UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Behavior Analysis Technologies

Session 4

Text Similarity

Applied Data Science

2024/2025

Text Similarity

- Text similarity is a **measure of how closely related two pieces of text are**, either based on their form (lexical similarity) or their meaning (semantic similarity).

- **Lexical Similarity:**
    - Based on surface forms (e.g., common words or n-grams);
    - Example: Comparing "quick brown fox" vs "quick blue fox"

- **Semantic Similarity:**
    - Focuses on meaning and context.
    - Example: "He bought a car" vs. "He purchased an automobile"

Lexical vs Semantic Similarity

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

Lexical	Semantic
Quick and Easy	Rich in meaning
Surface-level	Context-aware
Example: "big" vs "large" (not similar lexically)	Example: "big" vs "large" (semantically similar)

Measures of Lexical Similarity

- Lexical similarity measures how much the surface forms (i.e., words, characters) of two texts overlap.
- Quick and straightforward to implement.
- Useful for tasks like duplicate detection, spell checking, and clustering similar documents

Lexical Similarity Metrics

- Jaccard Similarity
- Cosine Similarity
- Edit Distance (Levenshtein Distance)

Jaccard Similarity

- Jaccard similarity:
  - Measures the **overlap between two sets of tokens** (e.g., words).
  - Values range from **0 (no overlap)** to **1 (complete overlap)**.

- Defined as the ratio of shared tokens and total tokens in two documents:

$$sim_{\text{Jaccard}}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \ ,$$

where X and y represent the terms that appear in documents d_1 and $_{d2}$, respectively.

Jaccard Similarity

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

- Jaccard similarity for term vector-based representations:

$$sim_{\text{Jaccard}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_i \mathbb{1}(x_i) \times \mathbb{1}(y_i)}{\sum_i \mathbb{1}(x_i + y_i)} ,$$

here $\mathbb{1}(x)$ is an indicator function (1 if x > 0 and 0 otherwise).

- Example:

	term 1	term 2	term 3	term 4	term 5
doc x	1	0	1	0	3
doc y	0	2	4	0	1

Table: Document-term vectors with term frequencies.

$$\mathbf{x} = \langle 1, 0, 1, 0, 3 \rangle \qquad \mathbf{y} = \langle 0, 2, 4, 0, 1 \rangle$$

$$sim_{\text{Jaccard}}(\mathbf{x}, \mathbf{y}) = \frac{0+0+1+0+1}{1+1+1+0+1} = \frac{2}{4}$$

Cosine Similarity

- Cosine Similarity:
  - Measures the **cosine of the angle between two vectors** (the larger the angle, the more dissimilar the documents are).
  - Typically applied to **TF-IDF** or **Bag-of-Words** vectors.
  - Ranges from **0 to 1**, where **1 means identical vectors**.

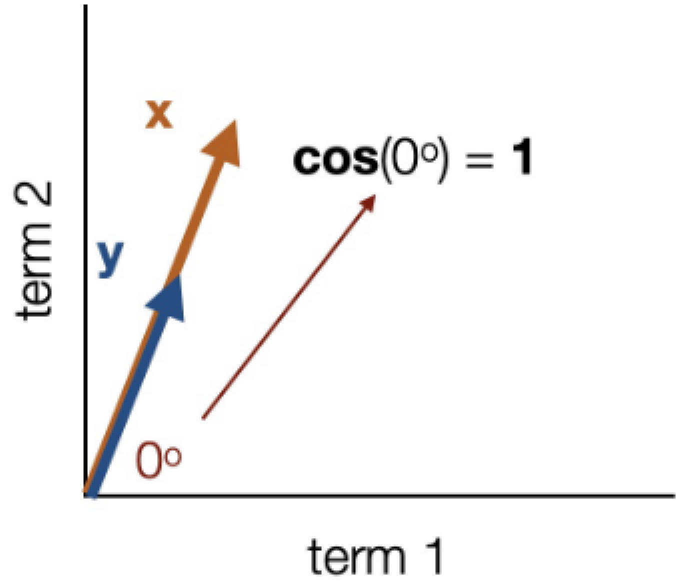
$$sim_{\cos}(x, y) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \ ,$$

where x and y are the term vectors corresponding to documents d_1 and d_2, respectively.

