# Introduction to LLM

## Practice Session 3

Word Representation I

# Lemmatization vs. Normalization

- Lemmatization reduces a word to its dictionary base form while keeping the same part of speech (POS).
  - – “runners” (plural NOUN) → “runner” (singular NOUN)
  - – “running” (VERB) → “run” (base VERB)
  - – “quickly” (ADV) → “quickly” (base ADV)

- Some pipelines use a looser heuristic called normalization, which can change POS to group similar meanings.
  - – “quickly” → “quick”
  - – “happily” → “happy”

- Key idea: Lemmatization is grammatically faithful. Normalization is semantically convenient.

# Mini POS Tagger Demo

**Mini POS Tagger – Features → Sparse Vector → POS Tag**

**The** **quick** **brown** **fox** **jumps** **over** **the** **lazy** **dog** **.**

```
Token 1/10: The
Predicted POS: DET

Human-readable features:
  - w=the
  - p3=the
  - s3=the
  - prev=<BOS>
  - next=quick
  - is_title
```

```
Vector dim: 27684
First 20 dims of this token's input vector:
  [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...]

Active feature indices (truncated):
  2: is_title
  5825: next=quick
  9646: p3=the
  10794: prev=<BOS>
  19530: s3=the
  26969: w=the
```

# Syntactic Ambiguity

- “The shooting of the hunters was terrible.”
  - Ambiguity: Were the hunters doing the shooting, or were the hunters the ones shot?
  - Parser behavior: The parse is the same for both readings.
    - “shooting” is the subject of “was” (shooting → nsubj → was)
    - “of the hunters” is just a PP modifying “shooting” (of → prep → shooting) and (hunters → pobj → of).
  - The parsing tree does not mark whether “hunters” are agents or victims. That role is semantic, not syntactic, so the dependency parse cannot disambiguate this case.

- Key idea:
  - Some ambiguities are solved by which word attaches to which head.
    - e.g., “Flying planes can be dangerous“.
      - (planes → dobj → flying), so it commits to “the act of flying planes is dangerous.”
      - (flying → amod → planes), so it commits to “as planes that fly”.
  - Some ambiguities survive parsing because both meanings share the same structure. Those require world knowledge or semantics, not just syntax.

# Clarifying “Basic vs Extended Lesk”

- Original Lesk (1986):
  - Compare glosses of all candidate senses with glosses of surrounding words’ senses; score by overlap between definitions. *(definitions ↔ definitions overlap)*
- Simplified Lesk (Algorithm explained in PS):
  - Compare sentence context directly to each sense’s gloss. *(definitions ↔ sentence context overlap)*
- Extended / Adapted Lesk (Banerjee & Pedersen, 2002):
  - Enrich each sense’s gloss with related synsets (hypernyms, hyponyms, etc.) to get more opportunities. *(expanded definitions ↔ sentence context overlap).*
- HW01 - Task 3
  - “***Basic Lesk”***:
    - Compare the context words in the sentence with the gloss (definition) of each candidate sense and choose the sense with the most word overlap. This is the ***Simplified Lesk*** algorithm.
  - “***Extended Lesk”***:
    - Same procedure, but all words are lemmatized first (so “running”, “runs” → “run”). This isolates the effect of ***lemmatization***.

# Level of Practice Session Tasks

- They are intentionally lightweight and guided.
  - You fill in small blanks rather than writing full systems.
- This makes you active participant: you modify and run code, not just watch me run code.
- Each task mirrors what we just explained live, but adds 1–2 new concepts.
  - By editing and executing the code yourself, you see how data flows and where to plug in new concepts.
  - We hopefully try to prepare you for HW which is more open and requires actual design decisions.
- We know not everyone comes from a CS-heavy disciplinaries, so we aim for an accessible middle ground for everyone.

