

Code Understanding Report

Generated: 2025-05-06 17:06:38

This report presents automated insights based on large language models and code analysis tools.

File: app.py

Summary

- This function takes a list of strings, and returns a matrix. The elements of the matrix represent the TF-IDF of each string in the list, with the rows being the different strings and the columns being the different features. The TF-IDF score for each feature is calculated based on the frequency of the words in the documents and on the frequency of the words in the documents within a corpus.

This is a common technique used for document classification and text mining. - The code snippet provided is a function definition in Python which returns the cosine similarity between two documents represented as lists of words. It's being used in a machine learning context where documents represent texts and the similarity between documents is to be determined by the cosine of the angle between their vectors. The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space. The cosine of the angle between them forms the cosine of the angle between the documents. - This python function calculates and stores the similarity score of text vectors in a list of tuples in the `s_vectors` global variable. It then searches the list for pairs of students who share the highest similarity score.

The function returns a list of tuples, each containing a pair of student names and the similarity score. The similarity score is calculated using the `similarity` function, which uses cosine similarity as the measure of similarity.

The `check_plagiarism` function is called at the end of a course, to ensure all text vectors have been compared to each other

Docstring

- `### Code: def vectorize(Text): return TfidfVectorizer().fit_transform(Text).toarray()`

Docstring:

This function takes a list of text documents and returns a matrix of TF-IDF features.

Parameters: - Text: A list of strings, where each string is a document.

Returns: - A sparse - `### Code: def similarity(doc1, doc2): return cosine_similarity([doc1, doc2])`

Docstring:

This function calculates the cosine similarity between two documents.

Parameters: - doc1: The first document. - doc2: The second document.

Returns: The cosine similarity between the two documents. - `### Code: def checkplagiarism():
global s_vectors for studenta, textvectora in s_vectors: newvectors = s_vectors.copy() currentindex =
newvectors.index((studenta, textvectora)) del newvectors[currentindex] for studentb, textvectorb
in newvectors: simscore = similarity(textvectora, textvectorb)[0][1] studentpair =
sorted((studenta, studentb)) score = (studentpair[0], studentpair[1], simscore)
plagiarismresults.add(score) return plagiarismresults`

Docstring:

This function checks for plagiarism in a list of text vectors. It does this by comparing each text vector with every other text vector in the list. The similarity between the two text vectors is calculated using the `similarity` function

Code Quality

Tool: `pylint`

Issues: 0`

```
text ***** Module tmpg6za54jt C:  
\Users\nmoha\AppData\Local\Temp\tmpg6za54jt.py:21:8: W0602: Using  
global for 's_vectors' but no assignment is done (global-  
variable-not-assigned)
```

Conclusion

This codebase represents a machine learning context in which text documents are compared using the cosine of the angle between their vectors. The cosine similarity is a measure of similarity between two vectors, and its value ranges between -1 and 1. A value of 1 indicates that the vectors are identical, 0 indicates they are orthogonal, and -1 indicates they are totally dissimilar. The cosine similarity is a common technique used for document classification and text mining.