Text Classification & Named Entity Recognition

From Feelings to Facts

ITAI 2373 – Module 08

Building on Module 7 Emotion Detection

Learning Outcomes

What You'll Master Today

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

- **Generalize** sentiment analysis into a broad text classification framework.
- **Implement** a complete text classification pipeline.
- **Identify and extract** key entities from text using NER.
- **Apply** these techniques to spoken language and audio transcripts.
- **Analyze** the ethical implications of automated classification and extraction.

Today’s Agenda

Our Investigation Roadmap

● Text Classification Overview

● Classification Pipeline

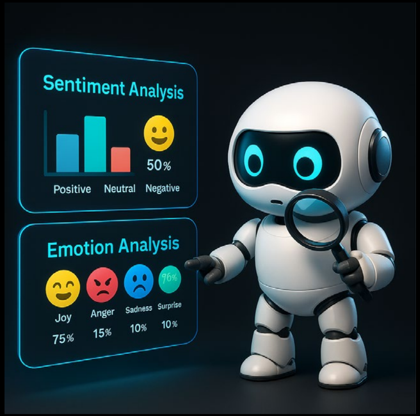
● Named Entity Recognition

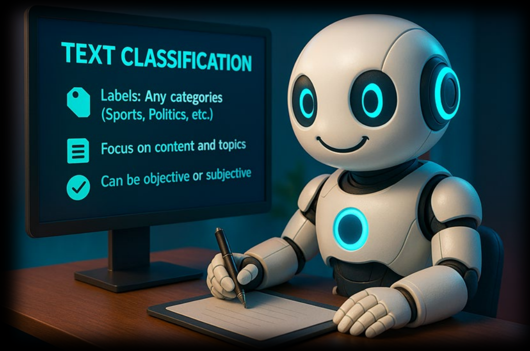
● Real-World Applications

● Ethics of Automated Labeling

From Sentiment to Classification

Same Techniques, Different Labels




TEXT CLASSIFICATION
Labels: Any categories (Sports, Politics, etc.)
Focus on content and topics
Can be objective or subjective

Sentiment Analysis

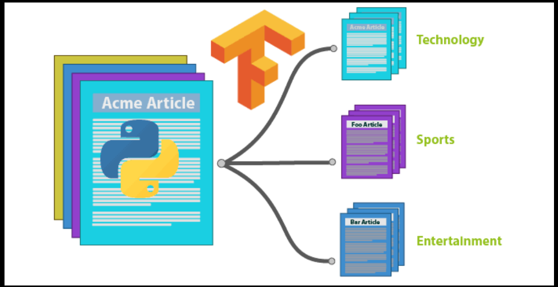
- Labels: Positive, Negative, Neutral
- Focus on opinions and emotions
- Subjective evaluation

Text Classification

- Labels: Any categories (Sports, Politics, etc.)
- Focus on content and topics
- Can be objective or subjective

The Text Classification Pipeline

A Systematic Approach to Classification



1. **Problem Definition**
   Define categories and success criteria

2. **Data Collection**
   Gather and label representative examples

3. **Preprocessing**
   Clean, normalize, and tokenize text

4. **Feature Extraction**
   Convert text to numerical features

5. **Model Training**
   Train and tune classification algorithms

6. **Evaluation**
   Measure performance with metrics

7. **Deployment & Monitoring**
   Integrate and track performance

Classification Algorithms

Choosing the Right Tool for the Job

Naive Bayes

- ✓ Fast training and prediction
- ✓ Works well with small datasets
- ✓ Good for short texts (emails, tweets)
- × Assumes word independence

Logistic Regression

- ✓ Provides probability estimates
- ✓ Easy to interpret feature weights
- ✓ Handles feature correlation well
- × Can overfit with many features

Support Vector Machines

- ✓ Handles high-dimensional data well
- ✓ Effective with clear boundaries
- ✓ Good with medium-sized datasets
- × Slower training than Naive Bayes

Feature Engineering for Classification

Turning Text into Numbers

Basic Features

Bag-of-Words
Count word occurrences, ignore order

TF-IDF
Weight by term frequency and document rarity

N-grams
Capture word sequences (2-3 words)

Advanced Features

POS Tags
Include part-of-speech information

Word Embeddings
Dense vectors capturing semantic meaning

Syntactic Features
Parse trees, dependency relations

Multi-Class vs Multi-Label Classification

One Label or Many?