# Timeline


**Python & ML Foundations**

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

**CL/NLP Basics**

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

**Word Representation I:**

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

**Word Representation II:**

- Statistical Methods
- Cosine Similarity
- Word2Vec
- CBOW & Skip-gram
- GloVe

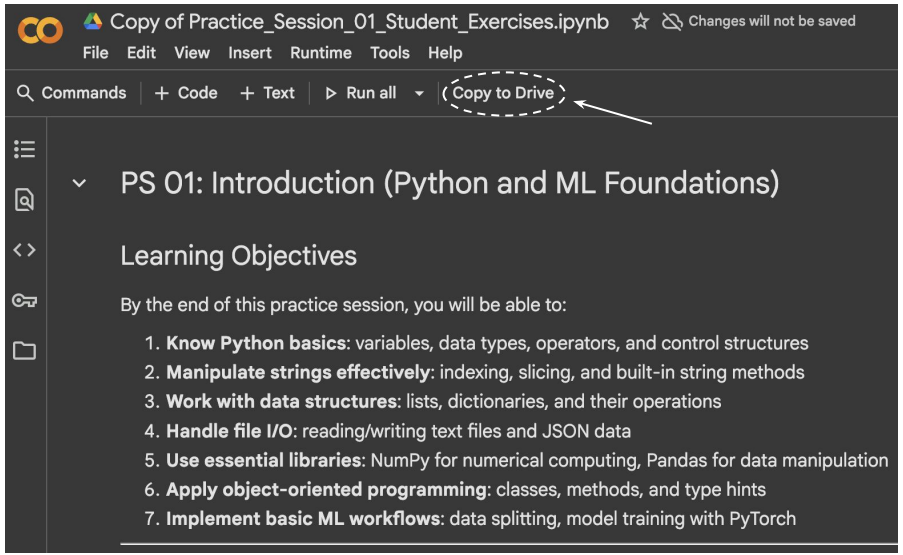
**Dense Retrieval & Neural Reranking**

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

**Question Answering & RAG**

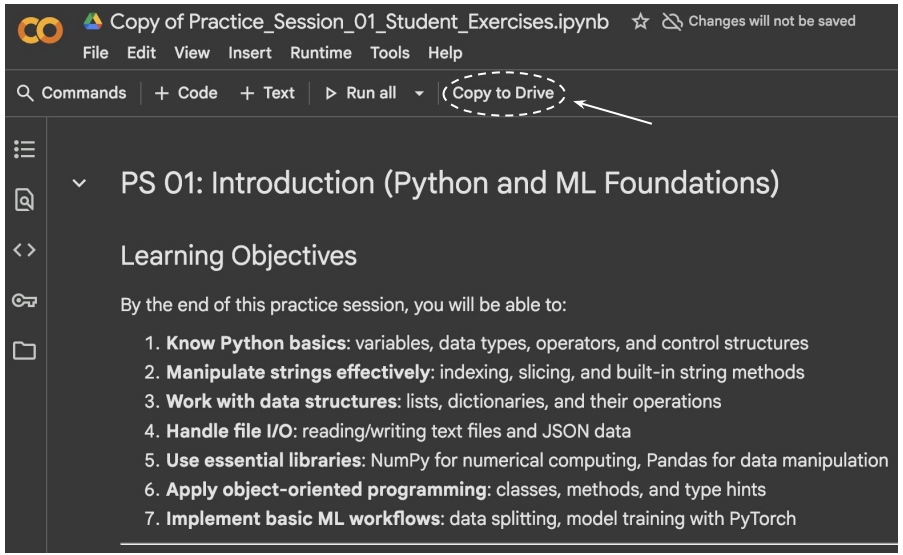
- Query Decomposition
- Metadata Filtering
- Self-RAG

# PS2: Colab Notebook (Available on Moodle)




- https://colab.research.google.com/drive/1EFXK8CyUVjg2n7Bx1_QOCG_6b3Jbxqib

# PS3: Colab Notebook (Available on Moodle)




- https://colab.research.google.com/drive/1IdxJCu6HOsoOJUXwnCJ6-jrF-ypjGL9G