NLP Topic Classification with Classical ML and Modern Transformers

Authored By.
By Brandyn Ewanek
Matriculation: 9216750
Customer ID: 10664359

1. Text Classification for Topic Modeling

The goal of this project is to extend beyond standard text classification to analyze and understand the bias present within distinct news categories.

1.1. Situational Analysis and Problem Definition

In the contemporary media landscape, news content is frequently shaped by various pressures that can introduce subtle or overt bias. This presents a significant challenge for consumers seeking to form an objective understanding of events. To better understand this phenomenon, this project will first focus on developing a robust system for classifying news articles from *20 NewsGroup***s** dataset available ok sci-kit learn, into distinct categories, such as science, technology, and religion.

However, the ability to perform these classifications is a foundational step toward a more complex objective. The central problem this project seeks to address is understanding the nature of the bias present within each news category. Once an accurate classification model is established, the subsequent goal is to analyze the categorized articles to gain a clearer perspective on the specific biases inherent to each topic. The ultimate aim is to develop a more prudent and analytical framework for consuming news in the future.

1.2. Project Objectives and Intended Outcomes

In direct response to the problem defined above, the primary objective of this project is to develop and evaluate a multi-faceted NLP system capable of both classifying news articles by topic and analyzing the sentiment and potential biases within those topics. To achieve this overarching aim, the project is guided by a set of specific, measurable, and achievable secondary objectives:

- **Objective 1:** To train and evaluate classical machine learning models—specifically Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting..
- **Objective 2:** To identify the best-performing classical model from the initial evaluation and further optimize through hyperparameter tuning.
- **Objective 3:** Gauge the zero-shot classification performance of a pre-trained DistilBERT model.
- **Objective 4**: Preform Fine-Tuning and Evaluate the DistilBert model
- **Objective 5:** Compare metrics of all models and decide on the best overall performance model for inclusion in final new classification system.
- **Objective 6:** Perform Sentiment Analysis with Vader Sentiment.
- **Objective 7**: Complete compiled ‘product’ that classifies given text, provides sentiment and reason for sentiment.

The tangible outcome of this project is a comprehensive analytical framework. This "product" will consist of a series of trained models and an NLP pipeline designed to classify news articles and analyze their content. The framework will provide a comparative assessment of various modeling techniques, thereby offering a practical tool for more critical media consumption.

1.3. Methodological Approach and Preparatory Work

This project employs a structured, comparative experimental methodology, designed to systematically evaluate a range of text classification techniques from classical machine learning to modern deep learning architectures. This approach was chosen to ensure a thorough and data-driven selection of the optimal model for the final analysis task. The methodology is divided into three distinct phases:

1. **Baseline Modeling with Classical Algorithms:** This involves training and evaluating Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting classifiers. For these models, text data is converted into a numerical format using the Term Frequency-Inverse Document Frequency (TF-IDF) vectorization technique.
2. **Advanced Modeling with Transformer Architectures:** The second phase explores the capabilities of state-of-the-art transformer models. This begins by testing an untrained model and then based on results, fine-tuning DistillBERT on the current dataset to evaluate its performance in comparison to classical machine learning models.
3. **Sentiment and Bias Analysis:** The final phase involves a sentiment analysis analysis with final classification. This sentiment will help to educate readers of bias present in articles.

The text data was prepared with TF-IDF, for classical classifiers like Logistic Regression , and ensemble methods like Random Forest and Gradient Boosting. The primary data sources 20 Newsgroups dataset. The computational environment included the installation of key libraries such as scikit-learn for classical modeling and Hugging Face's transformers for the deep learning components.

2. Project Execution and Critical Evaluation

2.1. Project Planning and Resource Management

The project was executed using a standard personal computer, with cloud-based resources leveraged for computationally intensive tasks. Key resources included:

- **Hardware**: Intel Core i7 CPU, 16GB RAM. GPU resources were accessed via Google Colab for fine-tuning the transformer model.
- **Software**: Python 3.9, Jupyter Notebooks for interactive development.
- **Libraries**: scikit-learn for classical ML models and TF-IDF, pandas for data manipulation, nltk for text preprocessing, and Hugging Face's transformers and torch for deep learning components.
- **Data**: Publicly available 20 Newsgroups datasets, accessed via scikit-learn's dataset loaders.