Multi-Class Classification

Example:

A news article belongs to exactly ONE category: Sports, Politics, Business, or Entertainment

- Each document has exactly one label
- Classes are mutually exclusive
- Standard algorithms work directly

Multi-Label Classification

Example:

A movie can have MULTIPLE genres: Action, Comedy, Romance, Sci-Fi

- Each document can have multiple labels
- Labels can co-occur
- Requires specialized approaches

Evaluation Metrics Deep Dive

Measuring Classification Success

Basic Metrics

◎ **Accuracy**

Correct / Total

Overall correctness, misleading with imbalanced data

Precision

İĨ CĂİĨ Ą ĞĨÅ

When we predict positive, how often are we correct?

Recall

İĨ CĂİĨ Ą ĞÍÅ

Of all positive cases, how many did we catch?

Advanced Metrics

F1-Score

Harmonic mean of precision and recall

Č ẮĂĨǾÑŃÒŒÒŎŌ Ắ Ī ÑŃMÕĨÅCĂĨǾÑŃÒŒÒŎŌ Ą ĪÑŃMÕĨÅ

Macro-Averaging

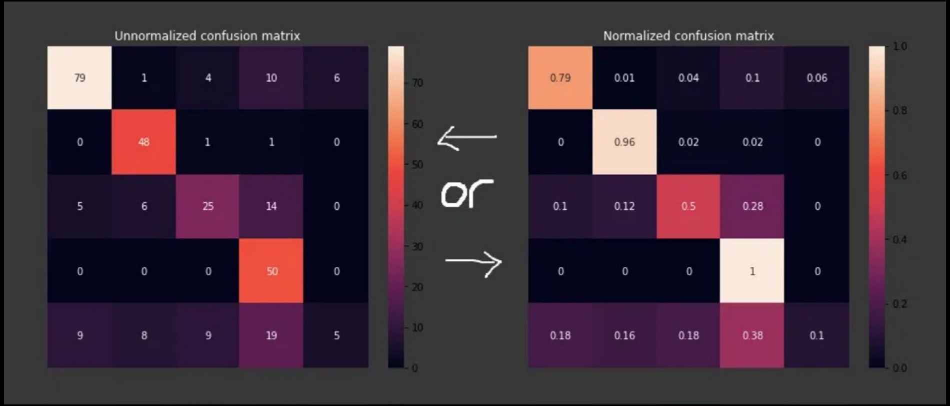
Calculate metrics for each class, then average (equal weight)

Micro-Averaging

Calculate metrics globally (weight by class frequency)

The Confusion Matrix

Understanding Classification Errors



Unnormalized confusion matrix

79	1	4	10	6
0	48	1	1	0
5	6	25	14	0
0	0	0	50	0
9	8	9	19	5

Normalized confusion matrix

0.79	0.01	0.04	0.1	0.06
0	0.96	0.02	0.02	0
0.1	0.12	0.5	0.28	0
0	0	0	1	0
0.18	0.16	0.18	0.38	0.1

What It Shows

- Actual vs. predicted class counts
- Diagonal shows correct predictions
- Off-diagonal shows errors

How To Use It

- Identify commonly confused classes
- Detect class imbalance issues
- Guide feature engineering efforts

Named Entity Recognition - NER

Finding the Facts in Text

Named Entity Recognition (NER) is the task of identifying and categorizing specific entities in text, such as names of people, organizations, locations, dates, and monetary values.

