Enhancing Sentiment Analysis through Domain Adaptation and Ensemble Learning

V. Prema[1], Dr. V. Elavazhahan[2]

[1]*Research Scholar Annamalai University Chidambaram premaveluchamy78@gmail.com*
[2]*Assistant Professor Government Arts college Vadalur thillaieleval@gmail.com*

Sentiment analysis, the computational study of opinions and emotions expressed in text, is a critical component in many applications ranging from customer feedback analysis to social media monitoring. However, achieving robust sentiment analysis across diverse domains remains a significant challenge due to variations in language use, context, and domain-specific vocabulary. This paper presents a novel approach to enhancing sentiment analysis through the synergistic application of domain adaptation and ensemble learning techniques, incorporating both homogeneous and heterogeneous bootstrapping. We leverage state-of-the-art pretrained language models, fine-tuning them on specific domains to improve their sensitivity to domain-specific nuances. Our methodology includes comprehensive strategies for selecting and preprocessing data to facilitate effective domain adaptation. To further enhance model performance and robustness, we employ both homogeneous bootstrapping (using the same model architecture with different initializations) and heterogeneous bootstrapping (using diverse model architectures) as part of our ensemble learning framework. This dual approach allows us to capture a wide range of patterns and interactions within the data. The findings underscore the potential of combining domain adaptation with both homogeneous and heterogeneous bootstrapping techniques to address the inherent challenges of sentiment analysis in diverse and dynamic environments.

Keywords: Sentiment Analysis, Social Media, Learning techniques.

1. Introduction

In the era of big data, sentiment analysis has emerged as a critical tool for extracting valuable insights from textual data across various applications, including customer feedback analysis, social media monitoring, and market research. By understanding public sentiment, businesses and organizations can make informed decisions, improve customer satisfaction, and manage brand reputation effectively. However, the accuracy and reliability of sentiment analysis systems are often compromised when applied to text from diverse domains due to variations in language use, context, and domain-specific terminology. Traditional sentiment analysis models typically perform well when trained and tested within the same domain but struggle with cross-domain generalization. This limitation presents a significant challenge in real-world applications, where models must often handle text from multiple domains, such as product reviews, social media posts, and news articles. Addressing this issue requires innovative approaches that can adapt to new domains with minimal performance degradation.

In this paper, we propose a novel approach to sentiment analysis that synergistically combines domain adaptation and ensemble learning. By utilizing both homogeneous and heterogeneous bootstrapping techniques, we create an ensemble of models that work together to deliver robust and accurate sentiment predictions across diverse domains.

Our contributions are as follows:

1. We develop comprehensive strategies for domain adaptation, including data selection and preprocessing, to fine-tune pretrained models effectively.

2. We implement ensemble learning techniques that incorporate both homogeneous and heterogeneous bootstrapping to enhance model robustness.

3. We conduct extensive evaluations across multiple benchmark datasets to demonstrate the effectiveness of our approach in cross-domain sentiment analysis.

4. We analyze the practical implications of our findings and discuss their potential impact on real-world applications.

The remainder of this paper is organized as follows. Section 2 reviews related work in sentiment analysis, domain adaptation, and ensemble learning. Section 3 details our methodology, including the techniques for domain adaptation and the ensemble learning framework. Section 4 describes the experimental setup and evaluation metrics. Section 5 presents the results and analysis, while Section 6 discusses the key findings, limitations, and practical applications. Finally, Section 7 concludes the paper and suggests directions for future research.

2. Related Work

Ensemble Learning

Several techniques for constructing heterogeneous ensembles are applied and comparatively evaluated by Kazmaier J, Van Vuuren JH [1] (2022) across four different domains of benchmark sentiment classification datasets, revealing median performance improvements over individual models. Alsayat A [2] (2022) established a sentiment analysis framework using deep learning and ensemble techniques tailored for COVID-19-related social media data. He used two key model development stages include creating a baseline classifier, such as an LSTM network, and proposing an ensemble model that combines various classifiers for enhanced performance Mohammed A, Kora R.[3] (2022) capitalizes on the variability of Tier-0 classifiers and their predictions, utilizing them to construct effective ensemble models in Tier-1 through the training of shallow meta-classifiers.

