Text Representations

Objective: Generate document representations using vector space models (TF/TF-IDF) and semantic vector models (word embeddings) to analyze a subset of the "20 Newsgroups" dataset.

This practice is publicly available at
https://github.com/alvaro-francisco-gil/text-mining/tree/main/02_text_representation

Extract Message Body

1. **Remove Headers**: Identify and remove everything before the first blank line in the email.
2. **Filter Out Email Addresses**: Remove lines containing email addresses using a regex pattern.
3. **Exclude Proper Nouns**: Remove lines with only 2-3 capitalized words.
4. **Remove Signatures**: Detect and exclude lines matching common signature patterns (e.g., "--", "Kind regards", "Sent from my iPhone") and any subsequent lines until a blank line is encountered.
5. **Preserve Quoted Lines**: Retain quoted lines (starting with ">") unless they are part of a signature or irrelevant.
6. **Reassemble Message Body**: Combine the remaining meaningful lines into the final cleaned message body.

Preprocessing

7. **Remove Punctuation**: All punctuation is stripped from the text.
8. **Remove Numbers:** Numbers are removed if the remove_numbers flag is set to True.H

9. **Convert to Lowercase**: The entire text is converted to lowercase for uniformity.
10. **Tokenization**: The text is split into individual words.
11. **Remove Stop-Words**: Common English stop-words are removed using an NLTK stop-word list.
12. **Lemmatization or Stemming:** Words are normalized by applying lemmatization (default) using WordNetLemmatizer or stemming using PorterStemmer, based on the use_lemmatization flag.
13. **Reassemble Text**: The processed words are joined back into a single string.

TF and TF-IDF Representations

14. **TF (Term Frequency)**: A TfidfVectorizer is used to compute the raw term frequency matrix.
15. **TF-IDF (Term Frequency-Inverse Document Frequency)**: A standard TfidfVectorizer computes the TF-IDF matrix for the documents.

Word2Vec Embeddings

16. For each preprocessed document, word embeddings are retrieved from a pre-trained Word2Vec model.
17. Two representations are generated:
    - **Average Vector**: The mean of all word vectors in the document.
    - **Sum Vector**: The sum of all word vectors in the document.

Output

1. **TF Matrix**: A sparse matrix representing term frequencies for each document.
2. **TF-IDF Matrix**: A sparse matrix representing weighted term frequencies (TF-IDF) for each document.
3. **Word2Vec Average Vectors**: Dense vectors representing the average of word embeddings for each document.
4. **Word2Vec Sum Vectors**: Dense vectors representing the sum of word embeddings for each document.

Conclusions

- This practice demonstrates how to preprocess text data and generate multiple types of document representations (TF, TF-IDF, and Word2Vec embeddings). Each representation captures different aspects of textual information.
- **TF and TF-IDF** are effective for sparse representations and are widely used in traditional machine learning models. However, they do not capture semantic relationships between words.
- **Word2Vec embeddings**, on the other hand, provide dense and semantically meaningful representations by leveraging pre-trained word vectors. These are particularly useful in deep learning applications or tasks requiring semantic understanding.
- Combining these representations can enhance downstream tasks like classification, clustering, or similarity analysis by leveraging both lexical and semantic features.
- The produced outputs provide a comprehensive basis for further analysis or modeling, showcasing the importance of selecting appropriate text representation techniques based on task requirements.