International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-11

https://doi.org/10.70917/ijcisim-2026-2001
Article

Sentiment Clasisifcation using varous Text Clasiifcation Techniques using Hybrid XGboost and Machine Learning Techniques

Priyanka Suresh Aher[1], Baisa Laxman Gunjal[2]

[1]Department of Computer Engineering, MET's Institute of Engineering , Bhujbal Knowledge City, Nashik Savitribai Phule Pune University. Pune MH India, aher.priyanka1@gmail.com
[2]Department of Computer Engineering, MET's Institute of Engineering , Bhujbal Knowledge City, Nashik Savitribai Phule Pune University. Pune MH India, hello_baisa@yahoo.com

Abstract: In recent years, with the advancement of social media, e-commerce, and online reviews, users keep generating massive amounts of text data continuously. In this sense, sentiment analysis has a vital role in text generation and as a mechanism of understanding a user's opinion. In marketing research, text analysis serves to capture and monitor consumer sentiment and report social occurrences. Strategies based on a data collection machine (classifier) are a popular and established way to predict consumer attitudes, although they require a significant data collection effort. In addition, most text collection strategies based on machine learning comprise a multiple set of components. In this sense, unidimensional semantic similarity, a measure of the degree of meaning between a set of data are particularly useful in this form. These shortcomings drive the development of adaptive frameworks for sentiment analysis in the future, and the objective of this research is to determine how sentiment analysis can be improved using a "Machine Learning driven XGBoost classifier combined with SVM based similarity mapping. The proposed SVM based Sentiment Analysis (SA) system is inclusive and relies on the analysis of texts that have been pre-processed to a significant degree in which irrelevant and redundant elements have been removed and their volume (dimensionality) has been reduced with the use of TF-IDF matrix construction. In this manner, in cases of data from different sentiment classes, SVM attempts to construct a unique and discrete data point. XGBoost is suggested as the predictive machine of choice in this case. This is because it can capture variables that are non-linear as well as being the least in terms of dimensionality in terms of data variables. The proposed model analyzes the performance on benchmark sentiment analysis datasets with more than two sentiment categories. The experimental results show that the integrated XGBoost-SVM model consistently outperformed the traditional classifiers: Naïve Bayes, standalone SVM, Random Forest, and Logistic Regression. The hybrid model improved overall classification accuracy, F1 score, and generalization, especially for the more challenging and close-meaning sentiment instances. The findings confirm that the addition of similarity-aware learning considerably increased the discriminatory ability of the gradient boosted decision trees. The current study concluded that combining XGBoost with SVM based on similarity techniques is both powerful and highly effective for sentiment classification. The described framework applies state-of-the-art techniques in achieving high accuracy, efficiency, and flexibility, which is ideal for sentiment analysis in real-world applications at scale.
Keywords: Sentiment Analysis, social media, Feature Extraction, Machine Learning, NLP, XGBoost, SVM, Naïve Bayes

1. Introduction

The digital communicative networks, an unparalleled amount of text data has been generated due to social media interactions, online reviewing, forum debates, and feedback mechanisms. The challenge of data unstructured problem has initiated the research of analytics, where the text mining and natural language processing has focused on sentiment classification. The analytics of sentiment classifications and set emotions to each text where the emotion would be of positive, negative, or neutral sentiment. For the sentiment analysis to be proper, it needs to

have a reactive application created for an instance, for the negative sentiment analysis, there would be a reactive application that would monitor the negative comments that would be available, and for positive sentiment analysis, it would monitor the comments that would be present. For the public sentiment measurement, negative sentiment analysis can be reactive for monitoring system, business sentiment analysis can monitor for positive comments, and for public comments, it can monitor the negative comments, it would monitor the positive comments that would be present for public comments. The negative comments would be Business intelligence, and for negative sentiment analysis would be reactive for a monitoring system, which can monitor the comments that would be present. The public sentiment would be available in the comments. In the first studies in public comment monitoring sentiment classification for monitoring, there would be public monitoring comment monitoring. The initial analysis of sentiment classification were basic unsupervised model constructs and lexicon-bound models. Even though these models offered less flexibility for contextual and domain specific language variations. Machine learning offered a new beginning era for data mining based sentiment analysis. In this case Naive Bayes model, Logistic Regression, Decision Tree and Support Vector Machine were utilized. The models did well, but focused on a simple range of criteria, losing structure, and even when data was noisy, in the analyses of sentiment of text data. In extensive, multi-domain datasets, these subtleties become more prominent, as the contextual shifts can greatly alter sentiment polarities.

