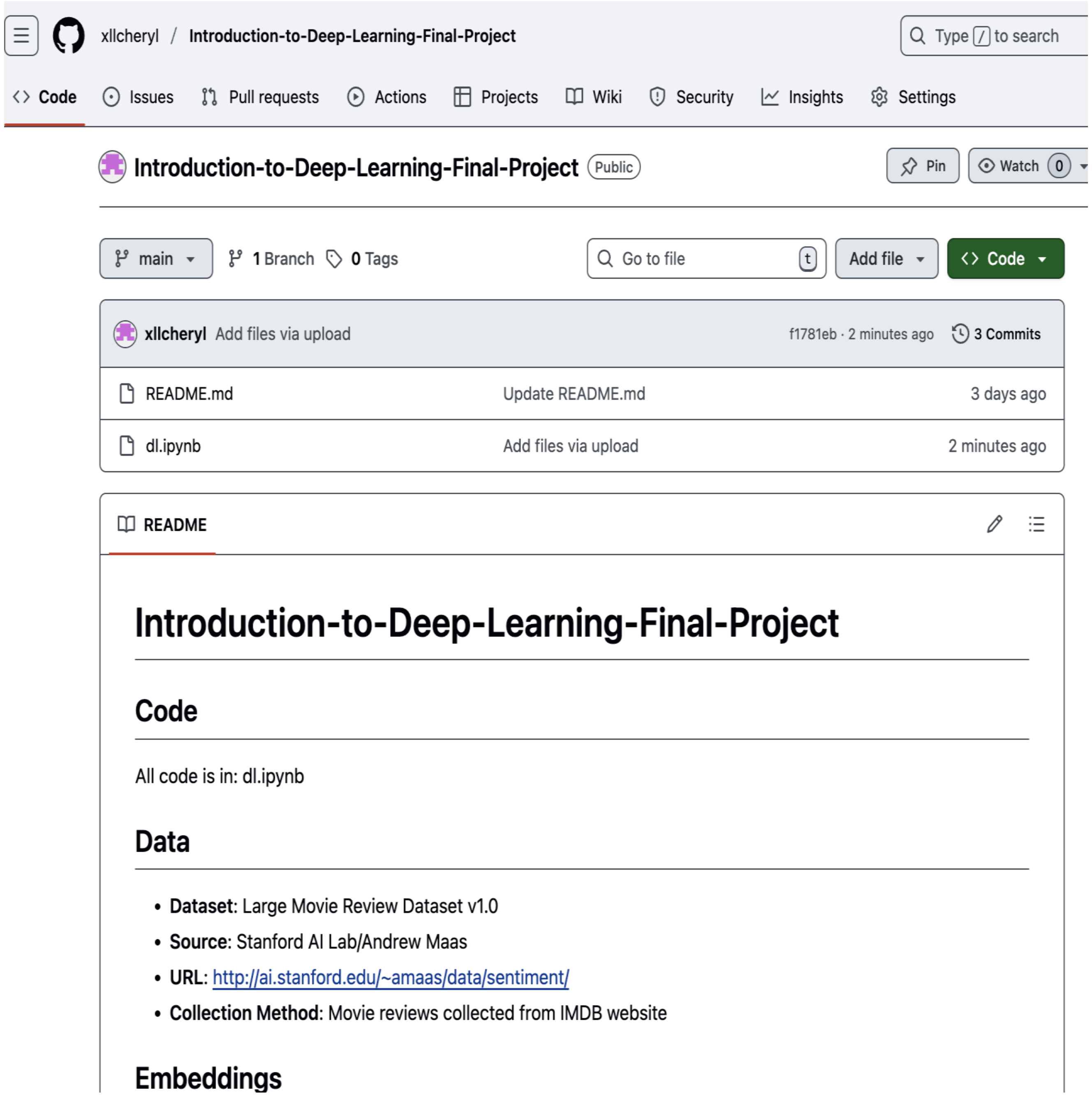
CSCA 5642 Introduction to Deep Learning Final Project

LINLI XIANG

GitHub Link

https://github.com/xllcheryl/Introduction-to-Deep-Learning-Final-Project.git



1. Project Overview

The primary goal of this project is to build, compare, and evaluate multiple classification models for sentiment analysis, including:

1. Baseline Models

- **Logistic Regression**
- **Support Vector Machine (SVM)**

2. Deep Learning Models

- **Deep Neural Network (DNN)**
- **Recurrent Neural Network (RNN)** with Long Short-Term Memory (LSTM) units

3. Research Paper Implementations

- 3.1 CNN Architectures
- 3.2 Transformer-Based Models

2. Data Collection

- **Dataset**: Large Movie Review Dataset v1.0
- **Source**: Stanford AI Lab/Andrew Maas
- **URL**: http://ai.stanford.edu/~amaas/data/sentiment/
- **Collection Method**: Movie reviews collected from IMDB website
- **License**: Academic use permitted

Dataset Characteristics

Data Provenance

- The dataset was created by researchers at Stanford University for academic research in sentiment analysis and text classification. The data was collected from IMDB movie reviews posted before 2011, ensuring a diverse range of movies and review styles. The dataset has become a benchmark for sentiment analysis tasks in the NLP community.

50,000 movie reviews (25,000 training, 25,000 testing)

Binary classification (positive/negative sentiment)

