LAB 02: **TEXT CLASSIFICATION**

TUTOR: **MINH N.TA**

CLASS FOR THE COURSE OF NATURAL LANGUAGE PROCESSING – IT4772E

SEMESTER 2024.2

CONTENTS

- Spam Filtering
- AI-Generated Text Detection

SPAM FILTERING

TOPIC INTRODUCTION

- **What is Spam Filtering?**
  Classifying messages (emails, SMS, etc.) as spam or not spam (ham).
- **Why is it important?**
  Protects users from scams, malware, and unwanted content.
- **Where is it used?**
  Email services, messaging apps, social media platforms, etc.

APPROACHES

- **Rule-Based**
  - Uses manually defined keyword lists (e.g., "Buy now", "Free", "Click here").
- **Machine Learning**
  - Treats spam filtering as a text classification task.
  - Basic pipeline:
    - Preprocessing: Lowercase, remove stopwords, punctuation, etc.
    - Feature Extraction: Bag of Words (BoW), TF-IDF.
    - Model: Naive Bayes, Logistic Regression, or SVM.
    - Evaluation: Accuracy, Precision, Recall, F1-score.

DISCUSSIONS

- **Strengths of ML Approach:**
  - Automatically learns patterns from data.
  - Scalable and more robust than rule-based filters.
- **Challenges:**
  - Requires labeled data.
  - Vulnerable to adversarial/spammy text evading detection.
  - Performance depends on preprocessing and feature quality.
- **Extension Ideas:**
  - Try deep learning (LSTM, BERT).
  - Handle class imbalance (spam is often the minority class).
  - Explore multilingual spam detection.

AI-GENERATED TEXT DETECTION

SOME CURRENT TRENDS IN NLP-BASED PROBLEMS

- Fact-Checking and Misinformation Detection
- AI-generated Text Detections
- Bias and Fairness in NLP Models
- Uncertainty Quantification of NLP Models
- Deepfake Text and Voice Detection

TOPIC INTRODUCTION

- **What is AI-Generated Text Detection?**
  Determining whether a given piece of text is written by a human or generated by an AI (e.g., ChatGPT, GPT-3/4).
- **Why is it important?**
  - Detect plagiarism or misuse in education and publishing.
  - Prevent misinformation or fake content online.
  - Protect authenticity and authorship.
- **Real-world use cases:**
  Academic integrity tools, news verification, content moderation.

APPROACHES

- **Traditional Machine Learning**
  - Treat as binary text classification: human vs. AI.
  - Pipeline:
    - Preprocessing: Lowercasing, removing punctuation, stopwords, etc.
    - Feature Extraction: TF-IDF vectors or Bag of Words.
    - Models: Logistic Regression or Naive Bayes.
    - Training: Use a balanced dataset of AI- and human-written texts.
    - Evaluation: Accuracy, F1-score, ROC AUC.

APPROACHES

- **Deep Learning with Fine-Tuned BERT**
  - Use a pre-trained BERT model (e.g., bert-base-uncased)
  - Fine-tuned on the same dataset for the binary classification task.
  - **Advantages:**
    - Learns contextual word representations.
    - Handles subtle semantic differences between AI and human writing.
  - **Implementation Tips:**
    - Tokenize text using BERT tokenizer.
    - Add classification head (dense layer with softmax/sigmoid).
    - Fine-tune with a small number of epochs for good performance.

DISCUSSION

- **Extension Ideas**
  - Try ensemble methods combining ML and BERT predictions.
  - Explore token probability patterns (e.g., GPT logprobs or burstiness).
  - Use explainable AI tools (e.g., LIME, SHAP) to interpret classification decisions.
  - Apply to multilingual or domain-specific texts (e.g., academic writing, code comments).


THE END
THANK FOR YOUR ATTENTION
makeameme.org