Text Classification Documentation: Suspicious Transaction and Activity Reports Analysis

Introduction

In the financial domain, vast amounts of textual data are produced in forms such as transaction reports and customer communications. This project aims to perform text analytics on Suspicious Transaction Reports (STRs) and Suspicious Activity Reports (SARs) to extract meaningful insights which can be instrumental in identifying suspicious activities like Money Laundering (ML), Terrorism Financing (TF), and Proliferation Financing (PF).

Project Scope

The analysis consists of several steps including data preparation, topic discovery, text classification, and data visualization.

Assumptions

- The keywords or rules used for initial labeling are adequate.
- The model can generalize and categorize new unseen financial reports accurately.
- The textual content of the reports is sufficient for analysis and classification.

Step 1: Library Installations and Loading

Libraries such as `pandas` and `nltk` are installed and loaded to facilitate data handling and natural language processing tasks.

```
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from nltk.probability import FreqDist

nltk.download('punkt')
nltk.download('stopwords')
```

Step 2: Data Loading

Data is loaded from text files and the content is read into Python strings.

```
with open("file_path", "r") as file:
    text_data = file.read()
```

Step 3: Data Cleaning

The data is tokenized, stopwords are removed, and punctuation marks are filtered out to prepare the data for analysis.

```
sentences = sent_tokenize(text_data)
words = word_tokenize(text_data)
...
filtered_words = [word for word in filtered_words if word not in punctuation]
```

Step 4: Text Analytics

Basic text analytics is performed to understand the distribution of words, document lengths, and common phrases.

```
freq_dist = FreqDist(filtered_words)
...
```

Step 5: Topic Modelling

Latent Dirichlet Allocation (LDA) is employed to discover the underlying topics within the dataset.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation
...
lda_model = LatentDirichletAllocation(n_components=n_topics, random_state=42)
lda_model.fit(dtm)
```

Step 6: Text Classification

A simplistic Naive Bayes classifier is trained using manual labeling based on keywords for categorizing reports into ML, TF, and PF.

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import classification_report
...
clf = MultinomialNB()
clf.fit(X_train_vec, y_train)
...
```

A custom function is also provided to categorize a report based on the presence of certain keywords.

```
def categorize_report(report):
    ...
    return "Category"

categorization = categorize_report(sample_report)
```

Work in Progress

The Naive Bayes classifier training and evaluation are in progress. Additional steps may include tuning the model, exploring other classification algorithms, and possibly

integrating more complex NLP techniques to improve classification accuracy.

This documentation provides an organized walkthrough of the code, summarizing the key steps involved in the analysis of Suspicious Transaction and Suspicious Activity Reports. Each section provides a brief description of the tasks performed and the Python code snippets for executing those tasks.