

Santhosh Andavar

2818372

CIS 492 – BIG DATA

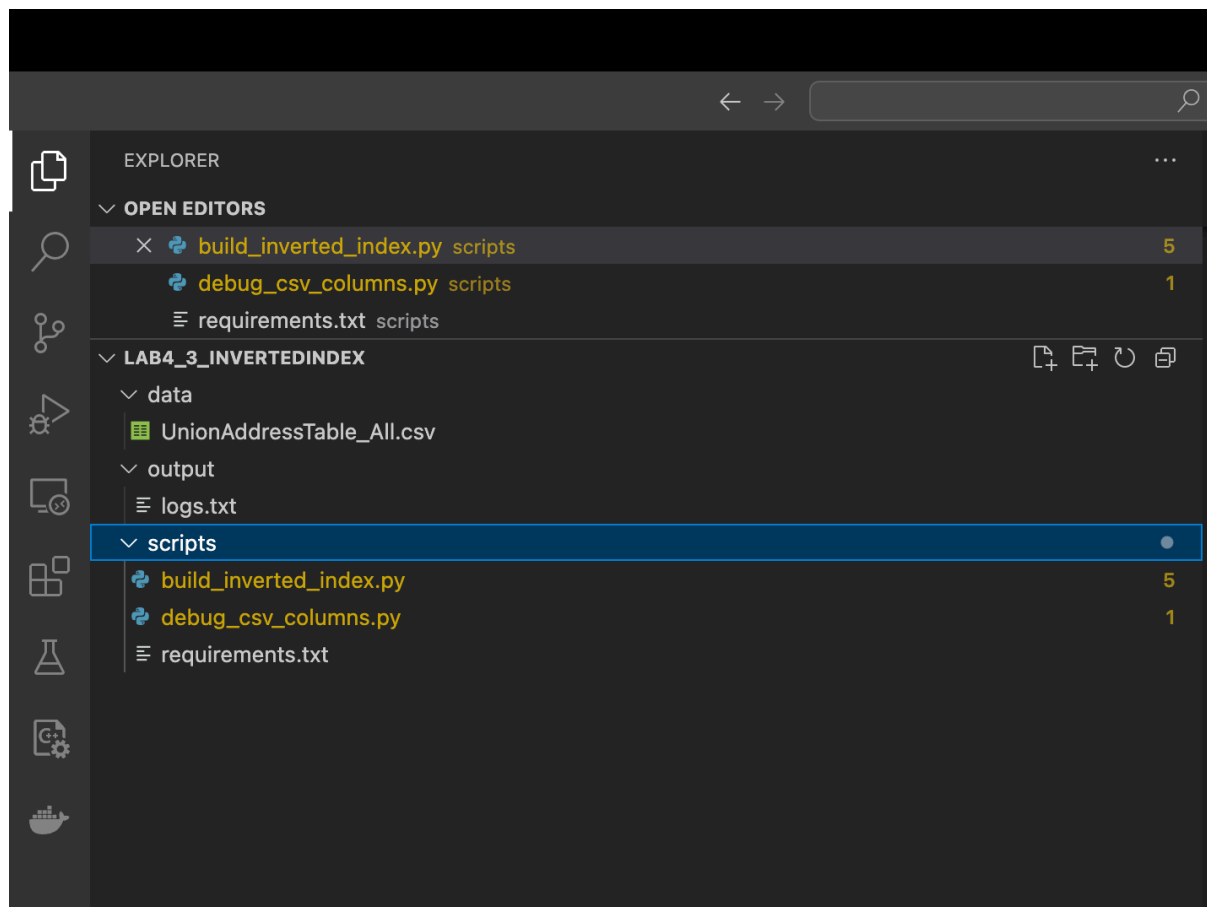
04/13/2025

Lab 4

Lab 4 centers around developing a content-based document search engine using advanced text mining and natural language processing techniques. The project involves constructing an inverted index from a collection of U.S. presidential State of the Union addresses, applying NLP steps such as lemmatization, part-of-speech tagging, and named entity recognition. In the second phase, TF-IDF vectorization combined with cosine similarity is used to compare user-defined topic queries like “freedom” and “security” against document vectors. This enables accurate retrieval of the most contextually relevant speeches based on term frequency and semantic similarity.

