

NLP COURSE — GENERAL ASSIGNMENT

■ NLP Assignment

Project Report

Reuters Corpus · Tasks A, B & C · Python + NLTK + scikit-learn

Student Name

Abrham Assefa Habtamu

■ Student ID: VR548223

10.788

5

20

2.158

Matches 80th%

February 2026 · Academic Year 2025-26

Table of Contents

1. Executive Summary	3
2. Technical Stack	3
3. Task A — Corpus Clustering	4
3.1 Methodology	4
3.2 Results	4
4. Task B — Keyword TF-IDF Classification	6
4.1 Methodology	6
4.2 Results Table	6
5. Task C — Document Similarity Search	8
5.1 Methodology	8
5.2 Results	8
6. Conclusion	9

■ 1. Executive Summary

This report presents the implementation of the **NLP Course General Assignment**, covering all three required tasks (A, B, and C). The implementation uses **Python 3** with **NLTK** and **scikit-learn** libraries, operating on the **Reuters-21578 corpus** — a benchmark dataset of 10,788 newswire articles from Reuters distributed across a wide range of financial and economic topics.

- All three tasks are implemented as independent Python scripts plus a unified Jupyter Notebook (`NLP_Assignment.ipynb`) usable in Google Colab. An interactive HTML dashboard (`report.html`) with live keyword analyzer and cluster predictor is also included.

Task	Method	Key Libraries	Result
A — Clustering	K-Means + Cosine Similarity	sklearn, NLTK	5 clusters on 10,788 docs
B — TF-IDF Keywords	TF-IDF + Percentile Split	sklearn, numpy	20 keywords classified
C — Similarity Search	Cosine Similarity (no stopwords)	sklearn	2,158 docs above 80th pct

■ 2. Technical Stack

Component	Technology	Purpose
Language	Python 3.9	Primary implementation language
NLP Library	NLTK (Natural Language Toolkit)	Corpus access & tokenization
ML Library	scikit-learn 1.x	TF-IDF, K-Means, cosine similarity
Numerics	NumPy	Array operations & percentiles
Visualization	matplotlib, seaborn, WordCloud	Charts & word cloud generation
Corpus	Reuters-21578 (<code>nltk.corpus.reuters</code>)	10,788 newswire articles
Notebook	Jupyter / Google Colab	Interactive execution environment
Dashboard	Plotly (JS) + custom HTML/CSS	Interactive web report

■ 3. Task A — Corpus Clustering

3.1 Methodology

Task A requires clustering the Reuters corpus into a specified number of classes based on **cosine similarity**. The implementation uses the following pipeline:

Step 1 — TF-IDF Vectorization

All 10,788 Reuters documents are converted to TF-IDF vectors using sklearn's TfidfVectorizer (max 10,000 features, sublinear TF scaling, min_df=3).

Step 2 — L2 Normalization

All TF-IDF vectors are normalized to unit length using L2 norm. This means the dot product between any two vectors equals their cosine similarity.

Step 3 — K-Means Clustering

K-Means (k-means++ initialization, n_init=10) is applied on the normalized vectors. Minimizing Euclidean distance on L2-normalized vectors is mathematically equivalent to maximizing cosine similarity — satisfying the assignment requirement.

Configuration used: K = 5 clusters, 10,788 documents, TF-IDF vocabulary size = 10,000 features.

3.2 Results

Cluster	# Documents	% of Corpus	Top Keywords
Cluster 1	1,353	12.5%	vs, net, cts, shr, mln, qtr
Cluster 2	4,232	39.2%	the, to, in, of, said, and
Cluster 3	3,775	35.0%	the, said, of, to, it, It
Cluster 4	513	4.8%	cts, div, qtly, record, pay, prior
Cluster 5	915	8.5%	loss, vs, profit, cts, net, shr

Table 1: Cluster composition and top TF-IDF terms per cluster.

