EXP.NO:5
5/8/25

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 platforms personalize learning
       paths and help students grasp concepts
       faster."
doc = nlp(text)
for token in doc:
    Print (f"{token.text:15} → {token.pos_}")
from sklearn.feature_extraction.text 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.",
"Chatbots assist students with answering
questions any time of day.", "Virtual
classrooms powered by AI enhance students."]
```

OUTPUT:

AI → PROPN
- → PUNCT
driven → VERB
platforms → NOUN
Personalize → VERB
learning → VERB
paths → NOUN
and → CCONJ
help → VERB

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.

```
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("\nTop relevant documents:\n")
for score, doc in ranked_docs:
    Print(f"Score: {score:.2f} → {doc}")
```

RESULT:

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