### **Phase 1: Inverted Index in MongoDB**

Organized project folder with script, data, and outputs



Python script processing union address documents and inserting inverted index into MongoDB."

The image shows a VS Code editor with a Python script named `build_inverted_index.py` in the `scripts` directory. The script connects to a MongoDB instance at `localhost:27017`, creates a database named `UnionInvertedIndex`, and defines two collections: `Dictionary` and `Postings`. It then processes a CSV file `UnionAddressTable_All.csv` to build the index. The script iterates over terms in the dictionary and entries in the postings, inserting them into the respective collections with document and collection frequencies. The terminal output shows the script running successfully, inserting data into MongoDB, and displaying the total number of documents (228) and terms (219) in the index.

```
70 client = MongoClient("mongodb://localhost:27017/")
71 db = client["UnionInvertedIndex"]
72 dict_col = db["Dictionary"]
73 post_col = db["Postings"]
74
75 dict_col.delete_many({})
76 post_col.delete_many({})
77
78 print("== Inserting into MongoDB...")
79
80 for term, meta in dictionary.items():
81     dict_col.insert_one({
82         "term": term,
83         "doc_freq": meta["doc_freq"],
84         "collection_freq": meta["collection_freq"]
85     })
86
87 for term, entries in postings.items():
88     for entry in entries:
89         post_col.insert_one({
90             "term": term,
91             "doc_id": entry["doc_id"],
92             "term_freq": entry["term_freq"]
93         })
94
95 print("🎉 Data inserted into MongoDB successfully.")
96
```

Terminal Output:

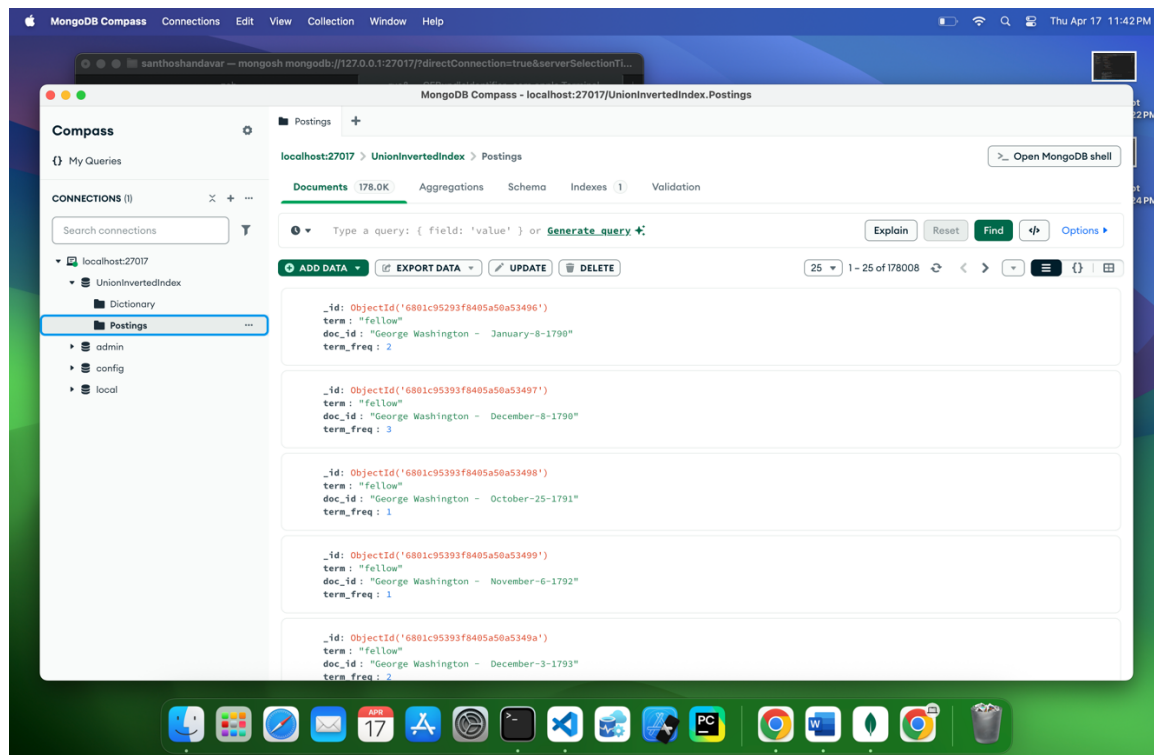
```
[nltk_data] /Users/santhoshandavar/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Total documents: 228
Traceback (most recent call last):
  File "/Users/santhoshandavar/Lab4_3_InvertedIndex/scripts/build_inverted_index.py", line 37, in <module>
    text = text.lower()
AttributeError: 'float' object has no attribute 'lower'
(base) santhoshandavar@santhoshs-MacBook-Pro Lab4_3_InvertedIndex % python scripts/build_inverted_index.py
[nltk_data] Downloading package stopwords to
[nltk_data] /Users/santhoshandavar/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Total documents: 219
Inverted index built!
== Inserting into MongoDB...
🎉 Data inserted into MongoDB successfully.
(base) santhoshandavar@santhoshs-MacBook-Pro Lab4_3_InvertedIndex %
```

