Comparative Analysis of DistilBERT and Traditional Classifiers for Multi-Class News Classification

Simge Cinar
Linköping University
732A81 Text Mining
simci637@student.liu.se

Abstract

Text classification is a crucial subfield of Natural Language Processing (NLP), aiming to categorize textual data into predefined categories. This project compares the performance of traditional machine learning methods, specifically the Naive Bayes classifier and Logistic Regression, with the more advanced DistilBERT model. Hyperparameter tuning is performed on the DistilBERT model, focusing on hidden size, dropout rate, and the number of epochs. The AG News dataset is used for this study, with subsamples of sizes 400, 1000, 5000, and 20000 to train the models. Results show that DistilBERT consistently outperforms traditional methods in all configurations. However, the gap in accuracy is notably larger when working with smaller datasets, highlighting that pre-trained model can be more useful with small training data sizes.

1 Introduction

Text classification is a fundamental task in Natural Language Processing (NLP) that involves assigning a given text sequence to one of several predefined categories (Sun et al., 2020).

In recent years, the rapid expansion of information on the Internet has led to an exponential increase in the volume of complex texts and documents, including news articles, highlighting the need for effective methods to organize and classify this vast content. Deep learning models play a pivotal role in natural language processing (NLP). The rapid advancements in deep learning have largely overshadowed traditional machine learning methods, offering more powerful and scalable solutions. Pre-trained deep learning models, such as BERT, have become integral to this shift, driving significant improvements in a wide range of NLP tasks (Wan, 2023).

This project aims to explore whether the advanced model DistilBERT can surpass traditional methods like the naive bayes classifier and logistic regression in performance, and whether fine-tuning its hyperparameters can further enhance its effectiveness.

2 Theory

2.1 Naive Bayes Classifier

The Naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' Theorem. It is a generative model that learns how the features of each class are distributed. It calculates the posterior probability of a class c given the features (words) in a document d (Jurafsky and Martin, 2025). It is expressed as:

$$\hat{c} = \underset{c \in C}{\operatorname{argmax}}\, P(d|c)P(c)$$

The likelihood term $P(d|c)$ is simplified by the naive assumption that the words in the document are conditionally independent, given the class c

$$P(d|c) = \prod_{i=1}^{n} P(w_i|c)$$

where w_i are the words in document d

The Naive Bayes classifier uses the 'bag of words' assumption, which means that the position or order of words in the document does not matter, only the frequency of each word matters (Jurafsky and Martin, 2025).

2.2 Logistic Regression

In multinomial logistic regression we want to label each observation with a class k from a set of K classes. Unlike naive bayes, which is a generative classifier, logistic regression is a discriminative classifier. It focuses on directly modeling the decision boundary between classes (Jurafsky and Martin, 2025). The formula for the prediction of the class is as follows:

$$\hat{c} = \underset{c \in C}{\text{argmax}}\ P(c|d)$$

Logistic regression typically uses gradient descent for optimization, with the cross-entropy loss function to quantify the difference between the predicted output and the actual target labels (Jurafsky and Martin, 2025).

2.3 DistilBERT Model

DistilBERT is a simplified, smaller version of the BERT model (Bidirectional Encoder Representations from Transformers). It is created by distilling BERT base model. Despite having 40% fewer parameters, DistilBERT retains 95% of BERT's performance, which makes it faster and more efficient (Sanh et al., 2020).

DistilBERT has the same general structure with BERT model. Each layer contains 12 attention heads with 768-dimensional hidden states. However, there are 6 hidden layers in DistiBERT whereas it is 12 in BERT model (Face, 2025a) (Face, 2025b).

BERT model is pre-trained using 'masked language model (MLM)' and 'Next Sentence (NSP) Prediction'. MLM randomly masks some tokens in the input and training the model to predict the original words based on the surrounding context. NSP trains BERT to understand relationships between sentences. Given two sentences, the model predicts whether the second sentence follows the first in the original text or is a randomly chosen sentence (Devlin et al., 2019). DistilBERT is trained with only MLM task (Sanh et al., 2020).

