**Recognising Similar Texts**

**A Course Project Report**

**Submitted by**

**BY**

**M.GNANESHWARNATH (192211148)**

**G. VENKATA CHALAPATHI (192211150)**

**In partial fulfilment for the completion of the course**

**CSA1355 Theory of Computation with Finite Automata**

****

**SIMATS SCHOOL OF ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL**

**SCIENCES**

**CHENNAI – 602 105**

**TAMILNADU, INDIA**

**June - 2024**

**Abstract**

Recognizing similar texts is a fundamental task in Natural Language Processing (NLP) with broad applications, including search engines, recommendation systems, document clustering, and plagiarism detection. This project aims to explore and implement various techniques to measure text similarity, providing a comprehensive learning experience for students in the field of NLP.

The project involves several key components:

1. **Text Similarity Metrics**: Understanding and implementing various metrics such as cosine similarity, Jaccard similarity, and Euclidean distance to quantify the similarity between text documents.
2. **Feature Extraction**: Preprocessing text data through techniques like tokenization, stop word removal, and lemmatization to prepare it for further analysis.
3. **Vectorization Methods**: Converting text into numerical representations using methods like Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), and word embeddings (Word2Vec, GloVe, BERT).
4. **Similarity Metrics Application**: Using libraries like scikit-learn, genism, and Hugging Face's Transformers to implement and compare different similarity metrics, evaluating their effectiveness in various contexts.

By working on this project, students will gain hands-on experience with essential NLP techniques, learn to preprocess and vectorize text data, and apply similarity metrics to real-world use cases. The project serves as a practical introduction to the complexities of text analysis and provides a foundation for more advanced NLP applications and research.

**Introduction**

Recognizing similar texts is a pivotal task in the domain of Natural Language Processing (NLP), a subfield of artificial intelligence focused on the interaction between computers and human language. This task involves determining the degree of similarity between two or more text documents, which is essential for a variety of applications such as search engines, recommendation systems, document clustering, and plagiarism detection.

**Objective**

The primary objective of this project is to develop an understanding of the methodologies used to measure and analyze text similarity. Through this project, students will engage with key NLP techniques, gaining practical experience in text preprocessing, feature extraction, vectorization methods, and similarity metrics.

**Importance**

The ability to accurately recognize similar texts is crucial in today’s data-driven world. For instance, search engines rely on text similarity to retrieve relevant documents in response to user queries. Recommendation systems use it to suggest content similar to what users have previously engaged with. In academic and professional settings, plagiarism detection systems employ text similarity measures to identify copied content.

**Learning Outcomes**

By completing this project, students will be able to:

* Understand and apply different text similarity metrics.
* Perform feature extraction and preprocessing on text data.
* Utilize various vectorization techniques to convert text into numerical representations.
* Evaluate the performance of different similarity metrics and understand their applications.

**Applications**

This project has practical implications for a wide range of applications:

* **Search Engines**: Enhancing the relevance of search results by identifying documents similar to the user's query.
* **Recommendation Systems**: Suggesting content based on the similarity to previously interacted items.
* **Document Clustering**: Grouping related documents for better organization and retrieval.
* **Plagiarism Detection**: Identifying instances of copied content by comparing documents for similarities.

**Methodology**

#### **1. Problem Definition**

* **Objective:** To develop a system that can accurately measure the similarity between pairs of text documents.
* **Scope:** Focus on implementing and comparing basic text similarity metrics using straightforward preprocessing and vectorization techniques.

#### **2. Data Preparation**

* **Data Collection:** Gather a small dataset of text documents covering different topics.
* **Data Cleaning:** Remove special characters, punctuation, and perform basic text normalization (e.g., lowercase conversion).

#### **3. Feature Extraction and Vectorization**

* **Tokenization:** Split text into tokens (words or n-grams).
* **Vectorization:** Use basic methods like Bag of Words (BoW) or TF-IDF to convert text into numerical representations.

#### **4. Similarity Metrics Implementation**

* **Select Metrics:** Implement simple similarity metrics such as cosine similarity or Jaccard similarity.
* **Evaluation:** Calculate similarity scores between document pairs and evaluate using basic metrics (e.g., accuracy, mean similarity score).

#### **5. Experimentation and Analysis**

* **Experimental Setup:** Design experiments to compare the performance of different vectorization methods and similarity metrics.
* **Result Analysis:** Analyze and interpret results to understand the effectiveness of each approach in capturing text similarity.

#### **6. Conclusion and Presentation**

* **Conclusion:** Summarize findings and conclusions based on experimental results.
* **Presentation:** Prepare a concise presentation highlighting methodology, results, and implications for practical applications in NLP.

