

Ex no 5
S18125

NLP task: Part of speech Tagging
and Information Retrieval.

Aim:

To develop an information retrieval System
to search and rank documents based on
relevance.

Program code:

```
pip install spacy  
python -m spacy download en_core_web_sm  
import spacy  
nlp = spacy.load("en_core_web_sm")  
text = "AI - driven platform personalize Learning  
path and help students grasp. concepts  
factor".
```

doc = nlp(text)

for token in doc:

from sklearn.feature_extraction.text

documents = [{"AI tools analyze students

Performance

tutoring Systems adapt to each students
grading and administrative tasks in schools,

"chatbots assist students with answering
questions any time of day", "virtual



AI-based teaching tools; slot 9:20 2 on 23
Learner performance analysis 2:10:2

Query

match learner performance no fault or
no benefit in slot share this same slot

AI -> Propn

- -> Punct

drive -> verb edge taken 9:2

platform -> noun

personalize -> verb edge taken

("use new tool") tool. page + 9:2

Learning -> verb

personalizing matching words -IA "about"

Paths -> noun

2+ goals) George stuck flat long time

and -> cons

(tool) g/n = lab

Top relevant documents:

Score: 0.16 -> AI helps automate

grading and administrative tasks

in schools.

Score: 0.05 -> AI tools analyze

student performance and

provide real time feedback.

learner's job to analyze work



Shot on OnePlus

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

Corpus = documents \rightarrow [query]

vectorizing = $\text{tfidf vectorizer}()$

$t + \text{rdg_matrix} = \text{vectorizer}.fit\text{-transform}(\text{coefs})$

similarity = Cosine-similarity

2. Groundwater flow
Groundwater

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

```
Print("In Top 500 relevant documents:\n")
```

for Score, doc in ranked-docs:

printf("Score: %d\n") %> {doc}')

Results:

Thus the information retrieval system
to search and rank documents based on
shot on OnePlus has been developed successfully.



Shot on OnePlus