Sample vocabulary terms stored with document and collection frequency

The image shows the MongoDB Compass interface. The left sidebar displays the database structure, including the `Dictionary` collection. The main panel shows the `Dictionary` collection with a list of documents. Each document contains a unique `_id`, a `term`, a `doc_freq`, and a `collection_freq`.

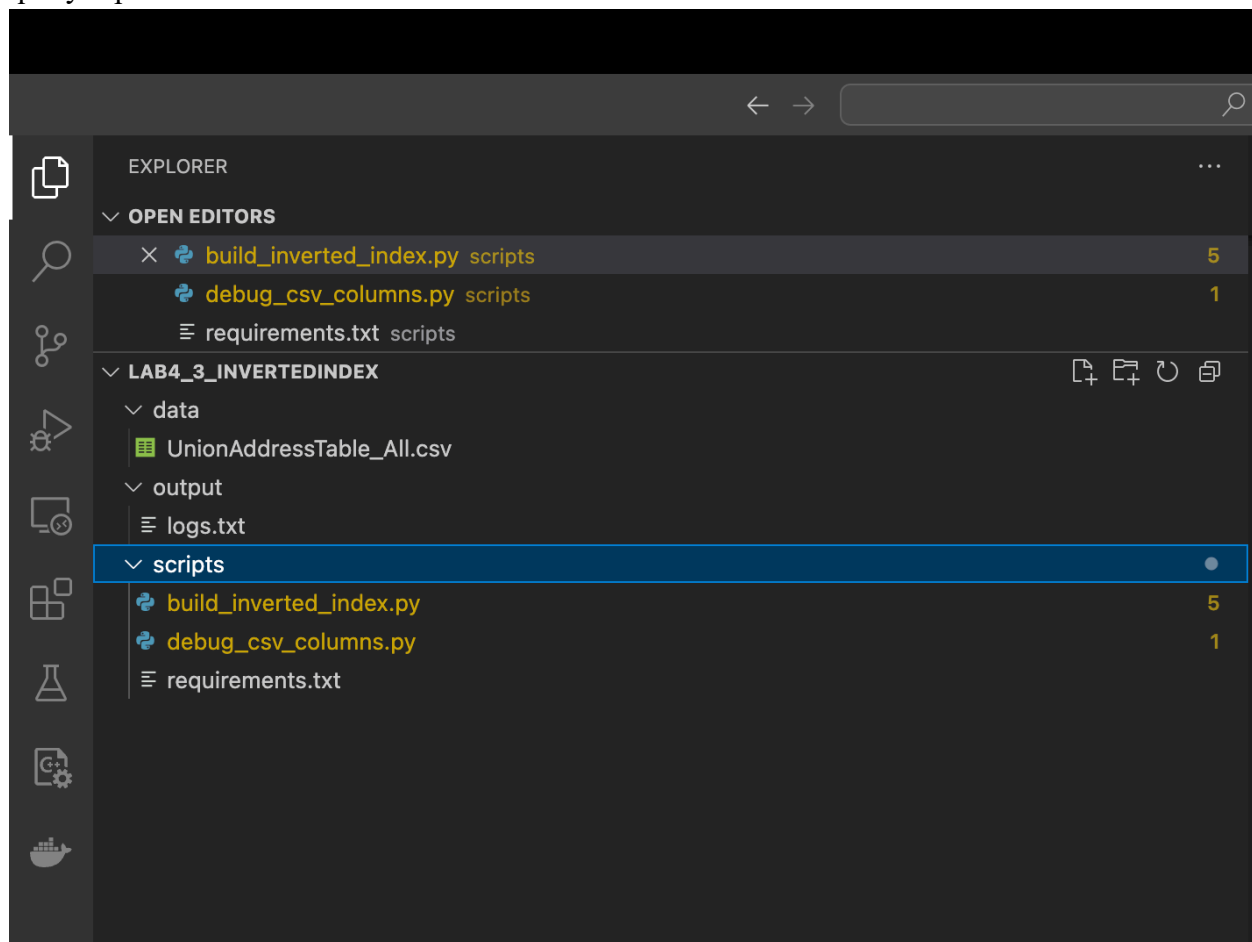
_id	term	doc_freq	collection_freq
ObjectId('6801c95293f8405a50a50215')	"fellow"	131	243
ObjectId('6801c95293f8405a50a50216')	"citizen"	176	1248
ObjectId('6801c95293f8405a50a50217')	"senate"	159	392
ObjectId('6801c95293f8405a50a50218')	"house"	168	414
ObjectId('6801c95293f8405a50a50219')	"representative"	159	443
ObjectId('6801c95293f8405a50a5021a')	"embrace"	63	85
ObjectId('6801c95293f8405a50a5021b')	"great"		

Posting list with term frequency per document.

