CL-II 2 IR

July 21, 2025

Implement a program for retrieval of documents using inverted files.

[1]:
```
'''NAME:Aher Swami Sandip
ROLL NO.01
COURSE: AI&DS
CLASS: BE
SUB:Computer Laboratory-II (Information Retrival)'''
```

[1]:
```
class InvertedIndex:
    def __init__(self):
        # Initialize an empty dictionary to store the inverted index
        self.index = {}

    def add_document(self, doc_id, document):
        # Tokenize the document into words
        words = document.split()
        for word in words:
            word = word.lower()  # Convert to lowercase for case-insensitive␣
 ↪search
            if word not in self.index:
                self.index[word] = []
            if doc_id not in self.index[word]:
                self.index[word].append(doc_id)

    def query(self, words):
        # Tokenize the query into words
        words = words.split()
        results = set()
        for word in words:
            word = word.lower()  # Convert to lowercase for case-insensitive␣
 ↪search
            if word in self.index:
                if not results:
                    results = set(self.index[word])
                else:
                    results &= set(self.index[word])
        return list(results)
```

```
    def display_index(self):
        # Display the contents of the inverted index
        for word, doc_ids in self.index.items():
            print(f"{word}: {doc_ids}")


# Example usage
documents = {
    1: "The quick brown fox jumps over the lazy dog",
    2: "Never jump over the lazy dog quickly",
    3: "Brown foxes are quick and jump high"
}

# Create an instance of the InvertedIndex
inverted_index = InvertedIndex()

# Add documents to the index
for doc_id, content in documents.items():
    inverted_index.add_document(doc_id, content)

# Display the inverted index
inverted_index.display_index()

# Query the index
query = "quick fox"
results = inverted_index.query(query)
print(f"Documents containing '{query}': {results}")
```

```
the: [1, 2]
quick: [1, 3]
brown: [1, 3]
fox: [1]
jumps: [1]
over: [1, 2]
lazy: [1, 2]
dog: [1, 2]
never: [2]
jump: [2, 3]
quickly: [2]
foxes: [3]
are: [3]
and: [3]
high: [3]
Documents containing 'quick fox': [1]
```

```
[ ]:
```