

# A Text Summarization Application

Natural Language Processing – Erasmus Project, 2024/25

Luca Bazzetto

Michele Zazzaretti

A dark blue diagonal gradient bar that starts from the bottom left and extends towards the top right, covering the lower half of the slide.

# Contents

- Introduction
- Methodology overview
- Theoretical foundations
- Implementation details
- Demonstration

# Aim & Scope

- **Aim:** Automate summarization for tasks like document review
- **Scope:**
  - Extractive approach using TF-IDF & cosine similarity
  - GUI for text input/upload and summary download

# Methodology

- **Development stack:** Python + NLTK + scikit-learn + Tkinter + NumPy
- **Pipeline stages:**
  - Text preprocessing (RegEx, tokenization, stopwords)
  - Feature extraction (TF-IDF)
  - Sentence ranking (cosine similarity)
  - Selection & summary generation

# Theoretical Foundations

- Extractive vs. Abstractive summarization
- RegEx for sentence splitting
- Tokenization & stopword removal

# TF-IDF & Cosine Similarity

**TF-IDF:** weights words by importance

$$TF\text{-}IDF(t, d) = TF(t, d) \cdot IDF(t)$$

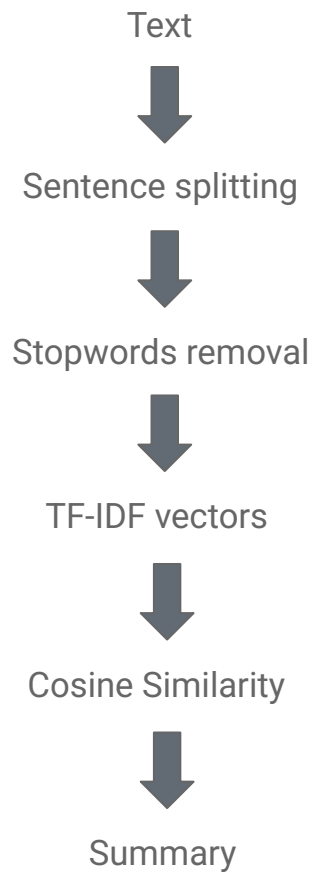
**Cosine similarity:** measures angle between sentence vector and document mean vector

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

# Implementation

- **NLTK**: tokenization, stopwords
- **re**: sentence splitting code snippet
- **scikit-learn**: TfidfVectorizer, cosine\_similarity
- **Tkinter**: GUI components
- **NumPy**: vector/matrix operations

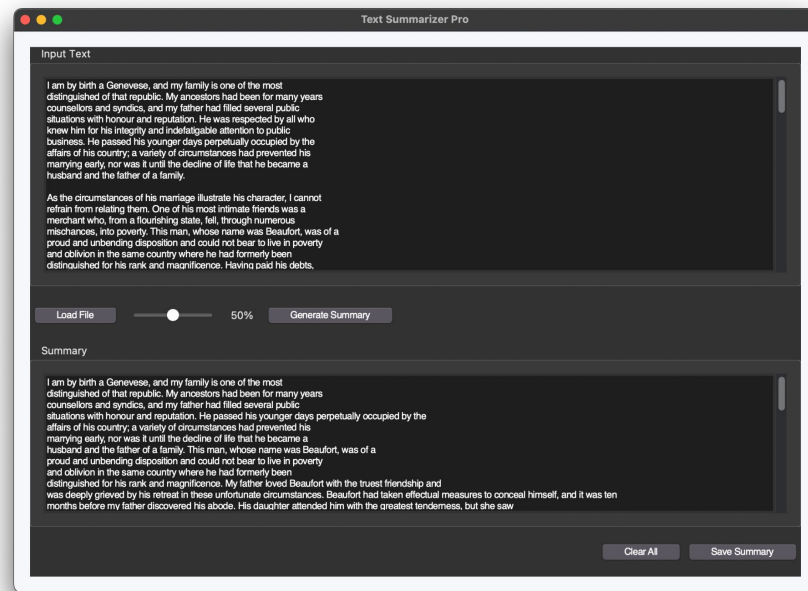
# Pipeline





# GUI Design

- Text paste, typing, file upload
- Summarize button and scrollable result area
- Copy/download summary, restart



# Summary

- Demonstration of summarizing a sample document
- Simplicity, interpretability, interactive GUI