Common Entity Types

- P **PERSON**
  "Elon Musk", "Barack Obama"
- O **ORGANIZATION**
  "Apple Inc.", "United Nations"
- L **LOCATION**
  "San Francisco", "Mount Everest"
- D **DATE/TIME**
  "January 15, 2023", "next Monday"
- $ **MONEY**
  "$50 million", "€100"

NER Approaches

From Rules to Deep Learning

Rule-Based

- Regular expressions and pattern matching
- Gazetteers (lists of known entities)
- Fast but limited coverage

Machine Learning

- Sequence labeling with features
- Learns patterns from annotated data
- Better generalization

Modern Approaches

- Pre-trained models (spaCy, BERT)
- Transfer learning from large datasets
- State-of-the-art performance

NER in Action with spaCy

Seeing Entities in Real Text

Apple Inc. was founded by **Steve Jobs** in **Cupertino** in **California** **1976**

spaCy Output

- O **Apple Inc.** → ORG (Organization)
- P **Steve Jobs** → PERSON
- L **Cupertino** → GPE (Geopolitical Entity)
- L **California** → GPE
- D **1976** → DATE

How It Works

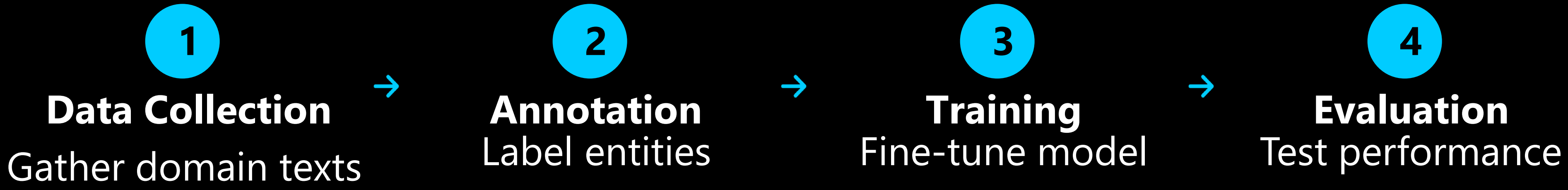
- ✔ Uses context to disambiguate entities
- ✔ Recognizes multi-word entities as single units
- ✔ Handles capitalization and other signals
- ✔ Pre-trained on diverse text corpora

Challenge:

"Apple" could be a fruit or a company - context matters!

Custom NER Training

Building Domain-Specific Entity Recognizers



1 **Data Collection** Gather domain texts → 2 **Annotation** Label entities → 3 **Training** Fine-tune model → 4 **Evaluation** Test performance

Domain-Specific Entities

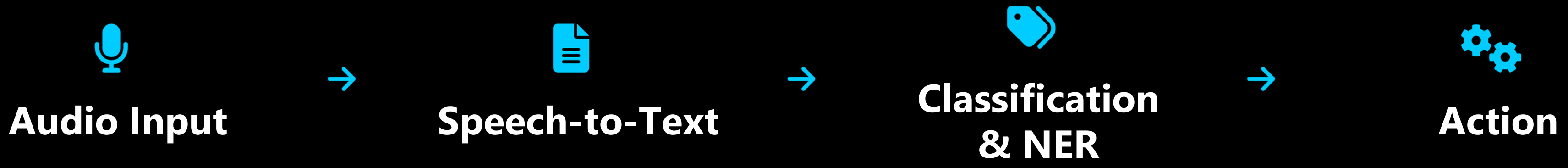
- **Medical:** Diseases, medications, procedures
- **Legal:** Citations, statutes, case names
- **Scientific:** Genes, proteins, chemical compounds
- **Technical:** Part numbers, error codes, specifications

Best Practices

- ✓ Start with pre-trained models when possible
- ✓ Use annotation tools (Prodigy, Doccano)
- ✓ Ensure consistent annotation guidelines
- ✓ Iterate based on error analysis

Audio and Speech Applications

From Voice to Structured Data



Audio Input → Speech-to-Text → Classification & NER → Action

Voice Assistant Applications

Intent Classification:
"Play music" vs. "Set timer"

Entity Extraction: "Navigate to [location]"

Slot Filling:
"Schedule meeting with [person]at [time]"

Command Execution:
"Order [product] from [store]"

Challenges & Solutions

Background Noise: Noise filtering, robust models

Accents & Dialects: Diverse training data

ASR Errors: Error-tolerant classification

Real-time Processing: Optimized pipelines

Real-World Applications

Classification & NER Across Industries

Customer Service

- Ticket Routing:
- Chatbots:
- Feedback Analysis:

Healthcare

- Clinical Document Classification
- Medical NER:
- Clinical Trial Matching:

News & Media

- Content Categorization:
- Entity Tracking:
- Content Filtering:

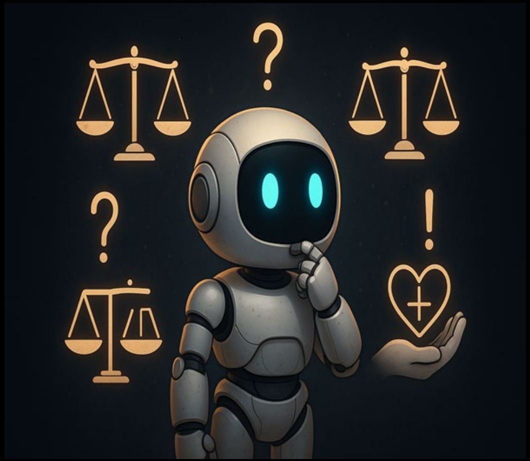
Legal

- Document Classification:
- Legal NER:
- Due Diligence:

Ethics of Automated Classification

Responsible AI in Text Analysis

- ⚖️ Potential Harms
- 🎯 Bias Amplification: Reinforces existing biases
- ⚠️ Misclassification Impact: Real-world harm to people
- 🔒 Privacy Risks: Exposes personal data
- 📦 Black Box Models: Hard to understand decisions
- ✅ Responsible Practices
- 📊 Diverse Data: Multiple perspectives
- 🔍 Bias Testing: Regular fairness checks
- 👥 Human Review: Critical decision oversight
- 📋 Clear Documentation: Model cards & use cases



Lab Preview

Hands-On Classification & NER

Classification Tasks

- **News Categorization:** Classify Reuters news articles into topics
- **Review Analysis:** Predict ratings from Amazon product reviews
- **Voice Command Classification:** Identify intents in spoken requests

NER Tasks

- **General Entity Recognition:** Identify standard entities with spaCy
- **Custom Entity Training:** Train a model to recognize technical terms
- **Performance Evaluation:** Measure precision, recall, and F1-score

Tools & Libraries

- **scikit-learn:** For traditional ML classifiers
- **spaCy:** For NER and text processing
- **Matplotlib/Seaborn:** For visualizing results

Learning Objectives

- ✓ Implement complete classification pipeline
- ✓ Compare performance of different algorithms
- ✓ Train and evaluate custom NER models

Performance Optimization

Making Systems Fast, Accurate, and Scalable

⚡ Speed Optimization

🔽 **Feature Selection:**
Use key features only

🎯 **Dimensionality Reduction:**
PCA or truncated SVD

🏗️ **Model Distillation:**
Smaller models mimic larger ones

💻 **Hardware Acceleration:**
GPU/TPU for inference

🎯 Accuracy Optimization

📊 **More Training Data:**
Focus on edge cases

⚙️ **Hyperparameter Tuning:**
Grid or random search

🔗 ***Ensemble Methods:***
Combine multiple models

🔍 ***Error Analysis:***
Target specific errors

⚖️ *Scalability Optimization*

📦 ***Batch Processing:***
Process inputs in batches

💾 ***Caching:***
Store common results

🌐 ***Distributed Processing:***
Split work across machines

☁️ ***Cloud Deployment:***
Auto-scaling resources

Model Deployment and Monitoring

From Development to Production

Deployment Strategies:

- REST APIs for real-time predictions
- Batch processing for large-scale analysis
- Edge deployment for low-latency applications

Monitoring Essentials:

- **Performance drift:** Accuracy degradation over time
- **Data drift:** Changes in input data distribution
- **Concept drift:** Changes in the relationship between features and labels