Phan et al [4] (2020) introduced a novel methodology utilising a feature ensemble model to analyse tweets with fuzzy sentiment. This approach considers several variables including lexical, word-type, semantic, location, and sentiment polarity of words. The method has been tested using actual data, and the results demonstrate its effectiveness in enhancing the performance of tweet sentiment analysis, as measured by the F1 score. word embeddings have been utilized as an alternative to the manual techniques [5][6] Fouad et al (2018) combined Bag of words with Lexicon, Emoticons and Part of speech (PoS) and gave better results with ensemble classifier.[7] The Continuous Bag-of-Words (CBOW) and Skip-gram models are two popular variants of the Word2Vec algorithm.[8] Pennington J (2014) introduced Glove and the extraction method outperforms CBoW and SkipGram with 93.2 F1 score [9]

Hybrid Approaches

Parveen (2023) proposed LTF-MICF with GARN (RNN and Attention) architecture [10] Kaur G et al(2023) Combined Review features and aspect features with LSTM classifier [11] Lin CH et al (2023) combined BERT and Distil BERT with BiLSTM &TCN classifier [12] Want et al (2020) combined textual information with Sentiment diffusion [13] Stochastic word embedding technique [14] was introduced by Hao et al (2020) O2SR, C2OR Vector representation [15] introduced by Zhu et al (2021) Wang et al (2021) proposed Refined GloVe and Refined Word2Vec [16]

Pre-trained model approaches

DistilBERT is a smaller and more efficient version of the well-known BERT (Bidirectional Encoder Representations from Transformers) concept. DistilBERT aims to maintain a same level of performance as BERT while considerably decreasing its size, resulting in improved speed, cost-effectiveness, and reduced resource requirements [18]. MobileBERT was developed via an innovative approach known as progressive knowledge transfer. The process involves distilling knowledge from the original BERT model to progressively smaller intermediate models, leading to the final MobileBERT model. Knowledge distillation is used to transfer information from larger models to smaller ones, helping MobileBERT retain important linguistic knowledge [19] A Lite BERT for Self-supervised Learning of Language Representations" introduces ALBERT, a more lightweight variant of the BERT (Bidirectional Encoder Representations from Transformers) model for self-supervised learning of language representations [20] Distilling BERT for Natural Language Understanding" introduces TinyBERT, a compressed and distilled version of the BERT (Bidirectional Encoder Representations from Transformers) model tailored for Natural Language Understanding (NLU) tasks [21] Pre-training Text Encoders as Discriminators Rather than Generators" introduces the ELECTRA model, a novel approach to pre-training text encoders for natural language processing (NLP) tasks. Unlike traditional pre-training methods that use a masked language model (MLM) objective to predict missing tokens [22]

3. Techniques for Domain adaptation and Ensemble Learning Framework

Domain Adaptation

Domain adaptation, which involves transferring knowledge from a source domain to a target domain, has shown promise in bridging this gap. By fine-tuning pretrained language models on target domain data, we can enhance their ability to capture domain-specific nuances and improve sentiment prediction accuracy. Nevertheless, relying on a single adapted model may still be insufficient to achieve robust cross-domain performance. Ensemble learning, which combines the predictions of multiple models, offers a powerful solution to enhance the robustness and accuracy of sentiment analysis systems. By integrating both homogeneous bootstrapping (using the same model architecture with different initializations) and heterogeneous bootstrapping (using diverse model architectures), we can leverage the strengths of various models and mitigate individual model biases.

Our Methodology

Our experimental setup involves extensive evaluations across multiple benchmark datasets representing different domains. The results demonstrate significant improvements in sentiment prediction accuracy and generalization performance compared to baseline models without domain adaptation and ensembling. Through detailed analysis, we identify the critical factors that contribute to these enhancements and discuss the practical implications of our approach. This research not only advances the state-of-the-art in sentiment analysis but also provides actionable insights for deploying robust, cross-domain sentiment analysis systems in real-world scenarios.

Homogeneous Bootstrap Sampling:

Bootstrap Sampling: Bagging involves creating multiple subsets (or samples) of the training dataset through bootstrap sampling

Let's denote our training dataset as = {D= $\{(x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n)\}$, where xi represents the input features and y_i represents the corresponding labels. For each model k in the ensemble, we generate a bootstrap sample D_k by randomly sampling with replacement from the original dataset D. This results in a new dataset D_k of the same size as D, but some examples may be duplicated and others omitted. Let f (x; θ) represent the base model (e.g., a neural network) with parameters θ. For each bootstrap sample D_k, we train an individual model)f_k $(x;\theta_k)$ using the base model f:

We optimize the parameters θ_k of f_k by minimizing a loss function) $L(\theta_k)$ on the bootstrap sample Dk. This can be represented as: $\theta_k = \text{argmin}_\theta L\ (\theta; Dk)$ The ensemble prediction $y_{pred}(x)$ for a new input x is computed based on the aggregation of predictions from all individual models f_k

