



THE UNIVERSITY of EDINBURGH
Centre for Data, Culture & Society



Text Classification in Practice: From Topic Models to Transformers

Zongxiao Wu & Joy Lan

Zongxiao.Wu@ed.ac.uk & plan@ed.ac.uk

20 November 2025



www.cdcs.ed.ac.uk

What is Text Classification?

apple, banana, cat, elephant, Edinburgh



What is Text Classification?

apple, banana, cat, elephant, Edinburgh

fruit

animal

place



What is Text Classification?

In the early morning light of Hawaii, the warm ocean breeze drifted inland and carried with it the scent of tropical pineapples and mangoes from the local farms. Ava stepped out of her small cottage and smiled at the lively sounds coming from the forest nearby. A playful monkey swung between the branches, chattering noisily as if announcing the start of a new day. Beside the monkey, a bright blue parrot perched on a branch, pecking thoughtfully at a piece of fresh papaya.

Ava had always loved how the island blended peaceful landscapes with vibrant wildlife. She grabbed a basket filled with bananas, oranges, and grapes, intending to bring them to the coastal market later that afternoon. Before heading out, she added a few more fruits she had harvested the day before—strawberries, watermelons, and juicy kiwis—making the basket nearly overflow with color.



What is Text Classification?

Information Overload:

Every day, billions of documents, posts, and messages are created. Manual reading and labeling is infeasible.

Solution?

Text classification automates the categorization process by assigning text to one or more categories, making unstructured data manageable.



What is Text Classification?

Definition

- Text classification is the task of assigning natural language text to one or more categories.
- It transforms unstructured text into structured data, enabling machines to “mimic human understanding” of the content at scale.
- Can operate at **document level** (entire articles), **sentence level** (single sentence), or even **sub-sentence level** (phrases, entities).



Examples of Text Classification

- **Sentiment Analysis:**

Classify tweets or product reviews as *positive, negative, or neutral*.

Example: “The camera quality is amazing!”

→ *Positive*.

- **News Categorization:**

Assign news headlines to categories: *Politics, Sports, Technology, Entertainment*.

Example: “Apple launches new AI chip”

→ *Technology*.



Examples Text Classification

- **Spam Detection:**

Classify emails or SMS as *Spam* vs. *Not Spam*.

Example: “Win a free iPhone!”

→ *Spam*.

- **Healthcare Applications:**

Classify clinical notes into disease categories.

Example: “Patient reports shortness of breath, prescribed inhaler.” → *Asthma*.

- **Legal and Policy:**

Categorize case law or government reports for easier retrieval.



Why Text Classification?

Advantages

- **Efficiency:** Handle massive text collections faster than human experts.
- **Scalability:** Adaptable across domains—finance, healthcare, e-commerce, social media.
- **Consistency:** Avoid subjective human bias, ensuring stable decision criteria.
- **Predictive Power:** Extract knowledge patterns that help forecast trends (e.g., market sentiment).



Why Text Classification?

Business Impact: (solution? or not?)

- Customer support: route tickets to correct department; marketing: segment customers by feedback sentiment, etc.

Foundation for Natural Language Processing (NLP):

- Many advanced tasks—question answering, summarization, recommendation—rely on classification as a subtask.



How Does Text Classification Work?

Overview:

- 1950s: Naïve Bayes, rule-based systems.
- 1990s: Statistical NLP models (linear, non-linear, probabilistic models)
- 2010s: Neural NLP models (RNNs, CNNs).
- 2018+: Transformers and pre-trained LMs (BERT, GPT, RoBERTa).
- 2022+: Large language models (ChatGPT, Gemini, Llama)



How Does Text Classification Work?

Step 1. Data Collection & Preprocessing

Step 2. Feature Representation (depend on the chosen model)

Step 3. Dimensionality Reduction (optional)

Step 4. Model Training

Step 5. Evaluation

Step 6. Deployment



How Does Text Classification Work?

Step 1. Data Collection & Preprocessing

- Gather domain-specific text (e.g., reviews, medical notes, legal documents).
- Clean the data:
 - a) Remove noise (HTML tags, emojis, special symbols).
 - b) Normalize case (lowercasing, except acronyms).
 - c) Handle spelling errors, abbreviations, slang.
 - d) Tokenization: split text into words/subwords.
 - e) Remove stopwords (e.g., “a,” “the,” “and”).
 - f) Stemming or lemmatization to reduce words to base form.



How Text Classification?

Step 2. Feature Representation (depend on the chosen model)

- **Bag-of-Words (BoW)**: simple word counts; ignores order.
- **TF-IDF**: weighs words by frequency vs. rarity.
- **Word Embeddings**: Word2Vec, GloVe, FastText capture semantic similarity.
- **Contextual Embeddings**: ELMo, BERT capture meaning depending on context.