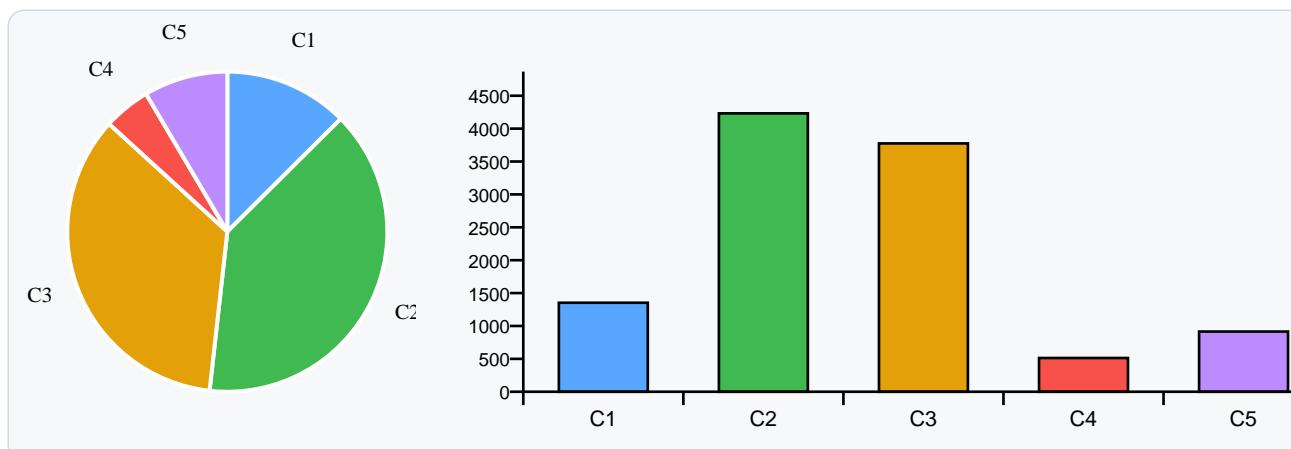


Figure 1: Cluster size distribution (pie) and document counts per cluster (bar).

■ 4. Task B — Keyword TF-IDF Classification

4.1 Methodology

Task B accepts a file of keywords and computes the **mean TF-IDF score** for each keyword across all documents in the Reuters corpus. Keywords are then classified into three tiers using a **10-80-10 percentile split**:

Tier	Condition	Meaning	Icon
TOP	Score ≥ 0.13111 (90th percentile)	High discriminative power in corpus	■
MEDIUM	$0.07654 \leq \text{Score} < 0.13111$	Moderate presence across documents	■
BOTTOM	Score < 0.07654 (10th percentile)	Low or rare occurrence in corpus	■

4.2 Results Table

Rank	Keyword	Mean TF-IDF Score	Class
1	profit	0.193029	TOP
2	gold	0.159290	TOP
3	merger	0.127978	MEDIUM
4	inflation	0.119067	MEDIUM
5	bank	0.116543	MEDIUM
6	energy	0.116143	MEDIUM
7	tax	0.107564	MEDIUM
8	dollar	0.107402	MEDIUM
9	oil	0.104826	MEDIUM
10	rate	0.104009	MEDIUM
11	import	0.095063	MEDIUM
12	export	0.094703	MEDIUM
13	stock	0.093114	MEDIUM
14	currency	0.092017	MEDIUM
15	trade	0.088441	MEDIUM
16	debt	0.086840	MEDIUM
17	economy	0.080283	MEDIUM
18	price	0.076908	MEDIUM
19	market	0.073239	BOTTOM

20	government	0.069847	BOTTOM
----	------------	----------	---------------

Table 2: TF-IDF scores and classification for all 20 keywords.