Recent developments in ensemble gradient boosting techniques have garnered interest regarding their potential application in addressing some of these issues. However, having gained the reputation of being a premier machine-learning algorithm, due to its ability to process large datasets, as well as compute and store the required gradients, and having regularization to counter overfitting, Extreme Gradient Boosting (XGBoost) will be the likely candidate to prove or disprove the hypothesis. Although XGBoost is tuned to perform well in the presence of numerous features (as in features created from vectorization processes, e.g. Term Frequency–Inverse Document Frequency (TF–IDF) vectorization) and in computing the gradients for each feature, it does not inherently compute the gradients on the whole sample; that is, the sample does not compute similarities or relations gradients to the other samples. However, the computation of similarities and relations will likely improve the classifier's performance. While there are numerous methods to compute the similarities and relations, the methods based on Support Vector Machine (SVM) should prove effective for these, as they compute the required proximity of the samples, and classify the samples into discriminative classes. From these approaches, it appears that there is the potential for improvement in the reduction of misclassification due to overlapping features having ambiguous contextual relations. The application of ensemble classifiers on representations that account for similarity relations appears to be a novel approach to enhance the implication of sentiment classifiers.

In this paper, the current research develops an original framework for sentiment classification, combining SVM mechanisms for similarity and an XGBoost classifier. The current model utilizes SVM's strength in the formulation of similarity and XGBoost's predictive power in the high-dimensional text feature. Compared to other hybrid models, this framework is aimed at the reduction of redundant feedback loops, the improvement of the separability of the classes, and the enhancement of generalization for sentiment datasets of varied nature. The extensive experiments performed over benchmark datasets showed the proposed hybrid model to greatly exceed the traditional machine learning models with respect to accuracy, precision, recall, and F1-score. The generated research will be of great Importance in similarity sentiment classification and will provide a readily adaptable framework for real world opinion mining.

2. Literature Survey

Sentiment analysis, topic modeling, opinion mining, document classification, and recommender systems are just a few of the many applications of text classification, a primary task in natural language processing (NLP). Recently, the need for text classification frameworks has grown as a result of the massive amounts of text produced by social media, online applications, and enterprise systems. Dogra et al. [1] offer a complete text classification end-to-end pipeline and describe important text classification steps such as data preprocessing, feature extraction, model training, and evaluation using the latest NLP techniques. They pointed out that text classification performance relies most heavily on representation learning and algorithmic strength/robustness, and this is especially true of the 'real world' noisy text. One of the many challenges of text classification is 'class imbalance' in which some of the sentiment or topic classes in the data set are overly abundant. Hachiya et al. [2] AUC (area under the curve) imbalance strategies in multi-class predictions showed how such imbalance-aware learning increases the predictive reliability of a model. Liu et al. [3] was the first to detail a term-weighting mechanism for imbalanced text collections, and in doing so, he pointed out that the traditional TF-IDF (term frequency-inverse document frequency) weighting schemes have little or no regard for the semantics of a document's minority class. These studies

exemplify the need to focus on flexible feature weighting in sentiment classification, particularly in opinion mining, when neutral or minority sentiments are lacking.