Step 3. Dimensionality Reduction (optional)

- High-dimensional vectors are sparse and inefficient.
- Methods: PCA, LDA, NMF, Autoencoders, Random Projection.
- Benefit: faster training, less overfitting.



How Text Classification?

Step 4. Model Training

- **Traditional (Statistical) Models:**
 - ☐ Naïve Bayes (linear model).
 - ☐ Support Vector Machines (margin-based, both linear and non-linear).
 - ☐ Latent Dirichlet Allocation (probabilistic).
- **Neural Models:**
 - ☐ RNNs (capture sequential order).
 - ☐ CNNs (detect local patterns like n-grams).
 - ☐ Transformers (self-attention, contextual understanding).



How Text Classification?

Step 5. Evaluation

- Metrics depend on task:
Accuracy (overall correctness), consider imbalanced datasets: accuracy may be misleading.
Precision, Recall, F1 (balance false positives/negatives); ROC-AUC (ranking ability), etc.

Step 6. Deployment

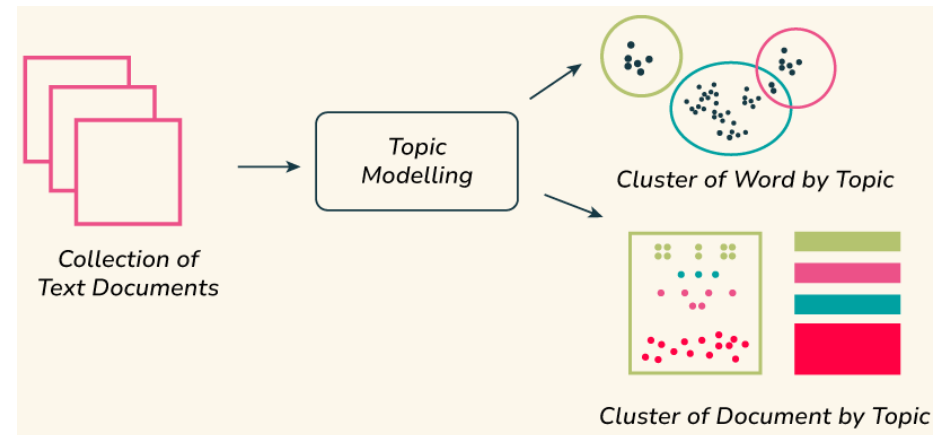
- Integrate into applications:
Email spam filters; real-time recommendation engines.
Customer service chatbots; healthcare diagnosis support tools, etc.



Models in Practice

Topic Models

- *Representative*: Latent Dirichlet Allocation (LDA).
- Learns hidden topics in a collection of documents.
- Each document is represented as a mixture of topics; each topic is a distribution over words.
- *Example*: News classification—documents mapped into topic space, then classified into categories.
- *Limitation*: Ignores word order; works better on long documents.



Models in Practice

Recurrent Neural Networks (RNN)

- *Representative*: Long Short-Term Memory (LSTM).
- Designed to capture sequential dependencies and long-term context.
- *Example*: Sentiment analysis of movie reviews—model understands word order (“not good” ≠ “good”).
- *Strengths*: Handles variable-length sequences.
- *Limitations*: Training can be slow; hard to capture very long dependencies.



Models in Practice

Transformers

- *Representative*: BERT (Bidirectional Encoder Representations from Transformers).
- Based on self-attention mechanism; models relationships between all words in a sequence simultaneously.
- Pre-trained on massive corpora, fine-tuned for specific tasks.
- *Example*: Fine-tuning BERT on IMDB dataset for sentiment classification, achieving state-of-the-art accuracy.
- *Strengths*: Captures bidirectional context, efficient transfer learning.
- *Limitations*: Computationally expensive; large memory requirements.



Hands-on Session: Text Classification in Google Colab

Our Github page: <https://github.com/DCS-training/Text-Calssification-in-Practice-From-Topic-Models-to-Transformers>

Three tasks:

- **Topic Models – LDA for News Classification**

Colab link: https://github.com/DCS-training/Text-Calssification-in-Practice-From-Topic-Models-to-Transformers/blob/main/Task1-LDA-News%20Classification/Task1_LDA_News_Classification.ipynb

- **Recurrent Neural Network – LSTM for Ecommerce Classification**

Colab link: <https://github.com/DCS-training/Text-Calssification-in-Practice-From-Topic-Models-to-Transformers/blob/main/Task2-LSTM-EcommerceClassification/LSTM.ipynb>



Hands-on Session: Text Classification in Google Colab

Our Github page: <https://github.com/DCS-training/Text-Calssification-in-Practice-From-Topic-Models-to-Transformers>

Three tasks:

- **Transformer – BERT for Tweets Sentiment Classification**

Colab link: <https://github.com/DCS-training/Text-Calssification-in-Practice-From-Topic-Models-to-Transformers/blob/main/Task3-BERT-TweetsClassification/BERT.ipynb>