#### **7. Documentation**

* **Document Methodology:** Provide clear documentation of steps followed, code implementation, and experimental results.
* **Future Directions:** Discuss potential future enhancements or extensions to the project.

**Code**

import numpy as np

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

# Sample text documents

documents = [

"Natural Language Processing is a fascinating field.",

"NLP involves the interaction between computers and humans using natural language.",

"Machine learning can be used to improve NLP tasks."

]

# Step 1: Vectorize the documents using TF-IDF

vectorizer = TfidfVectorizer()

tfidf\_matrix = vectorizer.fit\_transform(documents)

# Step 2: Compute the cosine similarity matrix

cosine\_sim\_matrix = cosine\_similarity(tfidf\_matrix)

# Step 3: Output the similarity scores

for i, doc in enumerate(documents):

for j in range(i+1, len(documents)):

print(f"Similarity between document {i+1} and document {j+1}: {cosine\_sim\_matrix[i, j]:.4f}")

**RESULT**

**Experimental Setup**

* **Dataset**: Collected a diverse dataset of text documents covering various topics, including:
  + Document 1: "Natural Language Processing is a fascinating field."
  + Document 2: "NLP involves the interaction between computers and humans using natural language."
  + Document 3: "Machine learning can be used to improve NLP tasks."
* **Preprocessing**: Applied basic text cleaning steps such as converting text to lowercase, removing special characters, and tokenizing the text into individual words.
* **Vectorization**: Used Term Frequency-Inverse Document Frequency (TF-IDF) to convert text documents into numerical representations.

**Similarity Metrics Evaluation**

* **Cosine Similarity**: Implemented to measure the cosine of the angle between TF-IDF vectors of text documents.

**Performance Results**

* **Cosine Similarity Results**:
  + Similarity between Document 1 and Document 2: 0.1633
  + Similarity between Document 1 and Document 3: 0.0000
  + Similarity between Document 2 and Document 3: 0.0633

**Analysis**

* **Document Pair 1 and 2**: The similarity score of 0.1633 indicates a low degree of similarity. Both documents discuss NLP and natural language, but the difference in context and specific focus results in a relatively low score.
* **Document Pair 1 and 3**: The similarity score of 0.0000 indicates no similarity based on the TF-IDF representation. This makes sense as Document 1 focuses on NLP as a field, while Document 3 discusses machine learning applications in NLP.
* **Document Pair 2 and 3**: The similarity score of 0.0633 indicates very low similarity. Document 2 and Document 3 are somewhat related through their mention of NLP, but the different contexts and details lead to a low similarity score.

**Limitations**

* **Dataset Size**: The small dataset size may not capture the full range of text similarities and could lead to less reliable results.
* **Vectorization Method**: TF-IDF, while useful, might not capture deeper semantic meanings and contextual similarities between texts.

**Future Work**

* **Advanced Vectorization**: Explore more sophisticated vectorization methods such as word embeddings (Word2Vec, GloVe) or contextual embeddings (BERT) to better capture semantic relationships.
* **Larger Dataset**: Utilize a larger and more diverse dataset to improve the robustness and generalizability of the similarity measurements.
* **Additional Metrics**: Experiment with additional similarity metrics, such as Euclidean distance or more advanced models, to compare performance and effectiveness.

**Conclusion**

The project successfully implemented and evaluated basic text similarity metrics using TF-IDF vectorization and cosine similarity. The results indicate that while TF-IDF and cosine similarity can provide insights into text similarities, more advanced techniques and larger datasets are necessary for more accurate and meaningful comparisons in real-world applications.

**References**

** Books and Articles**

* Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press. This book provides a comprehensive overview of information retrieval techniques, including text preprocessing, vectorization methods, and similarity metrics.
* Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson. This book covers fundamental concepts in NLP, including feature extraction, text similarity, and advanced vectorization techniques.

** Research Papers**

* Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. arXiv preprint arXiv:1301.3781. This paper introduces Word2Vec, a popular word embedding technique that captures semantic relationships between words.
* Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805. This paper presents BERT, a state-of-the-art model for contextual embeddings.

** Online Resources**

* Scikit-learn Documentation. (n.d.). Retrieved from https://scikit-learn.org/stable/documentation.html. This documentation provides detailed information on implementing TF-IDF vectorization, cosine similarity, and other machine learning techniques.
* Gensim Documentation. (n.d.). Retrieved from https://radimrehurek.com/gensim/. Gensim is a robust library for topic modeling and document similarity analysis.
* NLTK (Natural Language Toolkit) Documentation. (n.d.). Retrieved from <https://www.nltk.org/>. NLTK provides tools for text preprocessing, tokenization, and other essential NLP tasks.