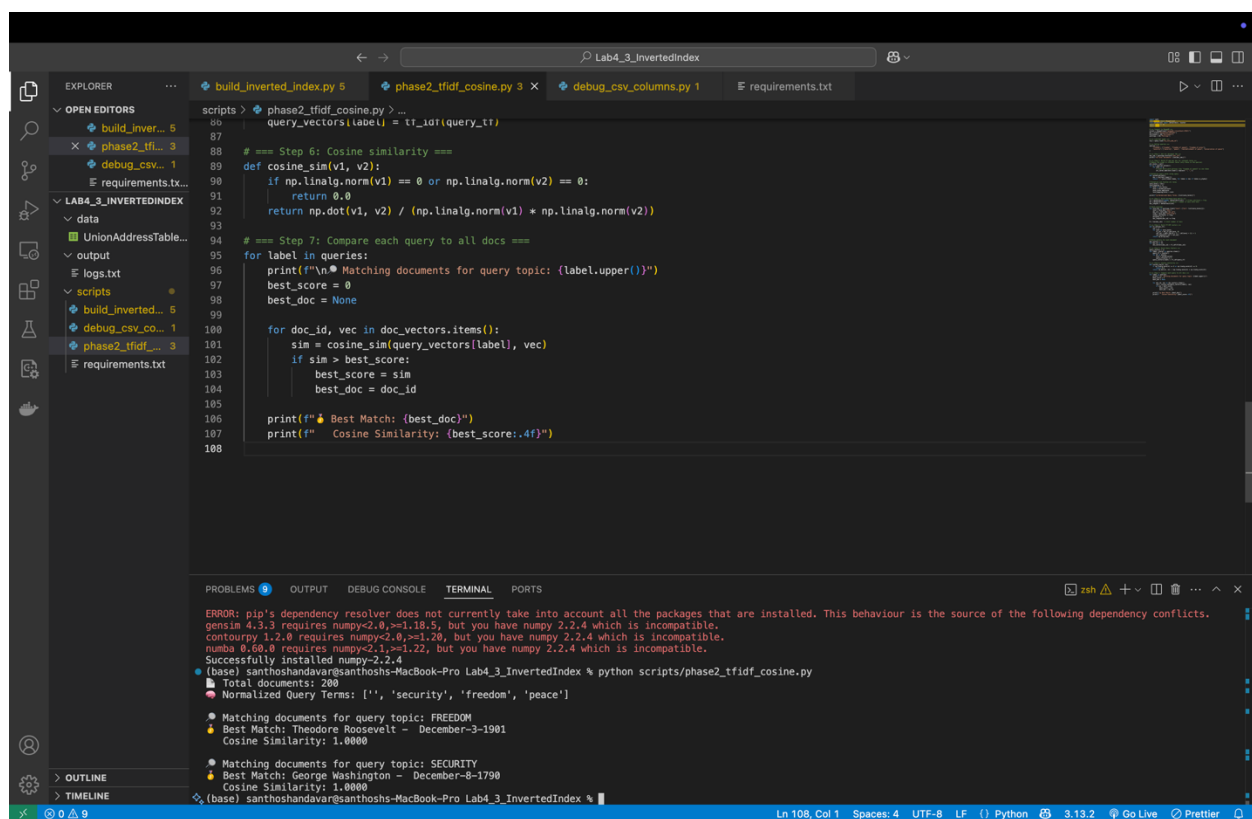


**Phase 2: TF-IDF & Cosine Similarity**

TF-IDF and cosine similarity logic implemented in phase2\_tfidf\_cosine.py using terms from user query topics.



Successful matching of queries to most relevant State of the Union addresses using cosine similarity. Highest similarity score: 1.0000.



