

NLP Task

aim:

To use NLP libraries to perform part of speech tagging and to develop an information retrieval system to search and rank documents based on relevance

code:

```
import spacy
```

```
nlp = spacy.load("en_core_web_sm")
```

```
text = "AI - driven platforms personalize learning paths and help student grasp concept faster"
```

```
doc = nlp(text)
```

```
for token in doc:
```

```
print(f" {token.text}: {token.pos_} {token.tag_}")
```

```
from sklearn.feature_extraction import TfidfVectorizer  
from sklearn.metrics.pairwise import cosine_similarity
```

documents = [

" AI tools analyze student performance and provide real time feedback",

" Intelligent tutoring systems adapt to each student's learning style ",

" AI helps automate grading and administrative tasks in schools.]

Output:

AI	→ PROPN
-	→ PUNCT
drive	→ VERB
platform	→ NOUN
personalize	→ VERB
path	→ NOUN
and	→ CONJ
help	→ VERB
faster	→ ADV
-	→ PUNCT

Top relevant documents:

Score: 0.16 → AI helps automate grading and administrative tasks in schools

Score: 0.10 → intelligent tutoring systems adapt to each student's learning style

Score: 0.09 → chatbots assist students with answering questions any time of day

Score: 0.06 → virtual classrooms powered by AI enhance student engagement

Score: 0.05 → A

query = "How does AI support students in learning?"

corpus = documents + [query]

vectorizer = TfidfVectorizer()

tfidf_matrix = vectorizer.fit_transform(corpus)

similarities = cosine_similarity(tfidf_matrix[-1],
tfidf_matrix[:-1]).flatten

ranked_docs = sorted(zip(similarities, documents),
reverse=True)

print("Top relevant documents : \n")

for score, doc in ranked_docs:

print(f"Score : {score} → doc {doc}")

Result:

This program has been executed
successfully.