$y_{pred}(x)=\text{Aggregation}(f_1(x),f_2(x)\ldots..f_k(x))$

Pseudocode:

```
models = []


for k = 1 to K:
    D_k = random_sampling_with_replacement(D, n)
    theta_k = train_model(f, D_k)
    models.append((f, theta_k))
```

```
function predict(x):
    predictions = []
    for (f, theta_k) in models:
        prediction = f(x, theta_k)
        predictions.append(prediction)
    final_prediction = aggregate_predictions(predictions)
    return final_prediction

input_x = new_input_data()
output_y = predict(input_x)
```

Heterogenous Bootstrapping

Pseudo code:

```
models = {}
D = [(x1, y1), (x2, y2), ..., (xn, yn)]

for k in range(1, K + 1):
    Dk = bootstrap_sample(D)
    fk, θk = train_model(Dk)
    models[k] = (fk, θk)

def predict_ensemble(x):
    predictions = []
    for k in range(1, K + 1):
        fk, _ = models[k]
        prediction_k = fk.predict(x)
        predictions.append(prediction_k)
```

```
    y_pred = aggregate_predictions(predictions)
    return y_pred

def aggregate_predictions(predictions):
    votes = [1 if pred == 1 else -1 for pred in predictions]
    if sum(votes) >= 0:
        return 1
    else:
        return -1

import random
def bootstrap_sample(data):
    n = len(data)
    return [data[random.randint(0, n - 1)] for _ in range(n)]

# Example Usage:
x_new = ...
predicted_label = predict_ensemble(x_new)
print("Predicted Label:", predicted_label)
```

4. Experimental Setup

Dataset

The dataset comprises user-generated reviews from YouTube, with binary sentiment labels (Positive/Negative) applied to each review. It has been balanced to ensure an equitable distribution of sentiment categories for sentiment analysis tasks in the areas of Finance and IMDB reviews. The dataset is evenly distributed and encompasses sentiments from various user demographics and communication patterns inherent to these sites.

Table 1: Summary of case study datasets

Dataset	Domain	Problem Type
Finance	Finance Reviews	Ternary
IMDB	Movie Reviews	Binary
Custom	All Reviews	Binary

Preprocessing

The dataset was subjected to a preprocessing procedure in order to confirm its appropriateness for training and assessing models for sentiment analysis. During this preprocessing stage, a series of essential procedures were implemented to sanitise and organise the data. This involved text normalisation to address variances in letter case and punctuation, the eradication of special characters and numerical values, and the exclusion of stop words to minimise textual interference. In addition, the text was subjected to tokenization, a process used to divide it into individual words or tokens, which allowed for further analysis. In addition, any duplicate or extraneous entries were eliminated to ensure the accuracy and consistency of the data. The dataset was cleaned and processed to remove any noisy or redundant information. This refined dataset was then used as the input for training and assessing the sentiment analysis models. By doing so, the models were able to concentrate on the significant content of the reviews while reducing the influence of extraneous aspects.

Table 2: Metadata on the selected datasets

Dataset	#Document	Class Distributions			Source	Supplementary details
		Positive	Negative	Neutral		
Finance	4000	1260	582	2157	Kaggle	Downsized
IMDB	4,000	19981	20019		Kaggle	Downsized
Custom	1000	500	500		YouTube	Scrapped through API Key

Environment and execution

The studies were performed on virtual machines (VMs) that were hosted on the Azure and Google Colab platforms. The Azure platform offers a virtual machine that has CPU resources specifically designed for executing tasks that require minimal resources. The presence of 28 GB of RAM enabled the efficient management of larger datasets and models. This platform offers cloud-based computing resources, eliminating the requirement for on-site hardware. The programming language utilised for the experiments was Python. All the experiment implementations, including data preprocessing, model fine-tuning, prediction, and performance evaluation, may be found in a dedicated GitHub repository. This repository functions as a thorough resource for accessing the codes and replicating the tests performed on both Azure VM and Google Colab platforms.