Potential Risk: The primary risk was the computational demand of fine-tuning a transformer model, which was mitigated by using a smaller, more efficient model (DistilBERT) and leveraging free-tier GPU access on Google Colab.

2.2. Implementation and Process Documentation

An initial exploratory data analysis was performed to understand class balance by visualizing the category distributions in the 20 Newsgroups dataset with countplots. To gain insight into the textual content, word clouds were generated both before and after stopword removal to highlight the impact of this preprocessing step on revealing more meaningful terms.

A core part of the implementation was the development of a preprocess_text function. This function standardizes the raw text by converting it to lowercase, removing punctuation, filtering out common English stopwords, and applying Porter Stemming to reduce words to their root forms.

The preprocessed text was then converted into numerical features using the TF-IDF (Term Frequency-Inverse Document Frequency) vectorization technique. This method creates a matrix where each row represents a document and each column represents a word, with the cell value being the TF-IDF score that reflects the word's importance to that document within the corpus.
Next, several classical machine learning models were trained: Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting. Each was evaluated using a classification report, which provides key metrics like Accuracy, Precision, Recall, and F1-score.

Finally, the process of hyperparameter tuning was initiated for the Random Forest model to optimize its performance. This involved iterating through different values for the n_estimators parameter and plotting the train and test scores to identify the value that best balances performance without overfitting.

2.3. Analysis of Models

The performance of each model was measured using standard classification metrics, including Accuracy, Precision, Recall, and F1-score. These metrics provide a comprehensive view of each model's ability to correctly classify news articles into their respective topics. The comparative performance of the models is summarized in the table below.

Train Dataset	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.9772	0.9775	0.9772	0.9772
Decision Tree	1.0	1.0	1.0	1.0
Random Forest	1.0	1.0	1.0	1.0
Gradient Boosting	0.9813	0.9814	0.9813	0.9812

Test Dataset	Accuracy	Precision	Recall	F1-score
Logistic Regression	0.8307	0.8341	0.8307	0.8295

Decision Tree	0.5643	0.5170	0.5643	0.5664
Random Forest	0.7768	0.7882	0.7768	0.7725
Gradient Boosting	0.7434	0.7771	0.7434	0.7528

Summary: Logistic Regression performs exceptionally well on the complex task of text classification. It shows the highest test accuracy of 0.83 which is considerably better than other models. This is surprising as a simple model like Logistic Regression shouldn't out perform ensemble methods on complex tasks like this. All models show significant signs of overfitting, seen by high train scores in comparison to test scores.

Based on this initial evaluation, the Random Forest model was selected as the most promising candidate for further optimization as logistics regression has few hyperparameters.

Random Forest Hyper Parameter Tuning

Rather than a computationally expensive grid search, a more iterative, visual approach was used to tune key hyperparameters. The parameters n_estimators, max_depth, and min_samples_split were tuned by iterating through a range of values and plotting the corresponding training and test scores.
This visual analysis revealed several key insights:

- **Number of Estimators**: The test score showed a steady improvement as the number of trees (n_estimators) increased, beginning to plateau around 260-280 trees.
- **Max Depth**: The test score peaked when max_depth was 13. Deeper trees caused the training score to continue rising while the test score declined, a clear sign of overfitting.
- **Min Samples Split**: This parameter showed less dramatic impact, but a value around 10 provided a good balance, preventing trees from creating splits based on few rows.

Based on this analysis, the final hyperparameters were selected: n_estimators=260, max_depth=13, and min_samples_split=10. The performance of this tuned model is as follows:

Dataset	Accuracy	Precision	Recall	F1-score
Train	0.8447	0.8579	0.8447	0.8399
Test	0.7315	0.7531	0.7315	0.7202

The tuning process successfully reduced the overfitting problem, however we were unable to achieve a higher test score than with the default hyperparameters.

