Class BS Data Science

Subject: Text Mining

Day 1: Date: 25/11/2024

Topic : TF-IDF

Recommended Books

1. Text Mining with R: A Tidy Approach by Julia Silge and David Robinson
2. **Foundations of Statistical Natural Language Processing** by Christopher Manning and Hinrich Schütze
3. **Mining the Social Web** by Matthew A. Russell
4. Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS

Slide 1: Title Slide

Title: Understanding TF-IDF: A Key Text Mining Technique
Subtitle: Enhancing Text Analysis with Weight-Based Features
Presented by: [Your Name]
Date: [Insert Date]

Slide 2: What is TF-IDF?

- **Definition:**
  TF-IDF is a statistical measure that evaluates how important a word is to a document within a collection (or corpus).
- **Purpose:**
  Assigns higher weights to words that are unique or relevant to a document and lower weights to common words across the corpus.

Slide 3: Components of TF-IDF

1. **Term Frequency (TF):**
   Measures the frequency of a word in a document.

$$TF(w,d) = \frac{\text{Number of occurrences of word } w \text{ in document } d}{\text{Total number of words in document } d}$$

2. **Inverse Document Frequency (IDF):**
   Measures how unique a word is across all documents in the corpus.

$$IDF(w) = \log\left(\frac{N}{1 + DF(w)}\right)$$

   Where:

   - N: Total number of documents.
   - DF(w): Number of documents containing the word www.
   - Adding 1 to DF(w) avoids division by zero.
   -

3. **TF-IDF Score:**
   The product of TF and IDF.

$$TF\text{-}IDF(w,d) = TF(w,d) \cdot IDF(w)$$

Slide 4: Why Use TF-IDF?

- Removes the bias of frequently occurring but less important words (e.g., "and," "the").
- Helps identify words that uniquely represent a document.
- Commonly used in tasks like text classification, clustering, and information retrieval.

Slide 5: Example Corpus for TF-IDF Calculation

Corpus of 3 Documents:

1. Document 1: *"Text mining is fun."*
2. Document 2: *"Mining data helps extract insights."*
3. Document 3: *"Data mining and text analytics are valuable."*

Slide 6: Step 1: Compute Term Frequencies (TF)

Word	TF (Doc 1)	TF (Doc 2)	TF (Doc 3)
Text	1/4	0	1/6
Mining	1/4	1/5	1/6
Is	1/4	0	0
Fun	1/4	0	0
Data	0	1/5	1/6
Helps	0	1/5	0
Extract	0	1/5	0
Insights	0	1/5	0
And	0	0	1/6
analytics	0	0	1/6
Are	0	0	1/6
Valuable	0	0	1/6

Slide 7: Step 2: Compute Document Frequencies (DF)

Word	DF (Number of Documents Containing the Word)
Text	2
Mining	3
Is	1
Fun	1
Data	2
Helps	1
Extract	1
Insights	1
And	1
analytics	1
Are	1
valuable	1

Slide 8: Step 3: Compute IDF

Using the formula:

$$IDF(w) = \log\left(\frac{N}{1 + DF(w)}\right)$$

Where $N = 3$ (Total documents):

Word	IDF
text	$\log\left(\frac{3}{1+2}\right) = 0$
mining	$\log\left(\frac{3}{1+3}\right) = -0.1249$
is	$\log\left(\frac{3}{1+1}\right) = 0.4055$
fun	$\log\left(\frac{3}{1+1}\right) = 0.4055$
data	$\log\left(\frac{3}{1+2}\right) = 0$

Continue similarly for other words. This will result in feature vectors for each document.

Slide 10: Applications of TF-IDF

1. **Text Classification:**
    - Example: Spam detection using email data.
2. **Information Retrieval:**
    - Example: Search engines rank documents based on relevance.
3. **Text Clustering:**
    - Example: Grouping news articles by topic.

Lab work

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Define the corpus
corpus = [
    "Text mining is fun",
    "Mining data helps extract insights",
    "Data mining and text analytics are valuable"
]

# Initialize TF-IDF Vectorizer
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(corpus)

# Display results
print("TF-IDF Feature Names:", vectorizer.get_feature_names_out())
print("TF-IDF Matrix:\n", tfidf_matrix.toarray())
```