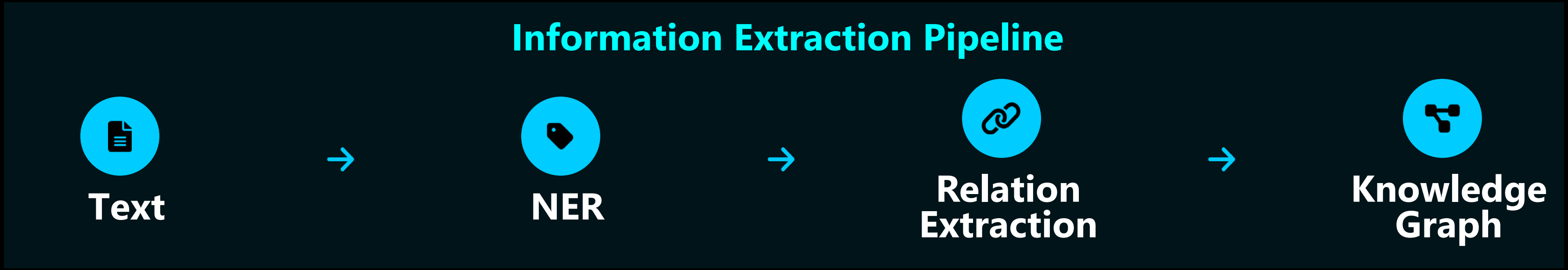
Maintenance:

- Regular retraining with new data
- A/B testing for model updates
- Rollback procedures for failed deployments

Deploy → **Monitor** → **Update** → **Validate**

Integration with Other NLP Tasks

Building Comprehensive Text Processing Pipelines



Information Extraction Pipeline

Text → NER → Relation Extraction → Knowledge Graph

Complementary NLP Tasks

- **Dependency Parsing:** Understand grammatical structure
- **Coreference Resolution:** Connect pronouns to entities
- **Dialogue Systems:** Maintain context in conversations
- **Machine Translation:** Preserve entities across languages

Knowledge Graph Applications

- **Semantic Search:** Find content by meaning, not just keywords
- **Question Answering:** Extract precise answers from text
- **Recommendation Systems:** Connect related content
- **Reasoning Systems:** Draw inferences from extracted facts

Future Directions

Emerging Trends in Classification & NER

Emerging Trends

- **Few-shot learning:** Good performance with minimal training data
- **Cross-lingual models:** Work across multiple languages
- **Multimodal integration:** Combine text, images, and audio

Advanced Techniques:

- **Transformer-based models:** BERT, RoBERTa for better context understanding
- **Zero-shot classification:** Classify without training examples
- **Continual learning:** Models that adapt continuously

Key Takeaways

Essential Concepts & Skills

- Classification operates at document level, NER at word/phrase level
- Feature engineering is crucial for model performance
- Different metrics measure different aspects of performance
- Domain adaptation bridges general models to specific applications
- Classification and NER as building blocks for complex NLP systems
- Human-AI collaboration for optimal results

Connection to Module 9

From Classification to Topic Modeling

Module 8

Text Classification & Named Entity Recognition

→

Building on foundations

Module 9

Topic Modeling & Advanced Text Analysis

From Supervised to Unsupervised

- **Classification:** Predefined categories with labeled data
- **Topic Modeling:**Discovering themes without labels
- **Shared Foundation:** Vector representations of text

New Techniques You'll Learn

- **LDA & NMF:**Statistical topic modeling approaches
- **Word & Document Embeddings:**Dense vector spaces
- **Hierarchical Clustering:** Organizing content by similarity

How They Work Together

- **Pre-filtering:**Classify documents before topic modeling
- **Enrichment:** Add entity information to topic models
- **Exploration:** Discover topics within specific categories

Real-World Applications

- **Content Analysis:** Understand themes across documents
- **Customer Feedback:** Identify emerging issues
- **Research:** Discover patterns in scientific literature

Module Summary

From Feelings to Facts

Our Learning Journey

Sentiment Analysis

Feelings in text

→

Text Classification

Categories of content

→

Named Entity Recognition

Specific facts in text

→

Integration

Complete NLP systems

What We've Learned

We've explored transforming unstructured text into structured information through classification and entity extraction, building complete pipelines from data collection to deployment.

Why It Matters

These techniques are fundamental building blocks of modern NLP systems, bridging the gap between human language and machine processing.

Tools & Techniques

We've worked with feature engineering approaches, algorithms from traditional ML to deep learning, and libraries like scikit-learn and spaCy.

Looking Ahead

Next, we'll explore topic modeling and advanced text analysis, moving from supervised to unsupervised learning to discover latent patterns in text.