3 Data

The AG News dataset from Kaggle (Anand, 2025) is used for this project. The dataset consists of three columns: Class Index, Title, and Description. The Class Index column represents four categories: 1 for world, 2 for sports, 3 for business, and 4 for sci-tech. The dataset is balanced, with the training data containing 30,000 rows per class, totaling 120,000 rows, and the test data containing 1,900 rows per class, for a total of 7,600 rows. For this project, the Description column is combined with Title column and that concatenated column is used as the input text, while the Class Index column is used as the label.

Different numbers of rows are used for subsets which are 400, 1000, 5000 and 20000 to reduce computational complexity and observe how algorithms perform differently when it is trained with different number of samples.

Data preprocessing is as follows:

1. **Remove links:** Removes all <a> tags (links) and their content from the text using BeautifulSoup library.
2. **Clean extra spaces:** Replaces multiple spaces with a single space and remove leading or trailing spaces.
3. **Remove backslashes:** Any backslashes (\) are replaced with spaces
4. **Fix special characters:** HTML entities #39; (apostrophe) and #36; (dollar sign) are replaced with the actual characters

4 Method

The Naive Bayes classifier and logistic regression models were utilized as baseline approaches, while the DistilBERT model was selected as the more advanced and sophisticated model for comparison.

4.1 Naive Bayes Classifier

The CountVectorizer() function is used with its default parameters to vectorize the text data. The MultinomialNB() function from scikit-learn is then applied as the Naive Bayes classifier. These steps are streamlined using a pipeline, combining the vectorization and classification processes into a single workflow as implemented in Lab 2 in this course.

4.2 Logistic Regression

The text data is vectorized using TfidfVectorizer() function with default parameters. Logistic regression is implemented using LogisticRegression() from scikit-learn, and the maximum number of iterations is increased from 100 to 500 to ensure convergence when applied to the full dataset.

4.3 DistilBERT Model

DistilBERT is implemented from HuggingFace, with the initial code obtained from (Taunk, Sep 17, 2020). While the original implementation addressed a multi-label classification problem, it was adapted to suit the multi-class classification task in this study, such as changing loss function.

The DistilBertTokenizer is initialized using the pre-trained model 'distilbert-base-uncased' with

truncation enabled, lowercase conversion applied, padding set to True and maximum length for tokenizer is selected as 256.

A neural network built on top of the pre-trained DistilBERT model from the PyTorch library is used. The input text is passed through the DistilBERT layer, which generates 768-dimensional hidden states. These hidden states are then passed through a linear layer that maps the 768-dimensional vector to a selected hidden size dimension, which is tuned to values of 768 or 256 during training. The output of this layer is followed by a ReLU activation and a dropout layer for regularization. Finally, the transformed features are passed through a classifier linear layer to produce a 4-dimensional output representing the class probabilities.

The batch size for both training and test data is set to 16. Cross-entropy loss is chosen as the loss function to address the multi-class classification problem, while the Adam optimizer is used with a learning rate of 1e-5. The number of epochs is fine-tuned iteratively to optimize performance.

The parameters that are fine-tuned in the neural network's structure are as follows:

- **hidden layer size**: [256, 768]
- **dropout rate**: [0,2, 0.5]
- **number of epochs**: [3, 5]

The DistilBERT model was run with 8 configurations on Google Colab's T4 GPU. As the data size increased, the runtime also grew.

4.4 Evaluation Metrics

The classification_report and confusion_matrix functions from the sklearn package were used. Since the dataset is balanced, the macro average and weighted average values in the classification report are the same. The support for each class is 1900.