Transformers

The next phase we gauged the zero-shot classification performance of a pre-trained transformer model. For this, the Hugging Face transformers library was used to deploy a distilbert-base-uncased model within a zero-shot classification pipeline. This technique tests the

model's ability to classify text into categories it has never been explicitly trained on, relying instead on its general language understanding.

The pipeline's output provides predicted label names, which are then mapped back to their corresponding numerical indices for evaluation against the test set's ground truth labels. The test performance of the DistilBERT model in this zero-shot setting is presented below.

Model	Accuracy	Precision	Recall	F1-score
DistilBERT (Zero-Shot)	0.0400	0.0677	0.0400	0.0132

The results clearly indicate that the zero-shot approach was not effective for this specific, multi-class classification task. An accuracy of approximately 4% on a 20-class problem is only slightly better than random chance. This outcome is not unexpected. The general-purpose distilbert-base-uncased model, without any task-specific fine-tuning, lacks the specialized knowledge to distinguish between the nuanced topics of the 20 Newsgroups dataset.

Fine Tuning

Following the zero-shot experiment, the DistilBERT model was fine-tuned on the 20 Newsgroups dataset. This process adapts the pre-trained model to the specific vocabulary and context of the task. The process involved several key steps:

1. **Data Conversion**: The data was converted into a Hugging Face Dataset object, the standard format for the Trainer API.
2. **Tokenization**: A distilbert-base-uncased tokenizer was used to convert the raw text into numerical IDs, with padding and truncation applied to ensure uniform sequence length.
3. **Model Loading**: The AutoModelForSequenceClassification class was used to load the DistilBERT model, configured with the correct number of output labels.
4. **Training**: The model was trained for 4 epochs using the Trainer API, with a learning rate of 2e-5 and a batch size of 16. A custom compute_metrics function was supplied to evaluate performance on the test set at the end of each epoch.

The fine-tuning process yielded a significant improvement in performance. The final evaluation on the test set produced the following results:

Model	Accuracy	Precision	Recall	F1-score
DistilBERT (Fine-Tuned)	0.8561	0.8589	0.8561	0.8563

Final Model Comparison and Selection:

To determine the best overall model, the performance of the fine-tuned DistilBERT is compared against the optimized classical models.

Model	Test Accuracy	Test F1-score (Weighted)
Tuned Logistic Regression	0.8307	0.8295
Tuned Random Forest	0.7315	0.7202
Fine-Tuned DistilBERT	**0.8561**	**0.8563**

The results show DistilBert was able to perform slightly better that the simple Logistic Regression, therefore DistilBert will be used in the final system.

2.4 Sentiment Analysis

By analyzing the sentiment of news articles to understand its distribution across different topics the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool was chosen. Unlike deep learning models that require training, VADER is a lexicon and rule-based tool particularly effective at capturing sentiment. Its key advantage is the compound score, a single, normalized metric ranging from -1 (most negative) to +1 (most positive) that provides a holistic measure of sentiment intensity. This allows for a more nuanced comparison of sentiment across categories than a simple positive/negative label.

The implementation involved applying a function using VADER's polarity_scores method to each article. This function assigns a categorical label (POSITIVE, NEGATIVE, or NEUTRAL) based on the compound score, while retaining the score itself for quantitative analysis. The analysis of the results provides a multi-faceted view of the sentiment landscape within the dataset.
The distribution of sentiment labels across categories shows that most topics are predominantly classified as 'POSITIVE'. However, a significant number of 'NEGATIVE' articles are present in contentious topics like talk.politics.mideast and talk.politics.guns, providing a high-level confirmation of their controversial nature.

A more granular analysis is possible by examining the average compound score for each category:

Category	Average Compound Score
comp.graphics	0.6094
misc.forsale	0.5413
rec.sport.baseball	0.5179
.....	.....

alt.atheism	0.1506
talk.politics.misc	0.1174
talk.politics.mideast	-0.2844
talk.politics.guns	-0.3237

This ranking clearly separates the categories, with technical and hobbyist topics like comp.graphics and rec.sport.baseball showing the most positive sentiment. Conversely, the political discussion groups, talk.politics.mideast and talk.politics.guns, are the only categories with a negative average sentiment, aligning with intuitive expectations.