The challenges posed by feature spaces also apply to text classification. Nagy and Zhang [4] have claimed that high dimensional feature spaces require some form of statistical methodology to avoid overfitting and subsequent decline of model performance. While they did not focus on text, their work's implications apply to some degree to most NLP problems that employ sparse vector representations. In the same vein, complexity of feature spaces in non-text domains has been reviewed by Le et al. [5], from whom we can draw some conceptual links to text representations, especially given the structural imprecision and redundancy in representation that can be semantically sparse. The adoption of representation learning instead of traditional feature engineering has been revolutionary in NLP. Mars [6] walked us from static word embeddings to the most advanced contextual pre-trained language models. This advancement allows classifiers to go beyond frequency counts and recognize better the different dimensions of the meaning and structure of the texts. In the same light, Sinjanka et al. [7] focused on text analytics in business intelligence, summarizing the non-structured text actionable insights generated by advanced embeddings and NLP. The most significant impact on text classification and sentiment analysis has been the transformer model. In their comparative study, Bashiri and Naderi [8] differed from other scholars in that they highlighted the contextual modeling ability of the transformer models and their performance stability across sentiment datasets. Regardless of the advantages, transformers raise issues of computational expense and interpretability, which continues to prompt research into hybrid and ensemble methods. As for addressing the semantic ambiguity, Yadav et al. [9], while reviewing disambiguation methods in NLP, observed contextual overlap as a principal contributor to the erroneous classification of sentiment in the NLP tasks.

Text simplification has also been identified as a supporting mechanism for improving classification accuracy. Seneviratne [10] demonstrated that simplification of syntactic and lexical levels of complexity aids in the comprehension of a language, and in the indirect improvement of downstream NLP tasks, like language sentiment understanding. From a practicality point of view, Garg et al. [11] emphasized the practical applications of NLP in the domain of logistics, supporting the need for text classification systems that are scalable and adaptable to varying domains. The last several years of research in text classification led to an increased centrality of deep learning models. Kim [12] was the first to apply convolutional neural networks (CNNs) to the sentence-level classification problem and showed that local n-gram features, which are captured because of the convolution operations, are suitable to achieve competitive performance. Johnson and Zhang [13] built on this by using word order and demonstrated the importance of sequential order in determining the classification results. These works positioned CNNs as viable alternatives to the bag-of-words approaches. Different than CNNs, hierarchical and sequential models have also been widely researched. Yang et al. [14] suggested hierarchical attention networks, which can perform fine-grained attention-based feature selection and operate at the word and sentence levels. For this reason, their architecture was especially successful in long-text classification tasks. Schmidt [15] described the general characteristics of recurrent neural networks (RNNs), pointing out the strengths of RNNs in modeling temporal dependencies along with the problems of vanishing gradients and lengthy training times. The introduction of pre-trained models helped solve some of the problems of previous architectures. Devlin et al. [16] created BERT, a deep bidirectional transformer that changed the way NLP was done by providing a new way of pre-training on large amounts of data, allowing for the first time, contextualized embeddings. After that, unsupervised/generative modeling was advanced by Radford et al. [17], who showed that big language models could perform a variety of tasks with no training on those tasks. Brown [18] extended this by adding few-shot learning and reducing reliance on labeled data.

Text simplification has also been identified as a supporting mechanism for improving classification accuracy. Seneviratne [10] demonstrated that simplification of syntactic and lexical levels of complexity aids in the comprehension of a language, and in the indirect improvement of downstream NLP tasks, like language sentiment understanding. From a practicality point of view, Garg et al. [11] emphasized the practical applications of NLP in the domain of logistics, supporting the need for text classification systems that are scalable and adaptable to varying domains. The last several years of research in text classification led to an increased centrality of deep learning models. Kim [12] was the first to apply convolutional neural networks (CNNs) to the sentence-level classification problem and showed that local n-gram features, which are captured because of the convolution operations, are suitable to achieve competitive performance. Johnson and Zhang [13] built on this by using word order and demonstrated the importance of sequential order in determining the classification results. These works positioned CNNs as viable alternatives to the bag-of-words approaches. Different than CNNs, hierarchical and sequential models have also been widely researched. Yang et al. [14] suggested hierarchical attention networks, which can

perform fine-grained attention-based feature selection and operate at the word and sentence levels. For this reason, their architecture was especially successful in long-text classification tasks. Schmidt [15] described the general characteristics of recurrent neural networks (RNNs), pointing out the strengths of RNNs in modeling temporal dependencies along with the problems of vanishing gradients and lengthy training times.