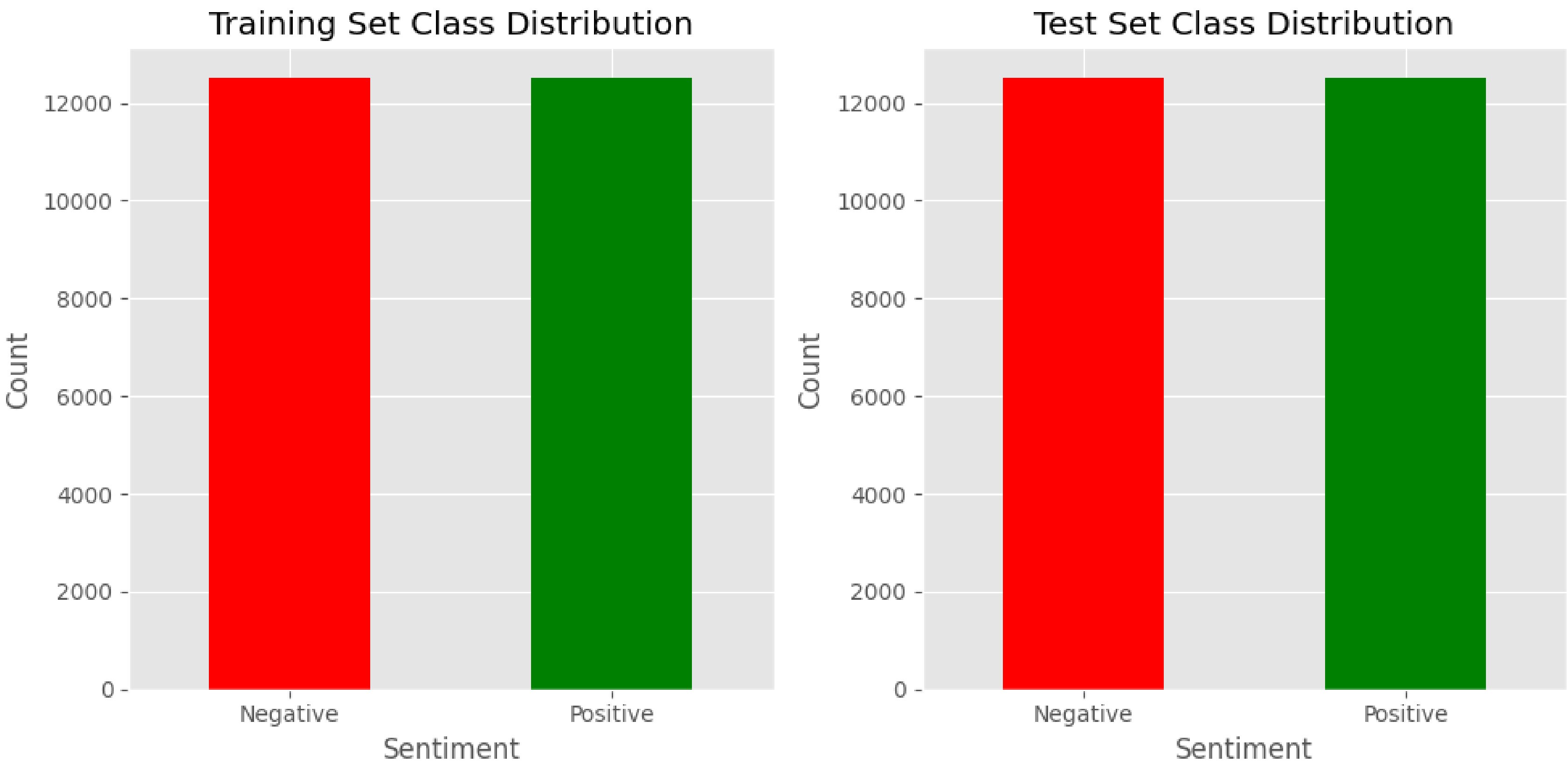
Even class distribution (50% positive, 50% negative)

No more than 30 reviews per movie to prevent bias

Raw text data with minimal preprocessing

Target Variable Distribution

Both training and test sets have perfectly balanced class distributions (50% positive, 50% negative)