Table 3: Finetuning Parameters

Hyperparameter	DistilBERT	MobileBERT	ELECTRA	ALBERT
Training epochs	2	3	3	3
Batch size per device during training	8	8	8	16
Batch size for evaluation	16	8	8	16
Warmup steps for learning rate scheduler	500			

strength of weight decay	0.01			
Logging steps		1000	1000	500
Evaluation Strategy		steps	steps	

5. Results and discussions

ML models, particularly Gradient Boosting Classifier and Random Forest, demonstrate strong performance in both accuracy and log loss metrics, with Gradient Boosting Classifier achieving the lowest log loss. While DL models initially showed competitive performance, the feed forward neural network experienced a slight decrease in performance after early stopping, indicating potential overfitting issues. ML models, with their simpler architectures and ensemble techniques, appear to be more robust and effective for this breast cancer classification task based on the provided metrics. However, further analysis, such as considering additional evaluation metrics or exploring different DL architectures, may provide a more comprehensive understanding of the comparative performance of ML and DL models

Table 4: Homogeneous Ensembling

Base Model	Dataset	Performance Metrics			
		Accuracy	Precision	Recall	F1 score
DistilBERT	Social media	83	94	73	82
	Finance Review	43	100	18	30
MobileBERT	Social media	52	52	99	69
	Finance Review	69	69	100	82
ELECTRA	Social media	85	89	82	85
	Finance Review	48	99	25	40
ALBERT	Social media	74	89	57	70
	Finance Review	69	69	97	82

Table 5: Performance of Machine Learning Model

Model	Parameter	Best Value	Best Score (neg_log_loss)
Random Forest	max_features	5	-0.0201
GradientBoosting Classifier	max_depth	4	-0.000459
	n_estimators	125	
	learning_rate	0.1	

Stacking	KNN n_neighbors	2	-0.0144
	TREE_max_depth	3	
	SVM C	1	
	final_estimator__max_features	2	

Fig1 : Loss Vs Epoch

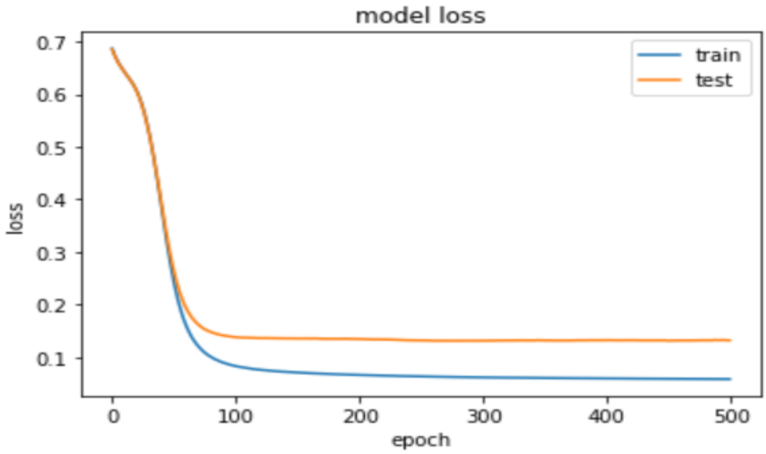
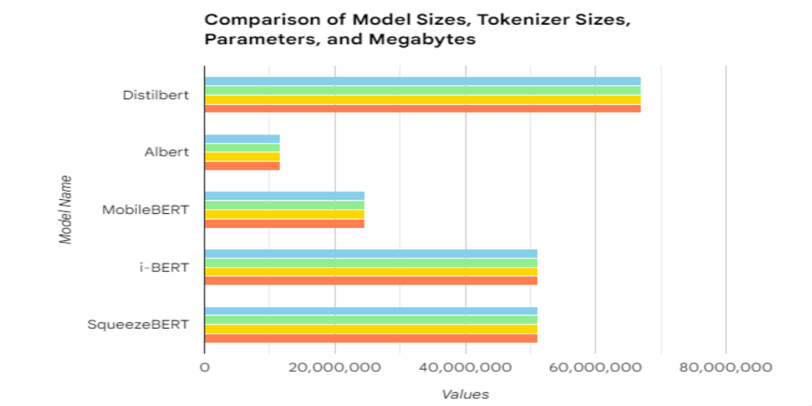


Fig2: Comparison Chart



6. Enhancement and Future Directions:

To enhance sentiment analysis through domain adaptation and ensemble learning, future research can focus on several key areas. First, incorporating multimodal data, such as text, images, and videos, alongside textual data, could provide richer sentiment insights from social media platforms. Advanced domain adaptation techniques like adversarial training and few-shot learning could further improve model robustness and adaptability to new domains with minimal labeled data. Implementing dynamic and continual learning approaches, such as online learning and lifelong learning, would allow models to continually adapt to new data and tasks. Enhancing model interpretability and transparency through explainable AI methods would build trust and facilitate the understanding of how sentiment predictions are made.