5 Results

The algorithm was trained on four subsamples from the original dataset, each containing a varying number of samples. The results for each method and for each subsample are summarized in the tables below. It is important to note that each DistilBERT model has 8 possible configurations, with the configuration yielding the highest and lowest accuracy presented in the table. The hyperparameters corresponding to the highest accuracy model for the DistilBERT model are as follows:

- Training sample size 400:
  Hidden Size=768, Dropout=0.2, Epoch=5
  Accuracy = 88.33%

- Training sample size 1,000:
  Hidden Size=256, Dropout=0.2, Epoch=5
  Accuracy = 89.11%

- Training sample size 5,000:
  Hidden Size=768, Dropout=0.2, Epoch=3
  Accuracy = 91.21%

- Training sample size 20,000:
  Hidden Size=768, Dropout=0.2, Epoch=3
  Accuracy = 93.05%

It was observed that with a smaller number of samples, the model performed better at epoch 5. However, as the sample size increased to 5000 and above, optimal performance was achieved at epoch 3.

	F1-score		
	Naive Bayes	Logistic Regression	DistilBERT
World	0.82	0.79	0.88
Sports	0.87	0.86	0.96
Business	0.74	0.73	0.83
Sci-tech	0.74	0.73	0.86
F1-score avg	0.79	0.78	0.88
Accuracy (*lowest*)	0.79	0.78	0.88 (*0.86*)

Table 1: Results with sample size 400

	F1-score		
	Naive Bayes	Logistic Regression	DistilBERT
World	0.86	0.84	0.90
Sports	0.92	0.89	0.97
Business	0.79	0.78	0.84
Sci-tech	0.79	0.78	0.86
F1-score avg	0.84	0.82	0.89
Accuracy (*lowest*)	0.84	0.82	0.89 (*0.88*)

Table 2: Results with sample size 1,000

	F1-score		
	Naive Bayes	Logistic Regression	DistilBERT
World	0.89	0.89	0.92
Sports	0.95	0.93	0.97
Business	0.84	0.83	0.87
Sci-tech	0.85	0.84	0.89
F1-score avg	0.88	0.87	0.91
Accuracy (*lowest*)	0.88	0.87	0.91 (*0.90*)

Table 3: Results with sample size 5,000

	F1-score		
	Naive Bayes	Logistic Regression	DistilBERT
World	0.90	0.91	0.94
Sports	0.96	0.95	0.98
Business	0.85	0.86	0.90
Sci-tech	0.87	0.87	0.90
F1-score avg	0.89	0.90	0.93
Accuracy (*lowest*)	0.90	0.90	0.93 (*0.92*)

Table 4: Results with sample size 20,000

	F1-score		
	Naive Bayes	Logistic Regression	DistilBERT
World	0.90	0.92	-
Sports	0.96	0.97	-
Business	0.86	0.89	-
Sci-tech	0.88	0.89	-
F1-score avg	0.90	0.92	-
Accuracy	0.90	0.92	-

Table 5: Results with sample size 120,000 (full data)

From the tables above, it can be observed that accuracy improves consistently as the sample size increases across all models. The average F1-score remains closely aligned with accuracy for all models and sample sizes, indicating balanced performance across classes. 'Business' and 'Sci-tech' classes has lower F1-score compared to other classes in each model. Additionally, there is minimal variation in performance among the eight different configurations of the DistilBERT model.

It should be noted that the gap between baseline models and DistilBERT is higher when sample size is smaller. Also DistilBERT model can reach to 93.05% accuracy with sample size 20,000 whereas naive bayes or logistic regression cannot reach to that accuracy even with full model. The confusion matrices for the models can be seen in the Appendix.

6 Discussion

The results demonstrate that due to its pre-trained nature, DistilBERT can achieve high accuracy and F1-scores even with a very small dataset, outperforming both Naive Bayes and Logistic Regression classifiers. These traditional models tend to struggle with smaller datasets, while DistilBERT's ability to leverage pre-trained knowledge enables it to perform exceptionally well even with limited data.