3. Exploratory Data Analysis (EDA)

01

Text Length Analysis

02

Word Frequency Analysis

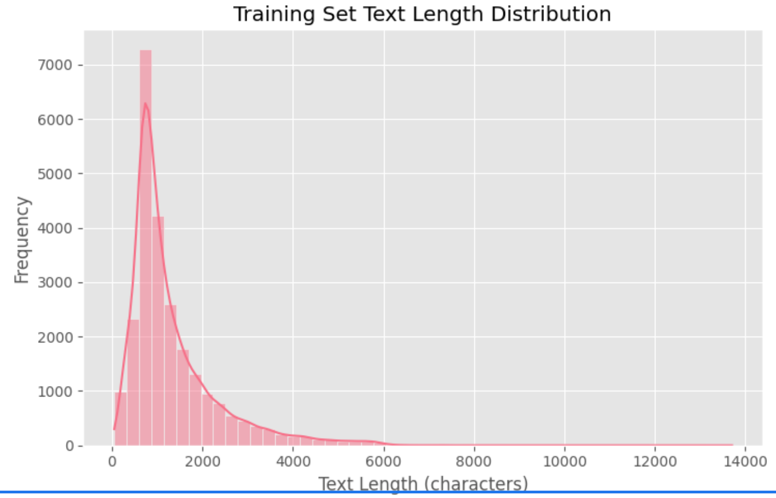
03

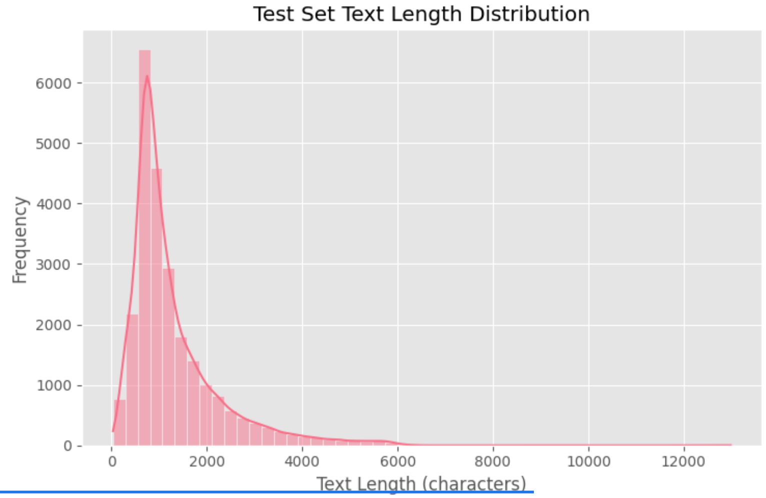
Text statistics

04

Data Cleaning and Preprocessing

Text Length Analysis





Training- Mean length: 1325.07 characters

Training- Median length: 979.0 characters

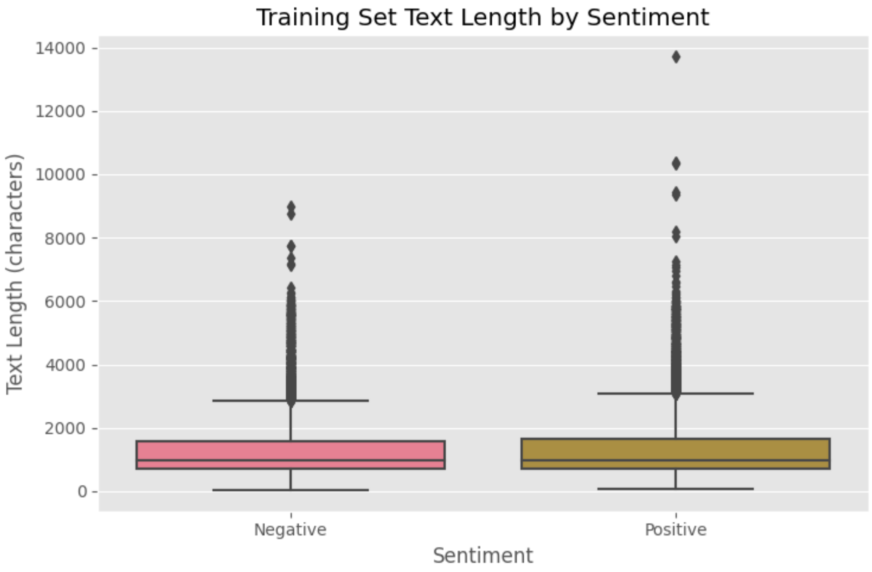
Test- Mean length: 1293.79 characters

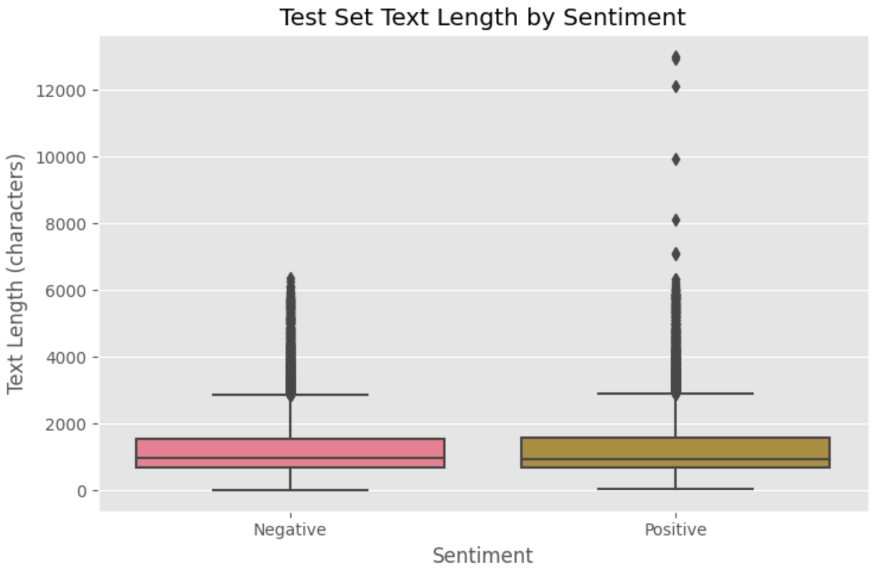
Test- Median length: 962.0 characters

Text Length by Sentiment

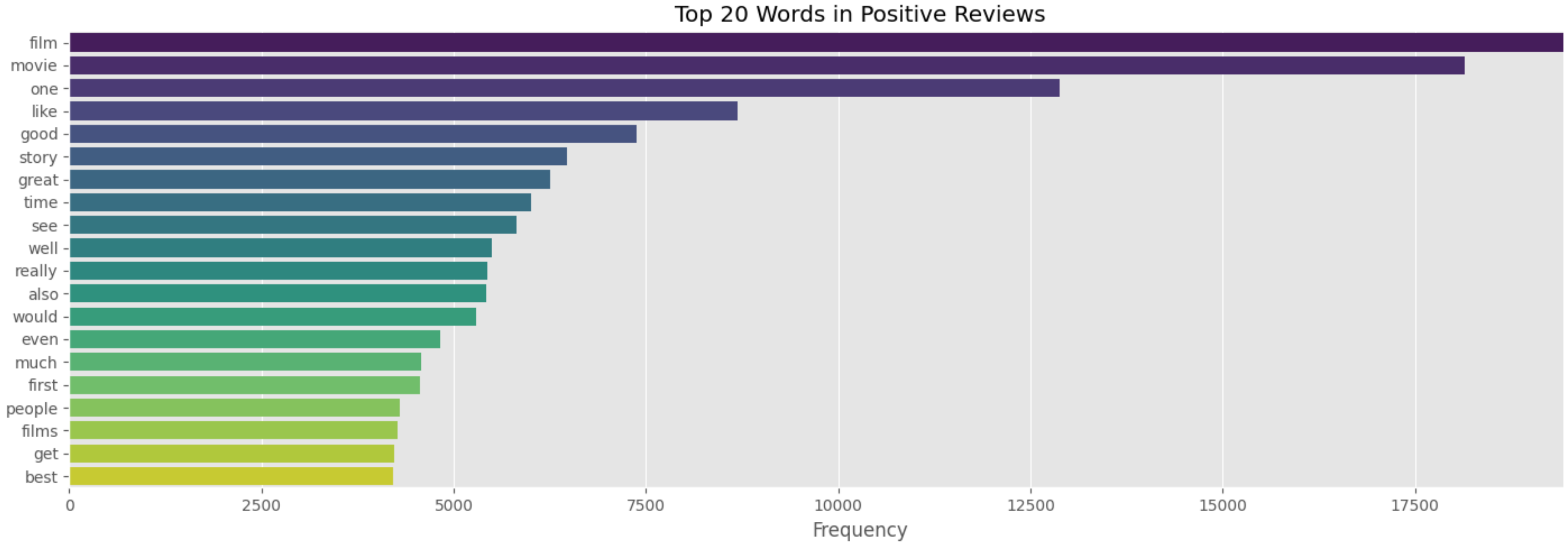
T-TEST FOR DIFFERENCE IN TEXT LENGTH:

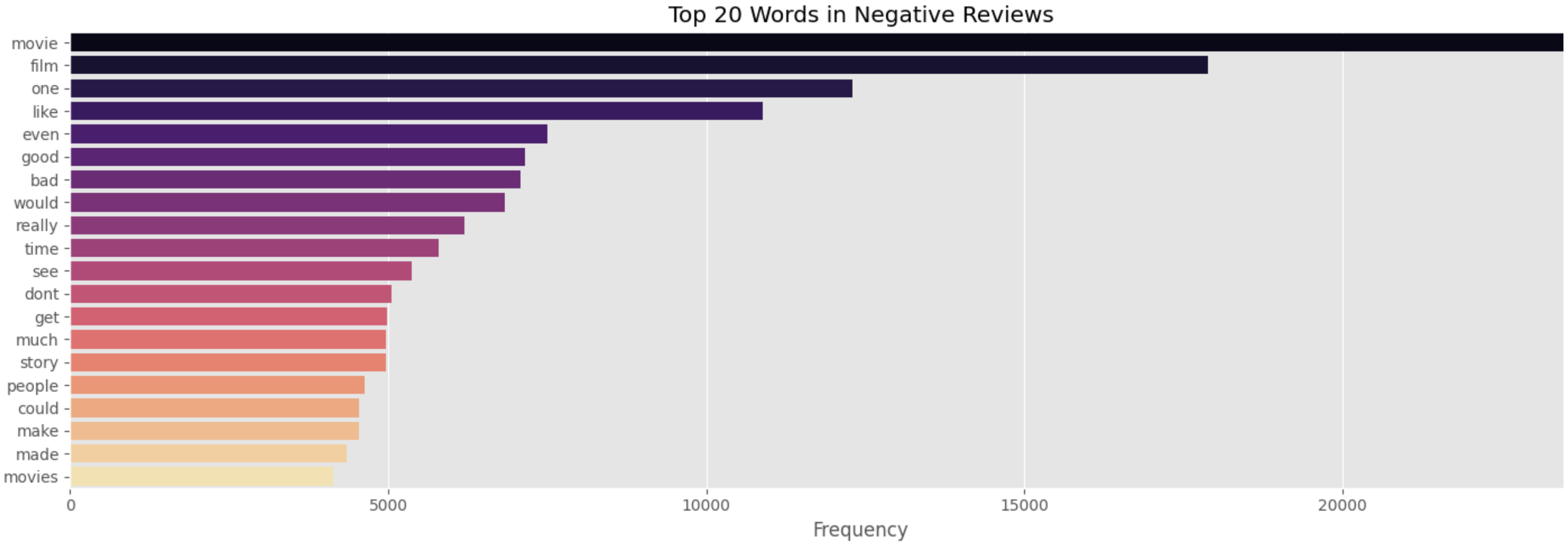
- **T-STATISTIC = 3.483**
- **P-VALUE = 0.000**





Positive reviews tend to be slightly longer than negative reviews on average, and this difference is statistically significant ($p < 0.05$). However, the effect size is small, so text length alone is not a strong predictor of sentiment.