Future directions also include extending the approach to cross-lingual sentiment analysis, developing personalized sentiment analysis models, and exploring temporal aspects of sentiment analysis. Integrating knowledge graphs and building real-time sentiment monitoring systems optimized for latency and computational efficiency are additional avenues for advancing sentiment analysis. Ethical considerations, bias mitigation, and domain-specific customization for critical sectors like healthcare and finance will be crucial for ensuring fair and accurate sentiment analysis outcomes.

References

1. Kazmaier J, Van Vuuren JH. The power of ensemble learning in sentiment analysis. Expert Systems with Applications. 2022 Jan 1; 187:115819.
2. Alsayat A. Improving sentiment analysis for social media applications using an ensemble deep learning language model. Arabian Journal for Science and Engineering. 2022 Feb;47(2):2499-511
3. Mohammed A, Kora R. An effective ensemble deep learning framework for text classification. Journal of King Saud University-Computer and Information Sciences. 2022 Nov 1;34(10):8825-37.
4. Phan HT, Tran VC, Nguyen NT, Hwang D. Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model. Ieee Access. 2020 Jan 3; 8:14630-41.
5. Jianqiang, Zhao, Gui Xiaolin, and Zhang Xuejun. "Deep convolution neural networks for twitter sentiment analysis." IEEE Access 6 (2018): 23253-23260.Rehman, A.U., Malik, A.K., Raza, B. and Ali, W., 2019. A Hybrid CNN-LSTM Model for Improving Accuracy of Movie Reviews Sentiment Analysis. Multimedia Tools and Applications, pp.1-17.
6. Ye, Zhe, Fang Li, and Timothy Baldwin. "Encoding Sentiment Information into Word Vectors for Sentiment Analysis." In Proceedings of the 27th International Conference on Computational Linguistics, pp. 997-1007.2018.
7. Fouad MM, Gharib TF, Mashat AS. Efficient twitter sentiment analysis system with feature selection and classifier ensemble. In The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018) 2018 (pp. 516-527). Springer International Publishing
8. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013 Jan 16.
9. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) 2014 Oct (pp. 1532-1543).
10. Parveen, N., Chakrabarti, P., Hung, B.T. et al. Twitter sentiment analysis using hybrid gated attention recurrent network. J Big Data 10, 50 (2023). https://doi.org/10.1186/s40537-023-00726-3
11. Kaur, G., Sharma, A. A deep learning-based model using hybrid feature extraction approach for consumer sentiment analysis. J Big Data 10, 5 (2023).

https://doi.org/10.1186/s40537-022-00680-6
12. Lin, CH., Nuha, U. Sentiment analysis of Indonesian datasets based on a hybrid deep-learning strategy. J Big Data 10, 88 (2023). https://doi.org/10.1186/s40537-023-00782-9
13. Wang L, Niu J, Yu S. SentiDiff: combining textual information and sentiment diffusion patterns for twitter sentiment analysis. IEEE Trans Knowl Data Eng. 2020;32(10):2026–39. https:// doi. org/ 10. 1109/ tkde. 2019. 29136 41.
14. Hao Y, Mu T, Hong R, Wang M, Liu X, Goulermas JY. Cross-domain sentiment encoding through stochastic word embedding. IEEE Trans Knowl Data Eng. 2020;32(10):1909–22. https:// doi. org/ 10. 1109/ tkde. 2019. 29133 79.
15. Zhu L, Li W, Shi Y, Guo K. SentiVec: learning sentiment-context vector via kernel optimization function for sentiment analysis. IEEE Trans Neural Netw Learn Syst. 2021;32(6):2561–72. https:// doi. org/ 10. 1109/ tnnls. 2020. 30065 31.
16. Dhakal S. Combining sentiment lexicons and content-based features for depression detection. IEEE Intell Syst. 2021; 36:99–105. https:// doi. org/ 10. 1109/ MIS. 2021. 30936 60.
17. Eklund M. Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data.
18. Sanh V, Debut L, Chaumond J, Wolf T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter. arXiv preprint arXiv:1910.01108. 2019 Oct 2.
19. Sun Z, Yu H, Song X, Liu R, Yang Y, Zhou D. Mobilebert: Task-agnostic compression of bert by progressive knowledge transfer.
20. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942. 2019 Sep 26.
21. Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, Wang F, Liu Q. Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351. 2019 Sep 23.
22. Clark K, Luong MT, Le QV, Manning CD. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555. 2020 Mar 23.
23. Kim S, Gholami A, Yao Z, Mahoney MW, Keutzer K. I-bert: Integer-only bert quantization. International conference on machine learning 2021 Jul 1 (pp. 5506-5518). PMLR.