While Paper "*NoisyAG-News: A Benchmark for Addressing Instance-Dependent Noise in Text Classification*" (Huang et al., 2024) primarily focuses on noisy labels, it also provides insights into performance with clean labels. Its experiment report on the AG News dataset shows an accuracy range of 0.93 to 0.94 using models such as BERT, RoBERTa, and XLNet, with a training sample size of 50,000. This highlights that more complex models, even with larger datasets, do not significantly improve performance compared to DistilBERT model.

Limitations: The DistilBERT model could not be applied to the entire AG news dataset due to computational constraints, particularly in terms of processing time.

7 Conclusion

It can be concluded that while traditional methods such as Naive Bayes and logistic regression can offer valuable insights, their performance tends to be heavily dependent on the size of the training data. In contrast, pre-trained models like DistilBERT, with their robust transfer learning capabilities, demonstrate a clear advantage, maintaining strong performance even when the available training data is limited. This highlights the potential of pre-trained models to outperform traditional approaches, particularly in scenarios with restricted data availability.

References

Aman Anand. 2025. Ag news classification dataset. https://www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset. Accessed: 2025-01-09.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Hugging Face. 2025a. Bert. Accessed: 2025-01-18.

Hugging Face. 2025b. Distilbert. Accessed: 2025-01-18.

Hongfei Huang, Tingting Liang, Xixi Sun, Zikang Jin, and Yuyu Yin. 2024. Noisyag-news: A benchmark for addressing instance-dependent noise in text classification. *Preprint*, arXiv:2407.06579.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released January 12, 2025.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification? *Preprint*, arXiv:1905.05583.

Dhaval Taunk. Sep 17, 2020. Finetune distilbert for multi-label text classsification task.

Zhongwei Wan. 2023. Text classification: A perspective of deep learning methods. *Preprint*, arXiv:2309.13761.