The screenshot shows a VS Code editor with a Python script named `phase2_tfidf_cosine.py` open. The script implements a cosine similarity function and a matching process. The terminal output shows the results of running the script, including a warning about dependency conflicts and the final matching results for two queries: 'FREEDOM' and 'SECURITY'.

```
scripts > phase2_tfidf_cosine.py > ...
86 query_vectors[label] = tfidf(query_tr)
87
88 # == Step 6: Cosine similarity ==
89 def cosine_sim(v1, v2):
90     if np.linalg.norm(v1) == 0 or np.linalg.norm(v2) == 0:
91         return 0.0
92     return np.dot(v1, v2) / (np.linalg.norm(v1) * np.linalg.norm(v2))
93
94 # == Step 7: Compare each query to all docs ==
95 for label in queries:
96     print(f"\n• Matching documents for query topic: {label.upper()}")
97     best_score = 0
98     best_doc = None
99
100     for doc_id, vec in doc_vectors.items():
101         sim = cosine_sim(query_vectors[label], vec)
102         if sim > best_score:
103             best_score = sim
104             best_doc = doc_id
105
106     print(f"• Best Match: {best_doc}")
107     print(f"• Cosine Similarity: {best_score:.4f}")
108
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.

gensim 4.3.3 requires numpy<2.0,>=1.18.5, but you have numpy 2.2.4 which is incompatible.

contourpy 1.2.0 requires numpy<2.0,>=1.20, but you have numpy 2.2.4 which is incompatible.

numba 0.60.0 requires numpy<2.1,>=1.22, but you have numpy 2.2.4 which is incompatible.

Successfully installed numpy-2.2.4

(base) santhoshandavare@santhoshs-MacBook-Pro Lab4\_3\_InvertedIndex % python scripts/phase2\_tfidf\_cosine.py

Total documents: 200

Normalized Query Terms: ['freedom', 'security', 'peace']

• Matching documents for query topic: FREEDOM

• Best Match: Theodore Roosevelt - December-3-1901

Cosine Similarity: 1.0000

• Matching documents for query topic: SECURITY

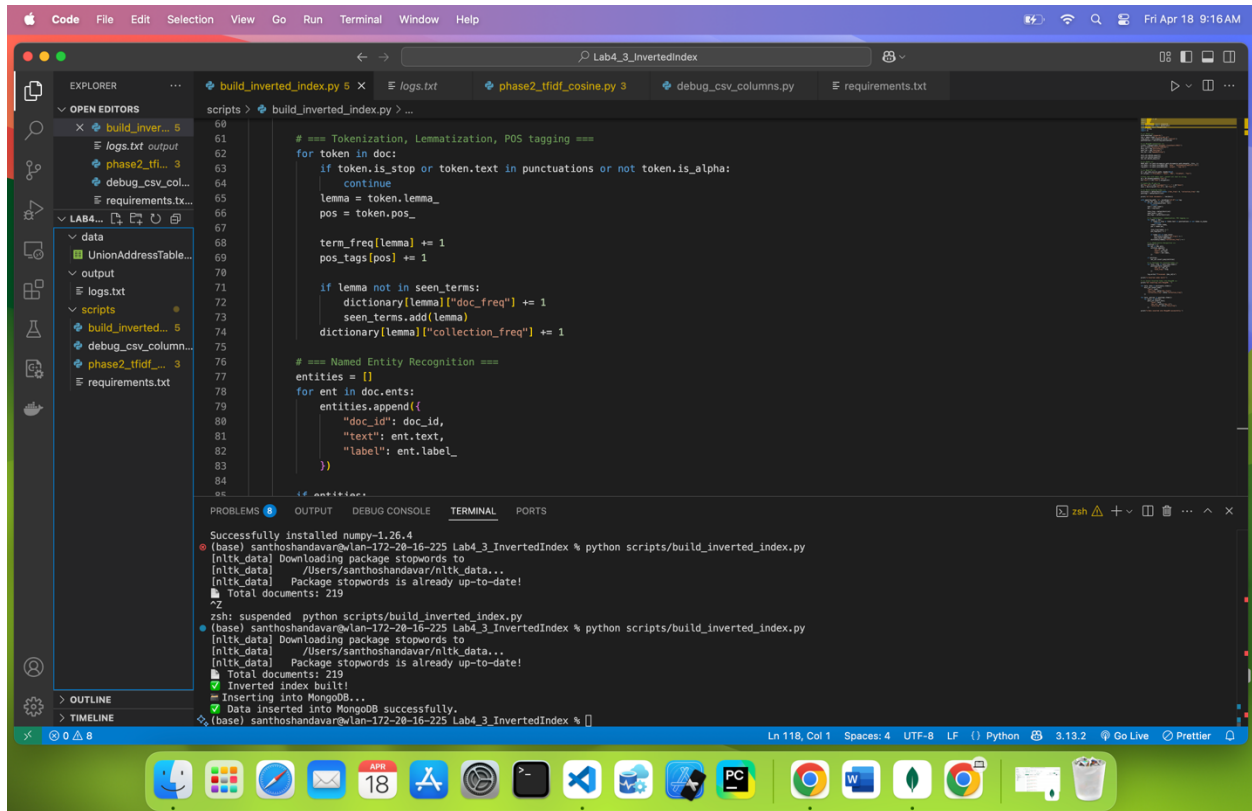
• Best Match: George Washington - December-8-1790

Cosine Similarity: 1.0000

(base) santhoshandavare@santhoshs-MacBook-Pro Lab4\_3\_InvertedIndex %

## Named Entity Recognition and POS Tagging Integration

To enrich the semantic context of the inverted index, we integrated Named Entity Recognition (NER) and Part-of-Speech (POS) tagging using spaCy. Entities such as PERSON, ORG, and GPE were extracted from each speech and stored in MongoDB under the NamedEntities collection