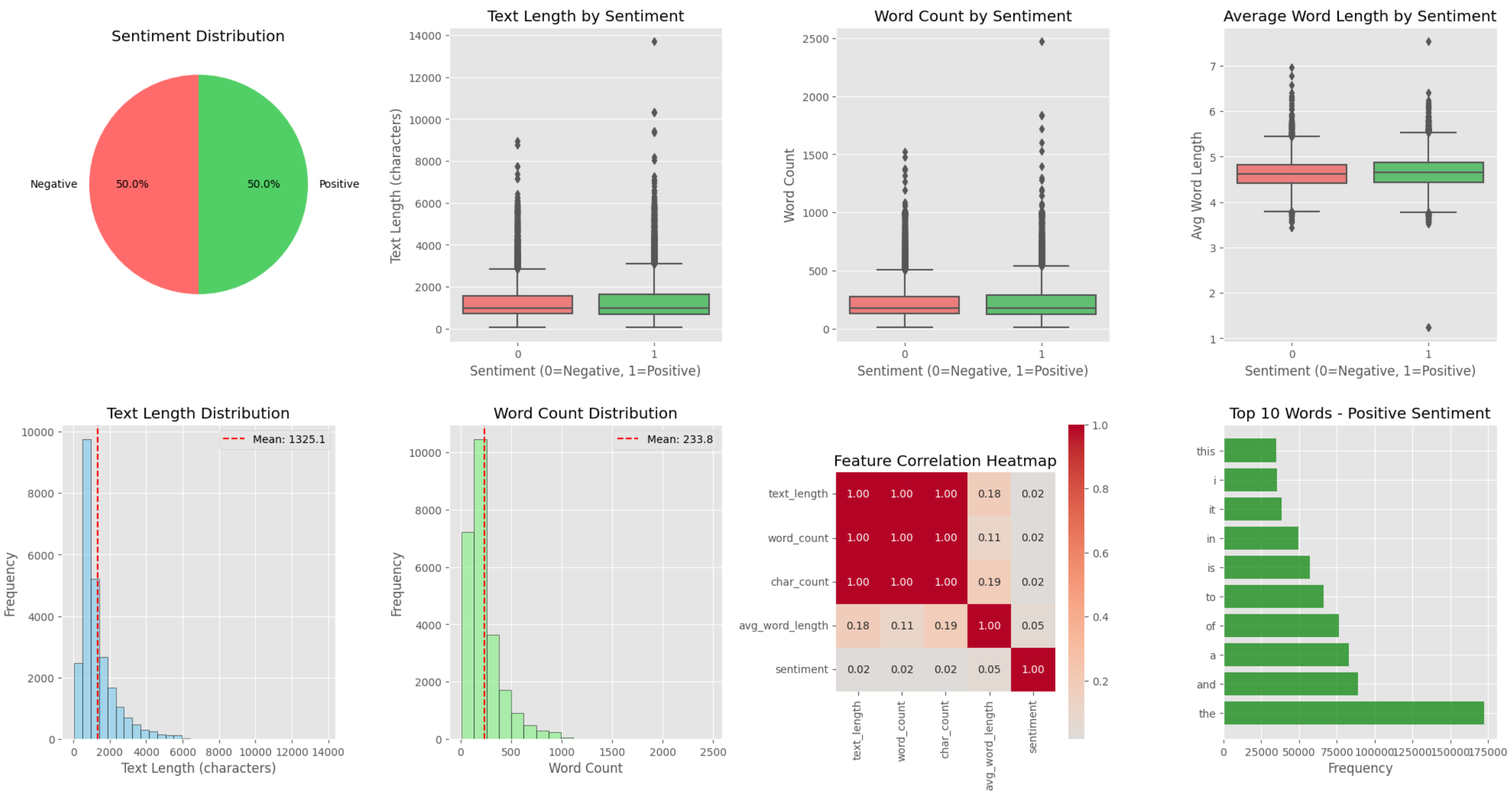




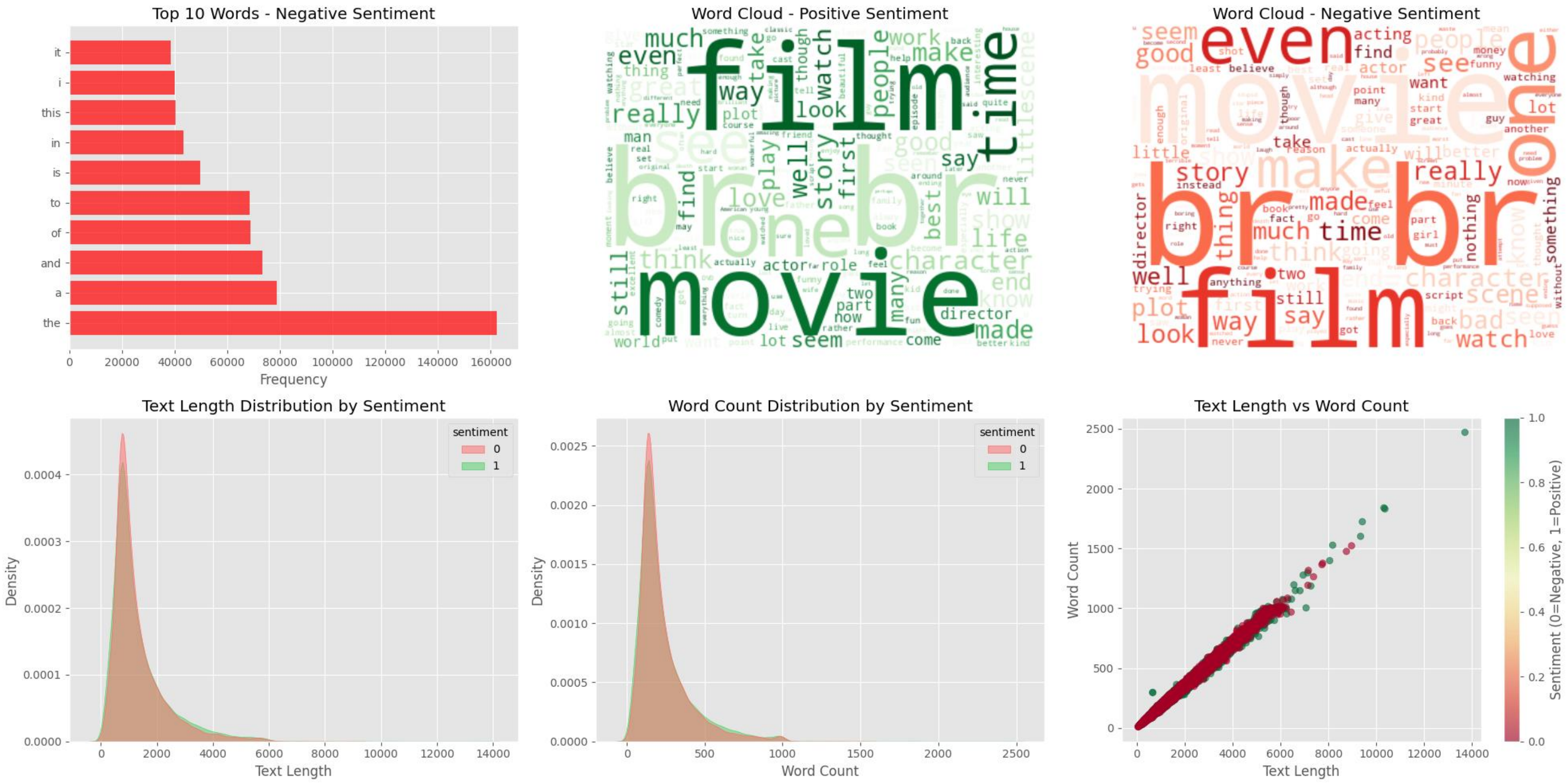
Word Frequency Analysis

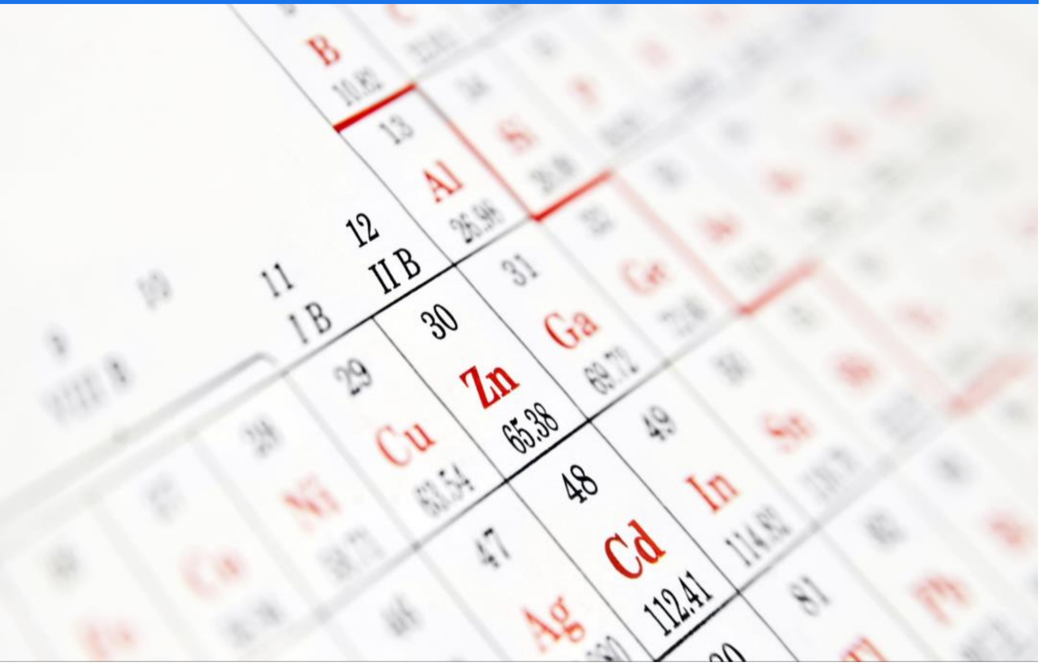
- Both positive and negative reviews share many common words related to movies (film, movie, story, character).
- However, positive reviews contain more positive sentiment words (great, best, good, excellent), while negative reviews contain more negative sentiment words (bad, worst, terrible, awful).
- This suggests that word choice is a strong indicator of sentiment.

Text statistics



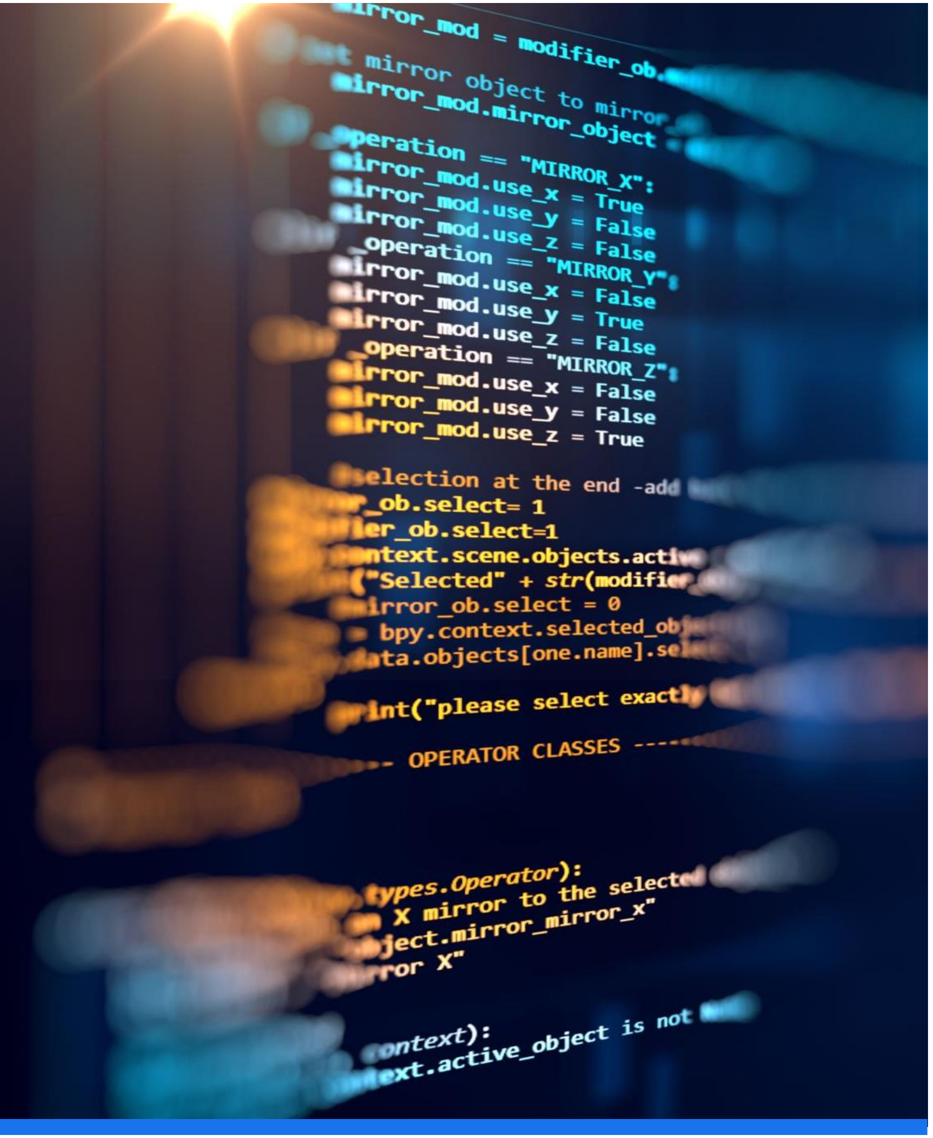
Sentiment distributio





Text statistics

- Text Length Statistics:
  - Min length: 52 characters
  - Max length: 13704 characters
  - Std deviation: 1003.13 characters
- Word Count Statistics:
  - Min words: 10 words
  - Max words: 2470 words
  - Std deviation: 173.73 words
- Positive Sentiment Texts:
  - Average length: 1347.16 characters
  - Average words: 236.71 words
- Negative Sentiment Texts:
  - Average length: 1302.98 characters
  - Average words: 230.87 words



Data Cleaning and Preprocessing

- **Text Cleaning**: Remove HTML tags, punctuation, and numbers
- **Normalization**: Convert to lowercase, lemmatize words
- **Stopword Removal**: Remove common English stopwords
- **Sequence Length**: Standardize to 200 tokens for neural networks
- **Vocabulary Size**: Limit to 10,000 most frequent words

4. Model Building and Training

Baseline Models

- LR
- SVM
- **Hyperparameter Tuning**

Deep Learning Models

- DNN
- LSTM

Research Paper Implementations

- CNN
- BERT

Class	Precision	Recall	F1-Score	Support
0	0.90	0.87	0.89	2 500
1	0.88	0.91	0.89	2 500
Avg / Total	**0.89**	**0.89**	**0.89**	**5 000**