A Appendix

A.1 Confusion Matrices Using 400 Training Samples

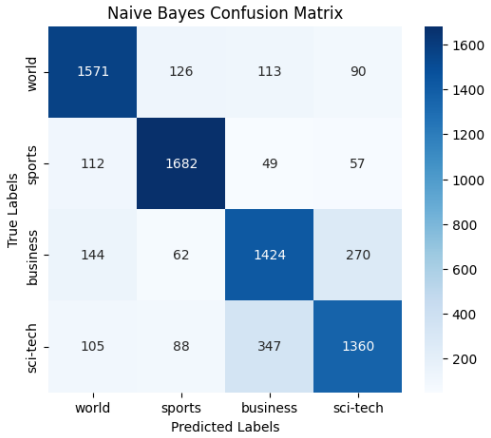

Figure 1: naive bayes confusion matrix with sample size 400

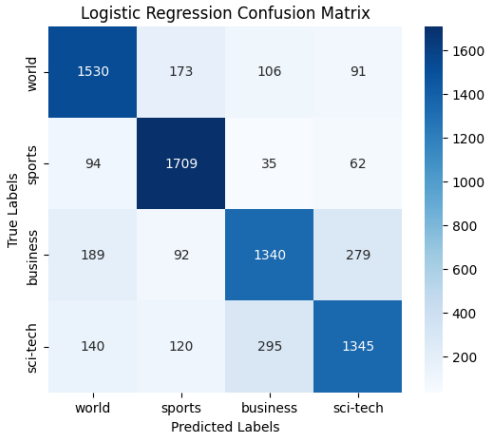

Figure 2: logistic regression confusion matrix with sample size 400

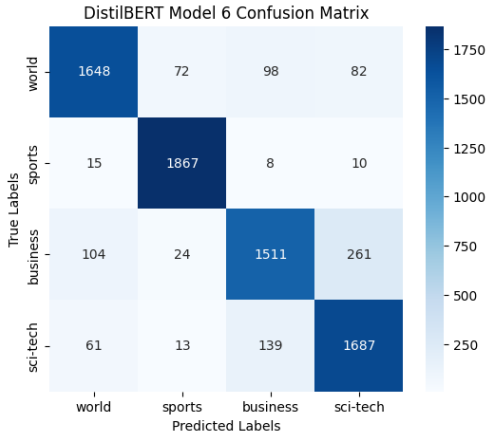

Figure 3: DistilBERT confusion matrix with sample size 400

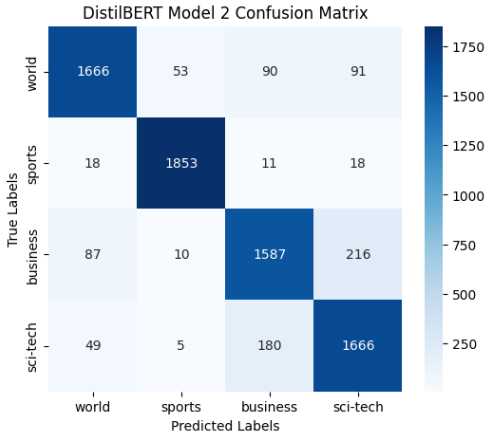

Figure 6: DistilBERT confusion matrix with sample size 1,000

A.2 Confusion Matrices Using 1,000 Training Samples

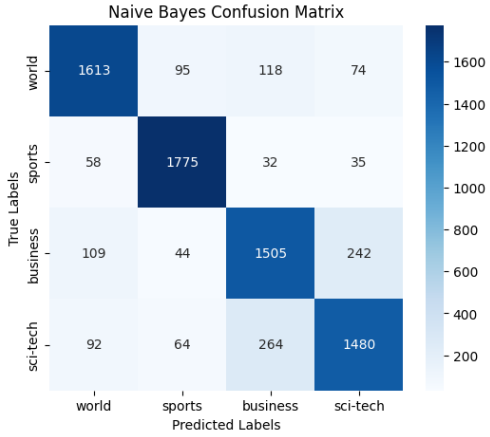

Figure 4: naive bayes confusion matrix with sample size 1,000

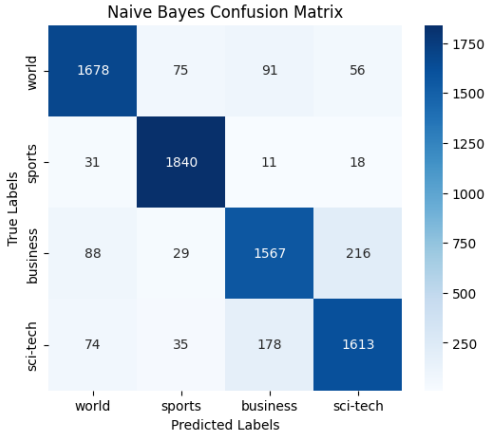

Figure 7: naive bayes confusion matrix with sample size 5,000

