Use Gensim's LDA model to perform topic modeling on the cleaned content.
- Define a function to get topic distributions for each document.
- Extract topic distributions for the entire dataset.

Combined Similarity Calculation:

- Define a function to calculate combined similarity using text vectors, topic vectors, and metadata vectors with specified weights.
- Calculate the combined similarity matrix for all documents.

Clustering:

- Define a function to cluster documents based on the combined similarity matrix using a specified threshold.
- Perform clustering and print the clusters.

Silhouette Score Calculation:

- Convert the combined similarity matrix to distances and calculate the silhouette score to evaluate clustering quality.

Classification:

- Define and initialize various classification models (Logistic Regression, Decision Tree, Random Forest).
- Define a function to evaluate each model using accuracy, precision, recall, and F1 score.
- Fit the models on the training data and evaluate on the testing data.
- Compare the performance of the models and plot the results.

Challenges:

- *Dataset acquisition:*

  The major challenge encountered in this task is the creation of a cohesive dataset, as the data has been sourced from various websites. This collection process ensures that only certain files have matching contextual information.

- *Various file formats handling:*

Handling different file formats (PDF, DOCX, TXT) requires specialized methods for data extraction, as each format has its unique structure and encoding.

- *Tuning the hyperparameters:*

Ensuring high Silhouette scores for classification models involves selecting the right alpha, beta, threshold and tuning hyperparameters, which can be time-consuming.

Future improvements:

- *Advanced text extraction techniques:*

Invest in research and development of advanced text extraction techniques specifically tailored for handling complex layouts, and tables within PDFs. This could involve leveraging computer vision algorithms for better text extraction accuracy.

- *Advanced word embeddings:*

BERT word embeddings can be considered as this captures the contextual information based on the whole document. These embeddings often outperform traditional word embeddings like Word2Vec in capturing semantics and context.

- *New hyperparameters consideration:*

Some of the hyperparameters can be explored in order to improve the model’s performance.