Automated Tagging of Incident Learning Narratives: A Technical Summary

# Abstract

This project aims to develop an automated system for tagging and categorizing incident learning entries. These tags serve to identify recurring themes, issues, or critical elements within the narratives to facilitate future reviews and analyses. The system employs Natural Language Processing (NLP) techniques, specifically Term Frequency-Inverse Document Frequency (TF-IDF) and n-gram extraction.

# Objective

The primary objective is to automatically parse through a database of incident learning narratives and assign tags to these entries. These tags serve as succinct summaries and categorizations of the narratives, facilitating easier future analyses, searches, and reviews.

# Data Source

The dataset is an Excel file that comprises several columns, one of which is labeled '105.Narrative.' This column contains textual summaries or narratives describing various incidents and serves as the primary data source for this project.

# Methodology

## Data Preprocessing

The first step in the data pipeline involves cleaning and preparing the text data for feature extraction. The textual narratives are converted to lowercase, and tokens that are not alphabetic are filtered out.

Python Code Example:

df['Preprocessed\_Narratives\_FullWords'] = df['105.Narrative'].apply(preprocess\_text)

## Feature Extraction

### Single-Word Tags

The system identifies the top 50 most frequently occurring words across all narratives. These high-frequency words are considered as potential single-word tags.

Python Code Example:

filtered\_top\_50\_words\_for\_tags = [word for word, freq in words\_freq[:50]]

### Context-Based Multi-Word Tags

The system also allows for predefined, contextually relevant multi-word tags such as 'Dosimetric Errors,' 'Patient Delayed,' and 'Treatment Planning.'

Python Code Example:

context\_based\_multi\_word\_tags = [...]

### N-grams

In addition to single-word and multi-word tags, the system also extracts frequent n-grams (bi-grams, tri-grams, and four-grams). An n-gram is a contiguous sequence of n items from a given sample of text or speech. For this project, n-grams that occur at least five times across all narratives are considered.

Python Code Example:

frequent\_bigrams = extract\_frequent\_ngrams(df['Preprocessed\_Narratives\_FullWords'], 2)

## Tagging Mechanism

### TF-IDF Vectorization

The Term Frequency-Inverse Document Frequency (TF-IDF) technique is employed to convert the narratives into a mathematical form. This vectorization technique produces a high-dimensional vector for each narrative, where each dimension corresponds to one potential tag. The TF-IDF score of each tag within each narrative is calculated to gauge its relevance.

Python Code Example:

vectorizer = TfidfVectorizer(vocabulary=all\_possible\_tags, ngram\_range=(1, 4))

### Relevance-Based Tag Selection

After obtaining the TF-IDF scores, the system selects the top five most relevant tags for each narrative. These tags are considered the most descriptive and are used to summarize the key elements of each narrative.

Python Code Example:

df['Top\_5\_Relevant\_Tags'] = [find\_top\_relevant\_tags(tfidf\_matrix[i, :]) for i in range(tfidf\_matrix.shape[0])]

# Results and Output

The final output is an Excel file that includes the original dataset augmented with a new column called 'Top\_5\_Relevant\_Tags.' This column contains up to five of the most relevant tags for each narrative. Additionally, the Excel file contains separate sheets that list all the unique tags, frequent bi-grams, tri-grams, and four-grams identified in the narratives.

# Conclusion

This project successfully automates the tagging process for a database of incident learning narratives. By employing advanced NLP techniques, the system is capable of summarizing, categorizing, and facilitating the analysis of these critical incident reports.

df['Top\_5\_Relevant\_Tags'] = [find\_top\_relevant\_tags(tfidf\_matrix[i, :]) for i in range(tfidf\_matrix.shape[0])]

# Results and Output

The final output is a new Excel file with the following components:

**1.** The original dataset augmented with a new column called 'Top\_5\_Relevant\_Tags.' This column contains up to five of the most relevant tags for each narrative.

**2.** Additional sheets that list all the unique tags, frequent bi-grams, tri-grams, and four-grams identified in the narratives.

# Conclusion

The project successfully automates the tagging process for a database of incident learning narratives. By employing advanced NLP techniques, the system is not only capable of summarizing and categorizing these critical incident reports but also facilitates their future analysis. This system holds significant potential for improving the efficiency and effectiveness of incident learning systems in clinical settings.