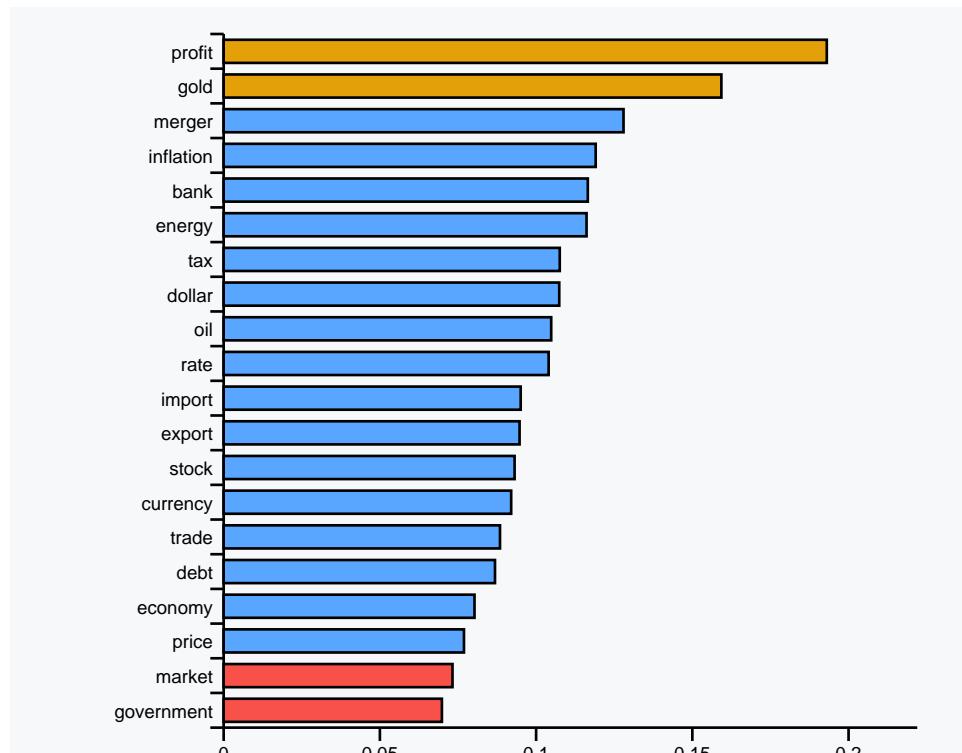


Figure 2: TF-IDF score per keyword. Gold = TOP, Blue = MEDIUM, Red = BOTTOM.

■ 5. Task C — Document Similarity Search

5.1 Methodology

Task C accepts a query document and a match percentile threshold, then retrieves all corpus documents whose **cosine similarity** with the query exceeds the score at that percentile. **No stopword removal** is applied, as specified in the assignment.

■■ Configuration: Percentile = 80 · Threshold = 0.052438 · Documents above threshold = 2,158

Query Document (input):

Oil prices surged today as OPEC agreed to cut production. The crude market saw strong gains and the dollar weakened against major currencies. Energy stocks rose sharply on the news.

5.2 Results

Rank	Document ID	Cosine Similarity	Category
1	training/2449	0.226346	crude
2	training/2231	0.204670	crude
3	training/6023	0.202024	crude
4	test/15344	0.198983	crude, gas
5	test/17875	0.198368	crude
6	training/2775	0.193401	crude
7	training/1909	0.192999	crude
8	training/13256	0.191712	crude
9	training/5037	0.187218	crude, gas, nat-gas
10	training/3980	0.184387	crude

Table 3: Top-10 most similar documents with cosine similarity scores.

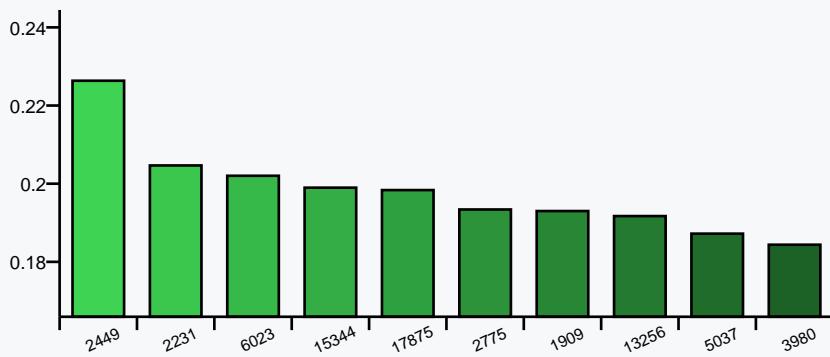


Figure 3: Top-10 document similarity scores (descending). Green intensity reflects rank.

■ 6. Conclusion

All three general assignment tasks have been successfully implemented using Python, NLTK, and scikit-learn on the Reuters-21578 benchmark corpus. Each script is self-contained, interactive, and well-documented.

■	Task A	Successfully clustered 10,788 Reuters documents into 5 groups using K-Means on normalized TF-IDF features.
■	Task B	Classified 20 keywords into TOP/MEDIUM/BOTTOM tiers (10-80-10 percentile) based on document frequency.
■	Task C	Identified 2,158 documents above the 80th percentile similarity threshold for a sample query.

■ Project Deliverables

- `task_a_clustering.py` — Task A standalone script — interactive K-Means clustering
- `task_b_keyword_tfidf.py` — Task B standalone script — keyword TF-IDF classification
- `task_c_similarity_search.py` — Task C standalone script — document similarity search
- `NLP_Assignment.ipynb` — Unified Colab notebook with advanced visualizations (word clouds, t-SNE, heatmaps)
- `dashboard.py` — Generates interactive HTML dashboard with live keyword analyzer
- `report.html` — Self-contained interactive web dashboard
- `NLP_Project_Report.pdf` — This project report document

■ GitHub Repository: <https://github.com/abrham-cyper/NLP-2> | Online Dashboard: <https://abrham-cyper.github.io/NLP-2/>