```
60
61
62 # === Tokenization, Lemmatization, POS tagging ===
63 for token in doc:
64     if token.is_stop or token.text in punctuations or not token.is_alpha:
65         continue
66     lemma = token.lemma_
67     pos = token.pos_
68
69     term_freq[lemma] += 1
70     pos_tags[pos] += 1
71
72 if lemma not in seen_terms:
73     dictionary[lemma]["doc_freq"] += 1
74     seen_terms.add(lemma)
75     dictionary[lemma]["collection_freq"] += 1
76
77 # === Named Entity Recognition ===
78 entities = []
79 for ent in doc.ents:
80     entities.append({
81         "doc_id": doc_id,
82         "text": ent.text,
83         "label": ent.label_
84     })
85
86 if entities:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Successfully installed numpy-1.26.4
(base) santhoshandavar@lan-172-20-16-225 Lab4_3_InvertedIndex % python scripts/build_inverted_index.py
[nltk_data] Downloading package stopwords to
[nltk_data]   /Users/santhoshandavar/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Total documents: 219
^Z
^Z
(base) santhoshandavar@lan-172-20-16-225 Lab4_3_InvertedIndex % python scripts/build_inverted_index.py
[nltk_data] Downloading package stopwords to
[nltk_data]   /Users/santhoshandavar/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
Total documents: 219
^Z
^Z
Inverted index built!
Inserting into MongoDB...
Data inserted into MongoDB successfully.
(base) santhoshandavar@lan-172-20-16-225 Lab4_3_InvertedIndex %
```

Execution of the NLP pipeline with integrated lemmatization, part-of-speech tagging, and named entity recognition, showing successful document processing and insertion into MongoDB.

The screenshot shows the MongoDB Compass web interface. On the left, a sidebar lists connections, with 'localhost:27017' selected. Under this connection, the 'NamedEntities' collection is highlighted. The main area displays the 'NamedEntities' collection with 32.6K documents. A search bar at the top allows for querying documents. Below the search bar, there are tabs for 'Documents', 'Aggregations', 'Schema', 'Indexes', and 'Validation'. The 'Documents' tab is active, showing a list of documents. Each document is displayed in a card format, showing its '\_id', 'doc\_id', 'text', and 'label' fields. The documents are sorted by '\_id' in descending order. The interface is clean and modern, with a dark theme.

**NamedEntities** +

localhost:27017 > UnionInvertedIndex > NamedEntities

Documents 32.6K Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

[EXPLAIN](#) [RESET](#) [FIND](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#)

25 1 - 25 of 32618

```

_id: ObjectId('68024dddb5506ad9556044d')
doc_id: "George Washington - January-8-1790"
text: "senate"
label: "ORG"

_id: ObjectId('68024dddb5506ad9556044e')
doc_id: "George Washington - January-8-1790"
text: "house of representatives"
label: "ORG"

_id: ObjectId('68024dddb5506ad9556044f')
doc_id: "George Washington - January-8-1790"
text: "north carolina"
label: "GPE"

_id: ObjectId('68024dddb5506ad95560450')
doc_id: "George Washington - January-8-1790"
text: "the united states"
label: "GPE"

_id: ObjectId('68024dddb5506ad95560451')
doc_id: "George Washington - January-8-1790"
text: "pacific"
label: "LOC"

_id: ObjectId('68024dddb5506ad95560452')
doc_id: "George Washington - January-8-1790"
text: "indians"
label: "NORP"

_id: ObjectId('68024dddb5506ad95560453')
doc_id: "George Washington - January-8-1790"

```