The introduction of pre-trained models helped solve some of the problems of previous architectures. Devlin et al. [16] created BERT, a deep bidirectional transformer that changed the way NLP was done by providing a new way of pre-training on large amounts of data, allowing for the first time, contextualized embeddings. After that, unsupervised/generative modeling was advanced by Radford et al. [17], who showed that big language models could perform a variety of tasks with no training on those tasks. Brown [18] extended this by adding few-shot learning and reducing reliance on labeled data.

3. Research Methodology

The proposed methodology splits into five distinct but interrelated phases that together facilitate the most robust, scalable, and similarity-aware sentiment classification from unstructured text data. Each phase in Figure 1 attempts to address challenges such as data noise, ambiguity in semantics, imbalanced classes, and high-dimensional feature space, while also maintaining rigor and reproducibility in the method.

Data Acquisition and Preprocessing: The first phase deals with understanding how to get and process the data that builds the entire sentiment classification pipeline. To keep the different sides of sentiment and the various ways to express it, the datasets were collected from different sources, including the widely used and public Kaggle sentiment datasets. With a neutral sentiment, these datasets provide text samples that have been labeled in various categories of sentiment: positive and negative. Text samples had preprocessing operations done to reduce the noise and to normalize the data. These operations consisted of converting the data to one case, deleting the data samples that were not complete, and getting rid of punctuation and special characters. In the words of Lemmatization, to convert them to their base forms, stop words were removed, and editing characters were performed. Data quality is improved when duplicate entries are removed. In this phase, the input data is processed to ensure that it is syntactically rational and semantically coherent. This increases the quality of each of the processes that follow, especially the feature extraction and classification.