Logistic Regression

ACCURACY: 88.8 %

Class	Precision	Recall	F1-Score	Support
0	0.89	0.87	0.88	2 500
1	0.87	0.89	0.88	2 500
Avg / Total	**0.88**	**0.88**	**0.88**	**5 000**

SVM

ACCURACY: 87.8 %

Hyperparameter Tuning by GridSearchCV

Class	P	R	F1	Support
0	.90	.87	.89	2 500
1	.88	.91	.89	2 500
Avg	**.89**	**.89**	**.89**	**5 000**

Logistic Regression
Best params: C = 1, penalty = l2, solver = liblinear
Accuracy: 88.8 %
Grid fits: 5 × 10 = 50

Class	P	R	F1	Support
0	.91	.87	.89	2 500
1	.88	.91	.89	2 500
Avg	**.89**	**.89**	**.89**	**5 000**

Linear SVM
Best params: C = 0.1, penalty = l2, loss = squared_hinge, dual = False
Accuracy: 89.0 %
Grid fits: 5 × 10 = 50

Epoch	Train Acc.	Val Acc.	Val Loss
1	72.7 %	89.1 %	0.270
2	89.9 %	88.4 %	0.295
3	93.6 %	87.4 %	0.366
4	97.0 %	87.0 %	0.464

Simple DNN Model

BEST VALIDATION ACCURACY (EPOCH 1): 90.9 %

Epoch	Train Acc.	Val Acc.	Val Loss
1	50.5 %	52.7 %	0.690
2	52.2 %	50.8 %	0.694
3	50.4 %	52.1 %	0.683
4	52.4 %	54.3 %	0.682
5	55.4 %	54.2 %	0.712
6	56.1 %	55.2 %	0.754
7	62.4 %	82.7 %	0.464
8	83.4 %	84.5 %	0.408
9	88.5 %	86.2 %	0.372
10	92.4 %	85.0 %	0.410

RNN Model (LSTM)

BEST VALIDATION ACCURACY (EPOCH 9): 86.2 %

CNN Models (Implementing the Paper Architectures)

ADAM, BATCH 32, 10 EPOCHS MAX

Epoch	CNN-rand	CNN-static	CNN-non-static	CNN-multichannel
1	86.4 %	84.3 %	86.6 %	86.2 %
2	87.4 %	84.9 %	88.3 %	87.5 %
3	82.9 %	85.9 %	85.5 %	83.9 %
4	85.5 %	79.5 %	87.2 %	82.9 %
5	87.0 %	73.4 %	79.9 %	**88.6 %**

Transformer-based Models (BERT)

Class	Precision	Recall	F1-Score	Support
0	0.94	0.93	0.94	12 500
1	0.93	0.94	0.94	12 500
Avg	**0.94**	**0.94**	**0.94**	**25 000**

