Introduction to LLM

Practice Session 5

Word Representation II

Exam Format and Relation to PS

- No Need to Memorize Libraries.
  - You are not expected to remember exact library names or function calls.

- Implementation Expectations.
  - Unlike PS/HW where full models are built from scratch, the exam may ask you to implement only a small part or component of a model.

- Focus on Understanding.
  - Know how each model works, the steps involved, and the expected inputs/outputs.
  - This will help you write clear pseudo-code when needed.

- Frank may release an exam template so you can get familiar with the format.
  - Pay attention to exam hints mentioned by Frank.

Pseudo-code for Building TF-IDF Matrix (enough for exam)

```
compute_tfidf_matrix(corpus):

    # Step 1 — Preprocessing
    tokenize all documents (split on white space)
    vocab = set of all unique terms (tokens) across all documents
    N = number of documents

    # Step 2 — Compute DF for each term
    for each term in vocab:
        DF[term] = number of documents where term appears at least once

    # Step 3 — Compute IDF for each term
    for each term in vocab:
        if DF[term] > 0:
          IDF[term] = log( N / DF[term] )
        else:
          IDF[term] = 0

    # Step 4 — Compute TF-IDF for each document
    initialize TFIDF matrix of size (N × |vocab|)
    for each document d_i:
        for each term t in vocab:
            TF = count of t in document d_i
            TFIDF[i][t] = TF * IDF[t]

    return TFIDF
```

```
corpus =
   [
      "the cat sat on the mat",
      "the dog ran in the park",
      "cats and dogs are pets",
      "the park has many trees"
   ]
```

$tf_{t,d}$ = how often does term t appear in document d

$$\text{idf}(t) = log\,\frac{|D|}{df_t}$$

Pseudo-code (not real code, but still logically correct)

```
cosine_similarity_pseudo(a, b):
    dot  =  Σ over i of (a[i] * b[i])
    len_a =  sqrt( Σ over i of (a[i]^2) )
    len_b =  sqrt( Σ over i of (b[i]^2) )
    if len_a == 0 OR len_b == 0:
        return 0
    return dot / (len_a * len_b)
```

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

Timeline

PS1

Python & ML Foundations

- Train/Dev/Test
- Hyperparameters
- Loss Fn.

PS2

CL/NLP Basics

- Morphology
- Syntax
- PoS
- Parsing
- Semantics
- Lemma
- Ambiguity

PS3

Word Representation I:

- Basics
- Evaluation
- Inverted Index
- TF-IDF
- BM25
- LSA
- nDCG

PS4

Word Representation II:

- One-hot Encoding
- Cosine Similarity
- Word2Vec
- CBOW & Skip-gram
- GloVe

PS11

Dense Retrieval & Neural Reranking

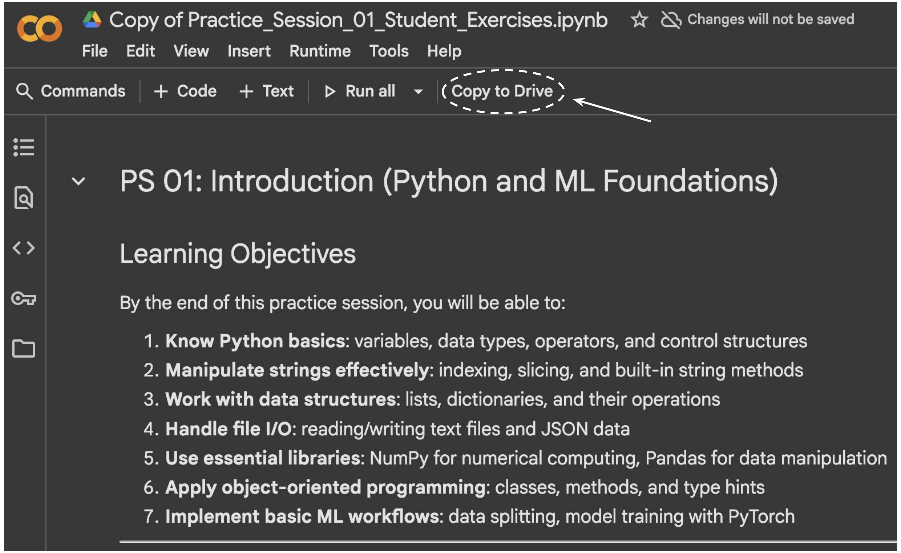
- Dense & Hybrid
- S-BERT
- Cross-Encoders

PS12

Question Answering & RAG

- Query Decomposition
- Metadata Filtering
- Self-RAG

PS4: Colab Notebook (Available on Moodle)



PS 01: Introduction (Python and ML Foundations)

Learning Objectives

By the end of this practice session, you will be able to:

1. **Know Python basics**: variables, data types, operators, and control structures
2. **Manipulate strings effectively**: indexing, slicing, and built-in string methods
3. **Work with data structures**: lists, dictionaries, and their operations
4. **Handle file I/O**: reading/writing text files and JSON data
5. **Use essential libraries**: NumPy for numerical computing, Pandas for data manipulation
6. **Apply object-oriented programming**: classes, methods, and type hints
7. **Implement basic ML workflows**: data splitting, model training with PyTorch

- https://colab.research.google.com/drive/1Q7jip-4fNGCUcaR9daywJXqrTTg3ILnt