The distribution of these scores within each category further illuminates the sentiment dynamics. The histograms reveal that the political topics have a sentiment distribution heavily skewed towards negative values. In contrast, categories like rec.autos and comp.graphics show distributions centered firmly in the positive range, with very few strongly negative articles. This detailed distributional analysis, made possible by VADER's compound score, provides a much richer and more insightful picture of the sentiment associated with each topic than a simple categorical label could achieve.

2.5 Integrated Analysis System

The culmination of this project is the development of a single, integrated function, analyze_document, which encapsulates the entire analysis pipeline. This function represents the project's final "product," transforming the series of experimental models into a practical and reusable tool.

The system's workflow is as follows:

1. **Topic Classification**: The extracted text is passed to the fine-tuned DistilBERT classification pipeline (category_predictor) to determine the document's topic.
2. **Sentiment Analysi**s: The same text is then analyzed by the sentiment pipeline (sentiment_analyzer) to get a sentiment label and score.
3. **Comparative Analysis**: The system retrieves the pre-calculated average sentiment for the predicted category and computes the deviation of sentiment from the norm.
4. **Generative Explanation**: In the most innovative step, if the sentiment deviation is significant, the function constructs a prompt for a generative AI model (Gemini 1.5 Flash). This prompt includes the text, the category, the sentiment scores, and the deviation, asking the model to provide a brief, human-like explanation for why the sentiment might be unusual.
5. **Output**: Finally, the function consolidates all this information into a single, formatted report, providing the user with a comprehensive, multi-layered understanding of the document.

This integrated function successfully demonstrates the practical application of the project's findings, creating a powerful tool that goes beyond simple classification to offer comparative and interpretive insights.

2.6 Reflection on Process, Performance, and Theoretical Application

This project's execution provided a valuable opportunity to assess the practical application of various theoretical NLP models. The structured, comparative methodology proved highly effective, creating a logical progression from simple baselines to complex, state-of-the-art techniques.

The initial phase, centered on classical machine learning algorithms, demonstrated the practical utility of the TF-IDF theoretical model. TF-IDF's principle of weighting words by their relative rarity was effective enough to establish a strong performance baseline, with the tuned Random Forest model achieving 73.2% accuracy. The hyperparameter tuning process was a direct practical application of the bias-variance tradeoff theory; the visualizations clearly showed how constraining tree depth (max_depth) prevented the model from overfitting to the training data, a common pitfall of ensemble methods. However, this phase also highlighted the inherent limitations of the bag-of-words approach: its inability to capture context ultimately capped the model's performance.

The transition to transformer-based models marked a significant theoretical shift from word frequency to contextual understanding. The zero-shot classification experiment, while theoretically promising, its practical application resulted in a near-random accuracy of 4%. This outcome powerfully demonstrated that a general-purpose language model's theoretical capabilities do not automatically transfer to a specialized, multi-class domain without adaptation.

The subsequent fine-tuning of the DistilBERT model, the process directly leveraged the principle of transfer learning, adapting the model's vast pre-trained knowledge to the specific nuances of the 20 Newsgroups dataset. The dramatic performance increase to 85.6% accuracy provided definitive evidence that for this task, the theoretical advantages of contextual embeddings are vastly superior to frequency-based methods.

From a process perspective, the phased plan was executed successfully. The primary challenge encountered was the domain mismatch of the pre-trained sentiment analysis model. This deviation from the expected outcome underscored a key lesson: the performance of pre-trained tools is highly contingent on the similarity between their original training data and the new application domain. This practical insight reinforces the theoretical importance of domain adaptation in NLP.

3. Synthesis and Future Implications

3.1. Summary of Findings and Key Conclusions

This project successfully navigated a comprehensive, multi-stage experimental process to develop a robust text classification system and apply it to sentiment analysis. The investigation yielded several key findings and conclusions that directly address the project's objectives.

