lab2_count_vectorization

Harito ID

2025-09-16

Lab 2: Count Vectorization

Objective

To represent text documents as numerical vectors. You will implement the Bag-of-Words model using a `CountVectorizer`. This component is fundamental for using text data in machine learning models.

This lab will reuse the `Tokenizer` from Lab 1.

Task 1: Vectorizer Interface

In `src/core/interfaces.py`, define a new abstract base class for a `Vectorizer`. It should have the following methods:

1. `fit(self, corpus: list[str])`: Learns the vocabulary from a list of documents (corpus).
2. `transform(self, documents: list[str]) -> list[list[int]]`: Transforms a list of documents into a list of count vectors based on the learned vocabulary.
3. `fit_transform(self, corpus: list[str]) -> list[list[int]]`: A convenience method that performs `fit` and then `transform` on the same data.

Task 2: CountVectorizer Implementation

1. **Create the file:** `src/representations/count_vectorizer.py`.
2. **Implement the `CountVectorizer` class:**
    - It should inherit from the `Vectorizer` interface.
    - The constructor `__init__(self, tokenizer: Tokenizer)` should accept a `Tokenizer` instance (from Lab 1).
    - It should have an attribute `vocabulary_` (a `dict[str, int]`) to store the word-to-index mapping.
3. **Implement the `fit` method:**
    - Initialize an empty set to hold unique tokens.
    - Iterate through each document in the corpus.
    - For each document, use the provided `tokenizer` to get a list of tokens.
    - Add all tokens to the set to collect a unique vocabulary.

- After processing all documents, create the `vocabulary_` dictionary by mapping each unique token to a unique integer index (e.g., by sorting the set and assigning indices).

4. **Implement the `transform` method:**
   - For each document in the input list:
     - Create a zero vector with a length equal to the size of the `vocabulary_`.
     - Tokenize the document.
     - For each token, if it exists in the `vocabulary_`, increment the count at the corresponding index in the zero vector.
   - Return the list of resulting vectors.

Evaluation

- Create a new test file: `test/lab2_test.py`.
- In this file, instantiate your `RegexTokenizer` from Lab 1.
- Instantiate your `CountVectorizer` with the tokenizer.
- Define a sample corpus:

```
corpus = [
    "I love NLP.",
    "I love programming.",
    "NLP is a subfield of AI."
]
```

- Use `fit_transform` on the corpus and print the learned vocabulary and the resulting document-term matrix (the list of vectors).