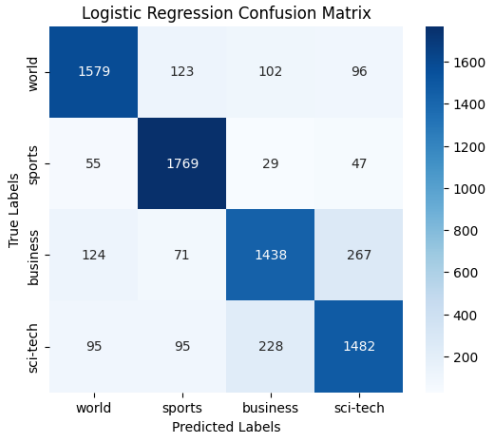

Figure 5: logistic regression confusion matrix with sample size 1,000

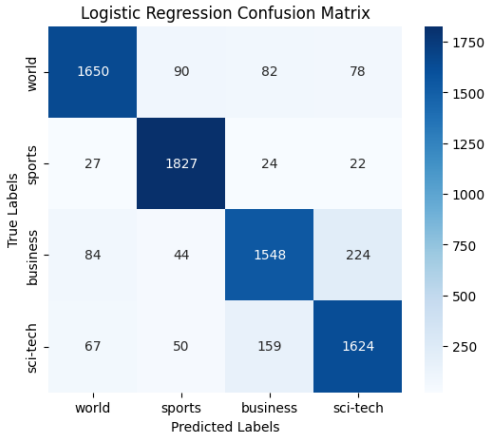

Figure 8: logistic regression confusion matrix with sample size 5,000

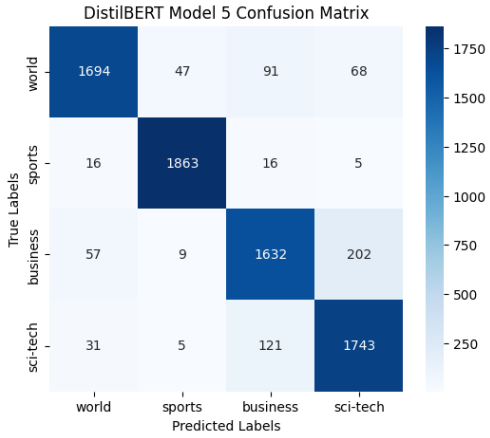

Figure 9: DistilBERT confusion matrix with sample size 5,000

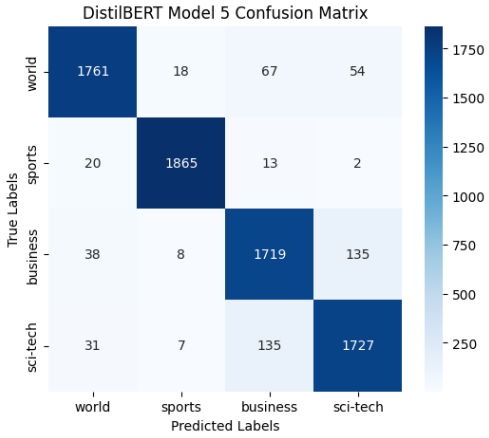

Figure 12: DistilBERT confusion matrix with sample size 20,000

A.4 Confusion Matrices Using 20,000 Training Samples

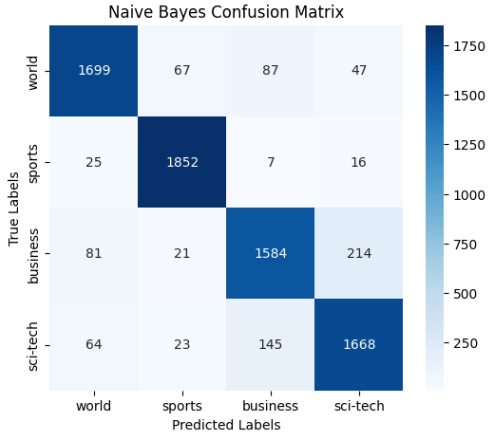

Figure 10: naive bayes confusion matrix with sample size 20,000

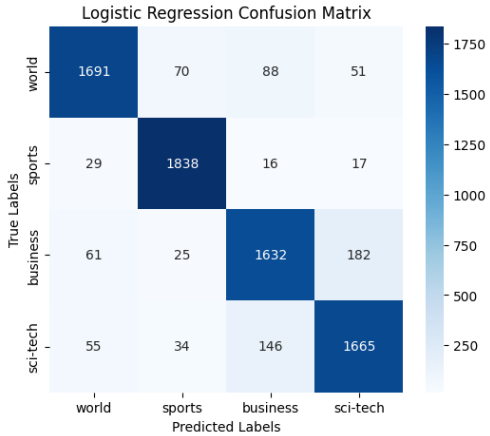

Figure 11: logistic regression confusion matrix with sample size 20,000