First, the comparative analysis of classification models demonstrated a clear performance hierarchy. While classical machine learning models like the tuned Random Forest provided a respectable baseline accuracy of 73.2%, the fine-tuned DistilBERT transformer model was unequivocally superior, achieving a test accuracy of 85.6%. *The primary conclusion from this is that for nuanced, multi-class text classification, the contextual understanding inherent to transformer architectures slightly outperformed the word-frequency-based approach of TF-IDF.* The initial zero-shot experiment, which yielded only 4% accuracy, further underscored this point, proving that task-specific fine-tuning is essential for adapting a general-purpose language model to a specialized domain.

Second, the sentiment analysis phase, which pivoted to using the VADER lexicon-based tool, provided a more nuanced understanding of sentiment in the specialized domain of newsgroups. The average compound scores revealed a logical and wide-ranging sentiment distribution, from positive scores in technical and hobby categories (e.g., comp.graphics) to highly negative scores in contentious political topics (e.g., talk.politics.guns). This leads to the *conclusion that for an out-of-the-box analysis on a domain without* sp*ecific sentiment training data, a rule-based tool like VADER can be more effective and insightful than a pre-trained deep learning model tuned on a mismatched domain.* While this approach lacks a formal accuracy metric without ground-truth labels, the intuitive results suggest it is a valid method for establishing a baseline sentiment understanding.

Finally, the project culminated in the successful design of an integrated document analysis system. This system serves as the project's final "product," combining the best-performing classification model, DistilBert, with sentiment analysis, VaderSentiment, and a generative AI component for interpretation. This demonstrates that the individual models and analyses developed throughout the project can be synthesized into a practical, value-added application, thereby fulfilling the project's overarching goal.

3.2. Recommendations

While the integrated analysis system developed in this project is a robust proof-of-concept, several avenues for future development could enhance its capabilities and practical utility.

First, the classification model's scope could be significantly expanded. The current DistilBERT model is fine-tuned exclusively on the 20 Newsgroups dataset. To create a more versatile tool, the model could be further trained on a diverse range of modern news article datasets from various sources. This would improve its ability to generalize to a wider array of topics and writing styles, making the system more robust for real-world applications.

Second, the project could be transformed into a practical application for general use. The current system exists as a Python function, requiring technical expertise to operate. The

development of an easy-to-use graphical user interface (GUI) or a simple web application would be a critical next step. Such an interface would allow non-technical users to easily upload a document or paste text and receive the analysis report. This would democratize the tool, making it a practical and accessible resource for anyone looking to better understand the potential biases and sentiment within their news sources.

3.3. Next Steps

This project provided a comprehensive, hands-on application of theoretical NLP concepts, yielding significant insights for future academic and professional development. The structured progression from classical machine learning to advanced transformer architectures has solidified a deep, practical understanding of the field's evolution and the specific trade-offs between different modeling paradigms.

One of the most critical takeaways is the tangible impact of transfer learning. Witnessing the dramatic performance delta between the zero-shot (4% accuracy) and fine-tuned (85.6% accuracy) transformer models provided a powerful, practical demonstration of this core deep learning principle. This experience has underscored the importance of domain adaptation and has directly shaped a future professional focus on fine-tuning and customizing large language models for specialized enterprise applications.

Furthermore, the challenges encountered during the sentiment analysis phase were highly instructive. The pivot to VADER after observing the likely domain mismatch of a standard pre-trained pipeline highlighted the risks of applying generic models to specific domains. It spurred an interest in more advanced techniques for bias detection and sentiment analysis that are robust to domain shifts. This practical experience has solidified my intent to pursue further specialization in the areas of model robustness, interpretability, and the responsible deployment of AI systems. The project has not only developed technical skills but has also cultivated a more critical and nuanced perspective on the application of NLP in real-world scenarios.

References

- Hutto, C.J. & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International AAAI Conference on Weblogs and Social Media.*
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

- Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5-32.
- Wolf, T., Debut, L., Sanh, V., et al. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, 1189-1232.