Class BS Data Science

Subject: Text Mining

Day 1: Date: 26/11/2024

Topic : Similarity Scoring in Text Mining

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 2: Goal of the Lecture

To understand how similarity scoring techniques are used in text mining to compare documents, sentences, or words.

Slide 3: Objectives of the Lecture

1. Explain the concept of similarity scoring in text mining.
2. Explore key techniques like cosine similarity, Jaccard similarity, and Euclidean distance.
3. Provide practical examples and applications of similarity scoring in NLP.

Slide 4: What is Similarity Scoring?

- **Definition:**
  Measures how closely related two text entities (words, sentences, or documents) are in terms of their meaning or structure.
- **Importance in Text Mining:**
  - Clustering similar documents.
  - Information retrieval and search engines.
  - Recommendation systems.

Slide 5: Types of Similarity Measures

1. **Lexical Similarity:**
   - Based on the exact match of words (e.g., Jaccard Similarity).
2. **Semantic Similarity:**
   - Measures meaning similarity (e.g., Cosine Similarity, Word Embeddings).

Slide 6: Techniques for Similarity Scoring

1. Cosine Similarity

- Measures the cosine of the angle between two vectors.
- **Formula:**

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

Where:

- $\mathbf{A} \cdot \mathbf{B}$ is the dot product of vectors.
- $\|\mathbf{A}\|$ and $\|\mathbf{B}\|$ are the magnitudes of the vectors.

2. Jaccard Similarity

- Measures the overlap between two sets.
- **Formula:**

$$\text{Jaccard Similarity} = \frac{|A \cap B|}{|A \cup B|}$$

Where A and B are sets of words or tokens.

3. Euclidean Distance

- Measures the straight-line distance between two vectors.
- Formula:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

4. Semantic Similarity (Word Embeddings)

- Compares word vectors in a continuous vector space.

Slide 7: Example of Cosine Similarity

Documents:

- **Doc A:** "Text mining is fun."
- **Doc B:** "Text mining helps extract insights."

Step 1: Vocabulary

Vocabulary: ["text", "mining", "is", "fun", "helps", "extract", "insights"]

Step 2: Term Frequency Vectors

- Doc A: $[1, 1, 1, 1, 0, 0, 0]$
- Doc B: $[1, 1, 0, 0, 1, 1, 1]$

Step 3: Cosine Similarity

$$\text{Similarity} = \frac{(1\cdot 1)+(1\cdot 1)+(1\cdot 0)+(1\cdot 0)+(0\cdot 1)+(0\cdot 1)+(0\cdot 1)}{\sqrt{(1^2+1^2+1^2+1^2+0^2+0^2+0^2)}\cdot\sqrt{(1^2+1^2+0^2+0^2+1^2+1^2+1^2)}}$$

$$\text{Similarity} = \frac{2}{\sqrt{4}\cdot\sqrt{5}} = \frac{2}{\sqrt{20}} = 0.447$$

Slide 8: Example of Jaccard Similarity

Sets:

- **Doc A:** {"text", "mining", "is", "fun"}
- **Doc B:** {"text", "mining", "helps", "extract", "insights"}

Formula:

$$\text{Jaccard Similarity} = \frac{|A\cap B|}{|A\cup B|}$$

$$\text{Similarity} = \frac{|\{\text{"text"},\text{"mining"}\}|}{|\{\text{"text"},\text{"mining"},\text{"is"},\text{"fun"},\text{"helps"},\text{"extract"},\text{"insights"}\}|} = \frac{2}{7}$$

$$\text{Similarity} = 0.2857$$

Slide 9: Applications of Similarity Scoring

1. **Search Engines:**
   - Ranking documents by relevance to a query.
2. **Plagiarism Detection:**
   - Comparing two documents for overlapping content.
3. **Recommendation Systems:**
   - Recommending similar items based on user preferences.
4. **Clustering:**
   - Grouping similar documents or sentences.

Slide 10: Python Implementation

Cosine Similarity Example

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Sample documents
docs = ["Text mining is fun", "Text mining helps extract insights"]
```

```
# Convert to term frequency matrix
vectorizer = CountVectorizer()
tf_matrix = vectorizer.fit_transform(docs)

# Compute cosine similarity
cos_sim = cosine_similarity(tf_matrix, tf_matrix)
print("Cosine Similarity Matrix:\n", cos_sim)
```

Slide 11: Challenges in Similarity Scoring

1. **Lexical vs. Semantic Similarity:**
    - Lexical methods may fail to capture synonyms (e.g., "happy" and "joyful").
2. **High Dimensionality:**
    - Large vocabularies lead to sparse matrices.
3. **Contextual Similarity:**
    - Fixed methods (e.g., cosine) may ignore context-dependent meanings.

Slide 12: Conclusion

- Similarity scoring is vital for comparing text entities in NLP.
- Cosine similarity and Jaccard similarity are widely used in text mining tasks.
- Semantic approaches like word embeddings address limitations of traditional methods.