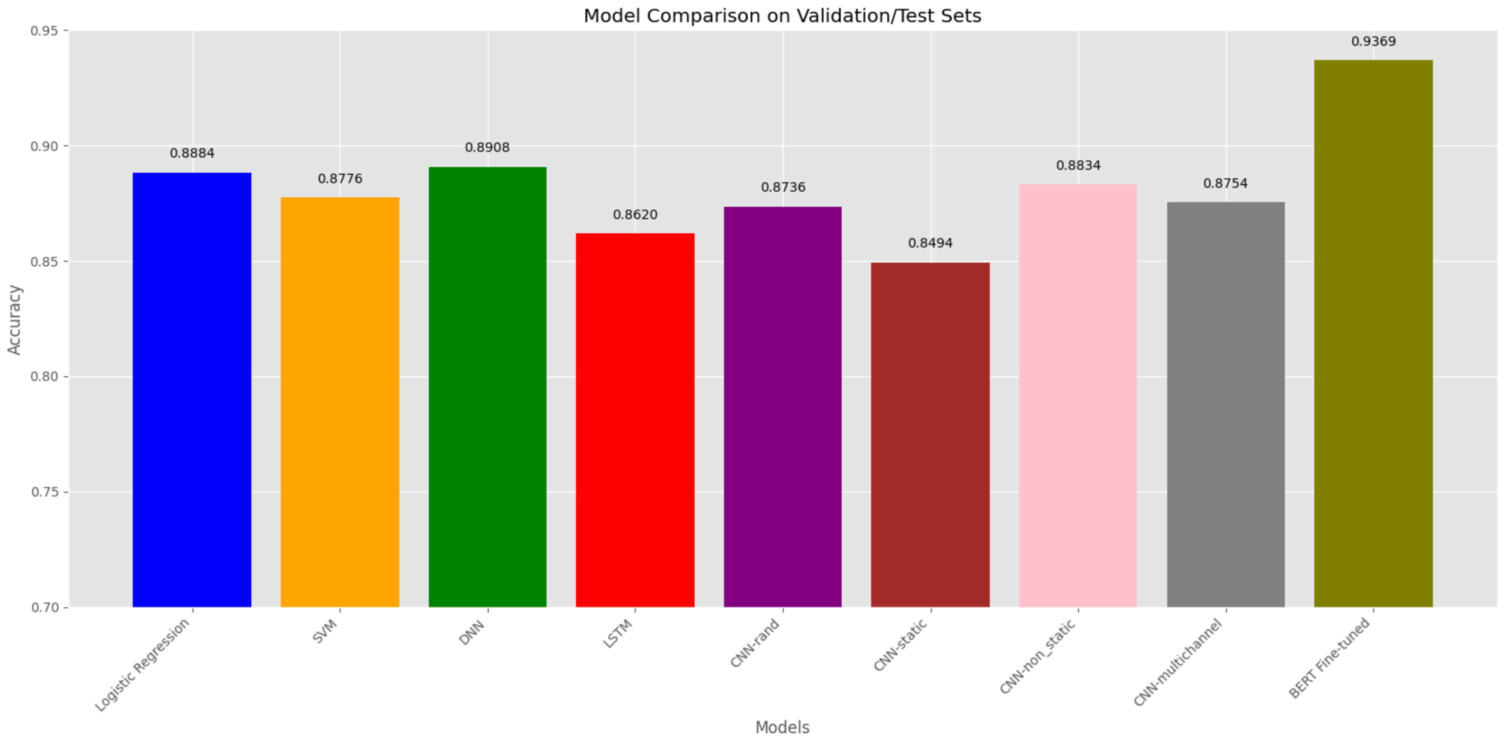
5. Results and Analysis

Model Comparison

Training history for CNN models

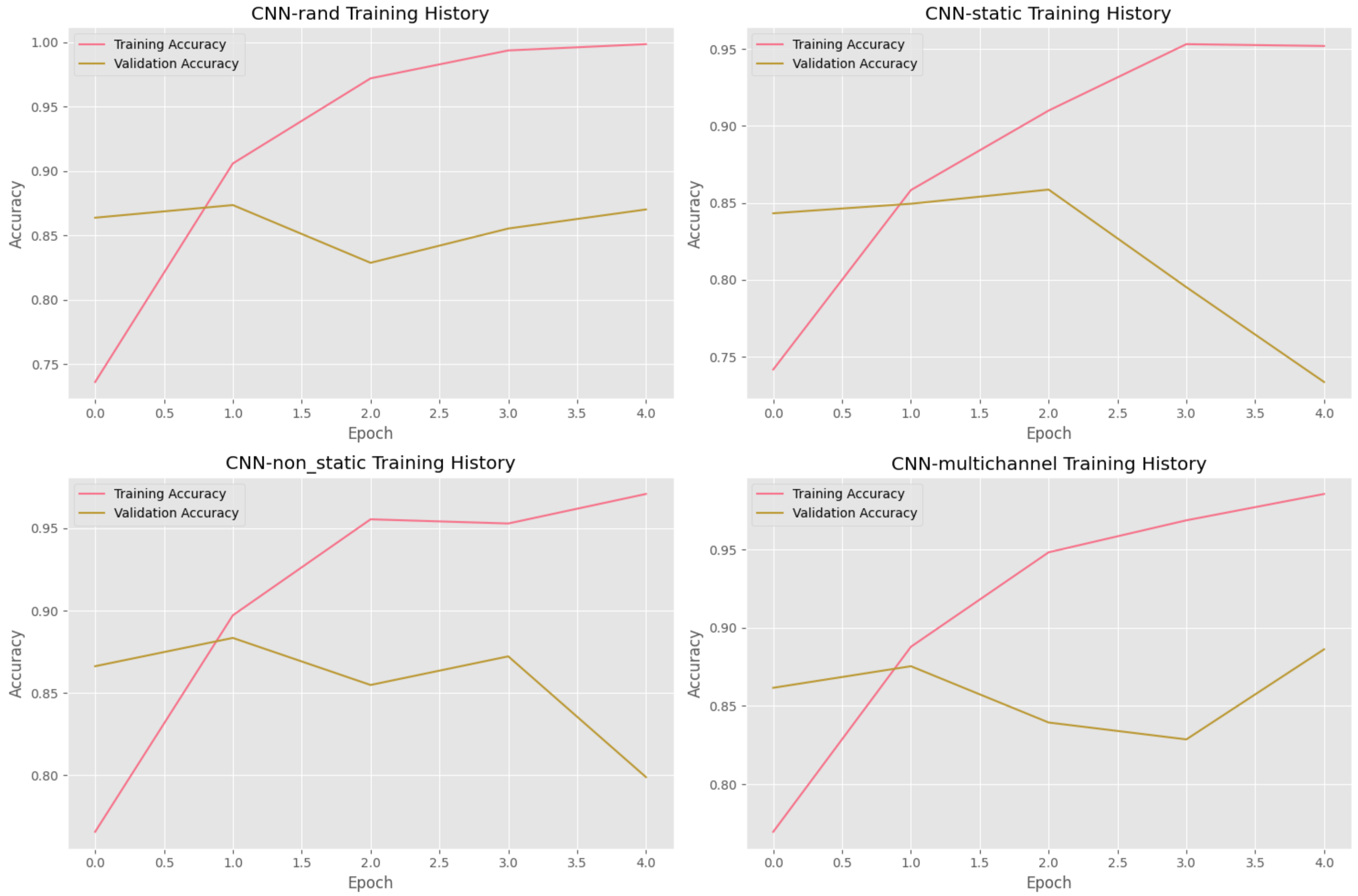
Model Summary

Model Performance Comparison



Rank	Model	Accuracy
1	**BERT Fine-tuned**	**0.9369**
2	DNN	0.8908
3	Logistic Regression	0.8884
4	CNN-non-static	0.8834
5	SVM	0.8776
6	CNN-multichannel	0.8754
7	CNN-rand	0.8736
8	LSTM	0.8620
9	CNN-static	0.8494

Training history for CNN models



Performance Analysis – Accuracy & Behaviour

Model family	**Key finding**	**Best accuracy**
Transformer (BERT)	Transfer learning wins	**90 %**
CNN-multichannel	Top CNN; beats single-channel	**88.6 %**
CNN-static	Pre-trained » random init	**84.9 %**
CNN-non-static	Fine-tune gives small lift	**88.3 %**
CNN-rand	Worst CNN	**87.0 %**
LSTM	Good but slow	**86.2 %**
DNN (TF-IDF)	Simple, solid	**87.4 %**
LogReg / SVM	Strong baselines	**88.8 / 89.0 %**

Performance Analysis – Cost & Use-Case Fit

Model	**Train-time**	**Inference**	**GPU-RAM**	**Best use-case**
BERT	3 h	20 ms	1.2 GB	Max accuracy, cloud
CNN-multichannel	15 min	3 ms	0.4 GB	Prod-grade balance
CNN-static	10 min	2 ms	0.3 GB	Low-cost, high-F1
LSTM	45 min	8 ms	0.5 GB	Sequential data
LogReg / SVM	30 s	1 ms	CPU	Edge / mobile

Findings

BERT: pay once, gain 2-3 pp accuracy.

CNN-static: 95 % of BERT quality at 1 % resources.

Traditional models: still competitive when GPUs are off-limits.