Cosine Similarity – Geometric Interpretation

	term 1	term 2
doc x	1	2
doc y	2	4

$$sim_{\text{cos}}(x, y) = 1$$

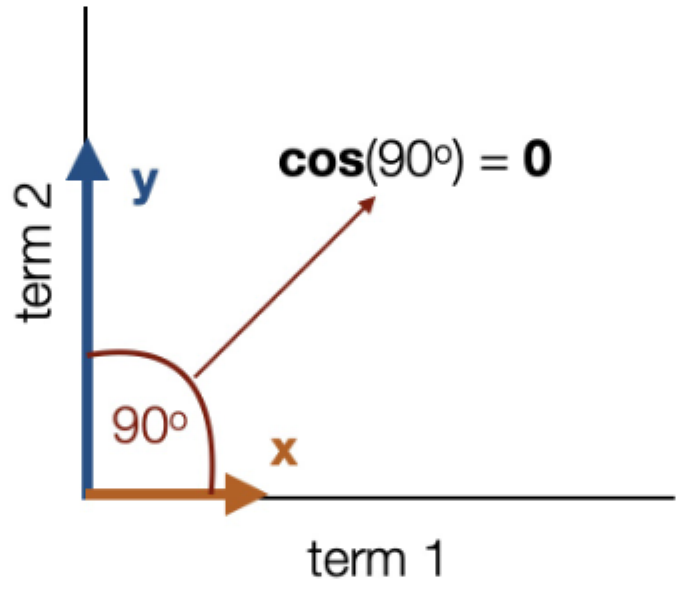


Cosine Similarity – Geometric Interpretation

UNIVERSIDADE CATÓLICA PORTUGUESA
BRAGA

	term 1	term 2
doc x	1	0
doc y	0	2

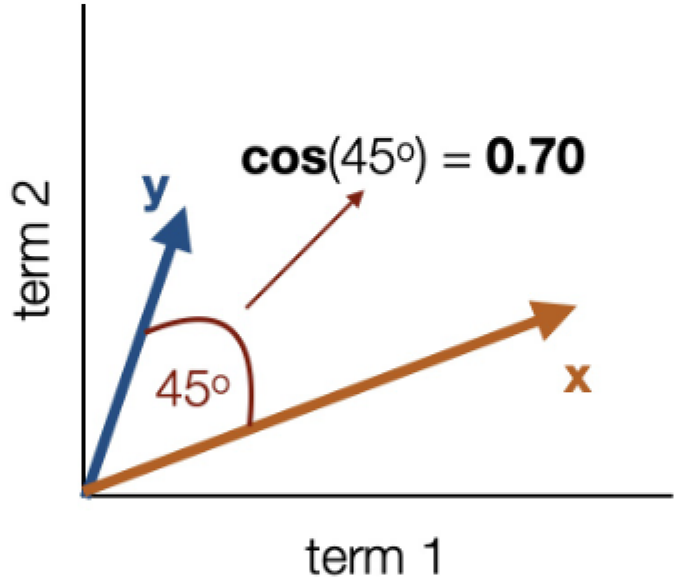
$$sim_{\cos}(x, y) = 0$$



Cosine Similarity – Geometric Interpretation

	term 1	term 2
doc x	4	2
doc y	1	3

$$sim_{\text{cos}}(x, y) = 0.7$$



Cosine Similarity

- Example:

	term 1	term 2	term 3	term 4	term 5
doc x	1	0	1	0	3
doc y	0	2	4	0	1

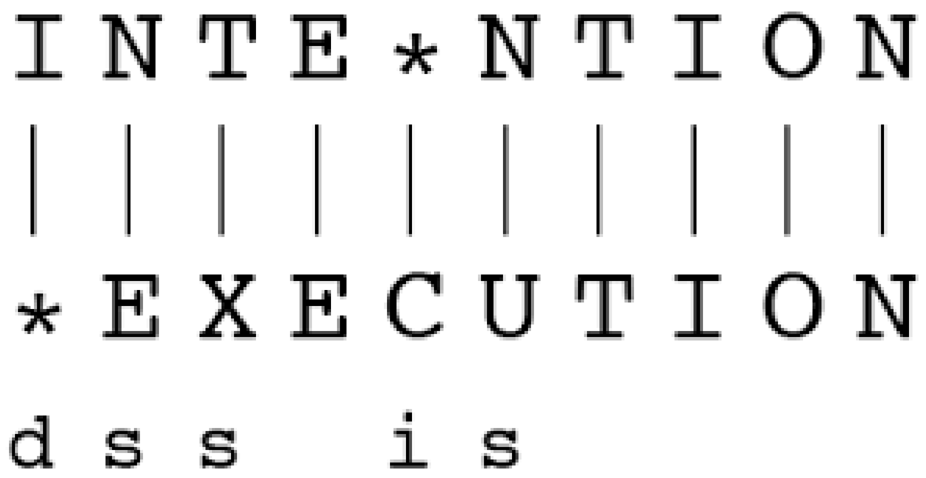
Table: Document-term vectors with term frequencies.

$$\mathbf{x} = \langle 1, 0, 1, 0, 3 \rangle \qquad \mathbf{y} = \langle 0, 2, 4, 0, 1 \rangle$$

$$\begin{aligned} \mathit{sim}_{\text{cos}}(x, y) &= \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \cdot ||\mathbf{y}||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}} \\ &= \frac{1 \times 0 + 0 \times 2 + 1 \times 4 + 0 \times 0 + 3 \times 1}{\sqrt{1^2 + 0^2 + 1^2 + 0^2 + 3^2} \sqrt{0^2 + 2^2 + 4^2 + 0^2 + 1^2}} = \frac{7}{\sqrt{11}\sqrt{21}} \end{aligned}$$

Edit (Levenshtein) Distance

- The minimum number of single-character edits (insertions, deletions, substitutions) required to transform one string into another.



Text Similarity

- Tasks requiring text similarity
  - Clustering Documents
  - Recommending Similar Products
  - Detecting Plagiarism
  - Search Engines and Information Retrieval

Exercises

- Implement jaccard and cosine similarity with numpy.
- Test the implemented functions. Create na heat map to compare sentence similarity.
- Implement a simple spell checker.