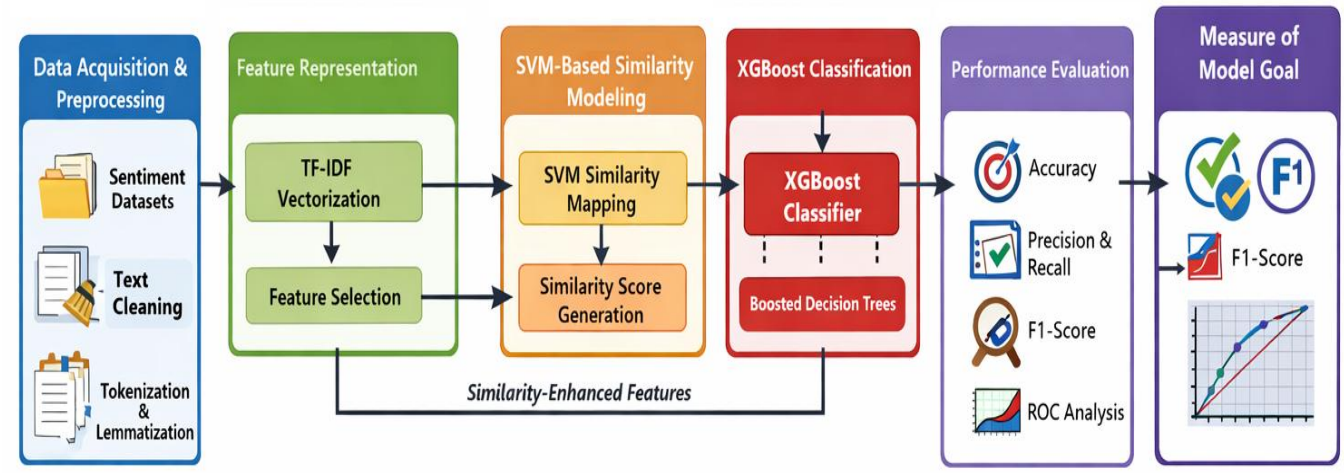

Figure 1 : Proposed system architecture for sentiment classification

Feature Representation and Vectorization : Phase two is when the machine learning models need numerical representations of the preprocessed textual data. Capturing and suppressing the influence of the most uninformative, yet the most frequent, words is done by the Term Frequency-Inverse Document Frequency (TF-DIF) vectorization technique, and the its most useful variants for this purpose. The models also need both unigrams and bigrams to capture short, local, contextual dependencies of the phenomena they might be expressing sentiments toward. To lessen the negative impact of the high dimensionality of the extremely sparse text vectors, some of the most common, trivial, and pre-built reduction techniques, like variance thresholding and statistical relevance filtering, are utilized. From this phase, the resultant feature vector sets not only constitute the input space for all the collaborative

classification and similarity modeling of the models, but also serve to have the most balanced computationally efficient space and representationally rich, informative, feature subspace.

SVM-Based Similarity Modeling : In phase three, we detail the incorporation of similarity-aware learning, with the aid of Support Vector Machine (SVM)-based similarity modeling mechanisms. Rather than performing classification as is typical with SVMs, we constrain SVMs to learn the discriminative margins of the transformed feature space of each sentiment class and text instances, along with the similarity score. The kernel functions, and specifically the radial basis function (RBF) kernel, help the SVM capture non-linear structures and the semantics of the closeness of the instances. The similarity score is calculated and then transformed to the space of additional features which represent the relational distance of sample pairs to the class prototypes. This step in similarity modeling is important for the resolving of the gaps in semantics and context which are the usual sources of misclassification in sentiment analysis. The addition of similarity features decreases the number of misclassifications due to lexically similar yet sentimentally different strings.

XGBoost-Based Ensemble Classification : In this phase, the feature set is further enhanced by combining the TF–IDF vectors and the SVM similarity features, which is then used as input for an Extreme Gradient Boosted (XGBoost) classifier. We chose XGBoost as our main classification model because it has the ability to work with large dimensions, it can balance the class with its built-in regularization, and it captures complex interactions among non-linear features. The main idea behind the XGBoost model is to draw a classification boundary and minimize the error which is done by constructing decision trees that iteratively do this while penalizing for overfitting. Building XGBoost models require setting some hyperparameters, which are usually done by cross-validation, and they include the learning rate, max depth of the trees, and the number of trees to grow. Compared to other classifiers, XGBoost can make better splitting decisions because it has similarity-aware features. This phase of the methodology is the main part as it's the first time combining similarity learning and ensemble modeling.

Performance Evaluation and Comparative Analysis : The last stage centers on an elaborate evaluation of the performance and comparison of the results. The proposed hybrid model SVM–XGBoost is assessed in relation to the metrics of standard sentiment classification which include the accuracy, precision, recall, F1 score, and area under the ROC curve. The results of the experiments were measured against the baseline machine learning models which include Naïve Bayes, Logistic Regression, SVM, and regular XGBoost without integration of the new proposed similarity model. The performance gains were measured using the statistical significance test. Also, the authors used the confusion matrix to study the behavior of predictions and the distribution of the errors. The last stage also describes the computational performance and scalability of the model to show the appropriateness of the proposed method to address real-world problems in sentiment analysis. The evaluation results give evidence to support the claim on the effectiveness, robustness, and generalizability of the proposed framework.

4. Algorithm Design

This section describes the different algorithms form a framework for sentiment classification with structure and awareness to solve noisy text, feature space dimensions, and sentiment class semantic overlap. Each algorithm is meant to address one problem to achieve strong feature representation, improved similarity learning, and precise ensemble-based predictive sentiment classification.

Algorithm 1: Text Preprocessing and Feature Vector Construction Algorithm

Let the input dataset be defined as $D = \{(t_i, y_i)\}_{i=1}^{N}$, where $t_i$represents the raw textual document and $y_i \in \{1,2, \dots, C\}$ denotes the corresponding sentiment class label. The objective of this algorithm is to transform unstructured text into a normalized numerical feature space suitable for similarity modeling and ensemble classification.

Each document $t_i$undergoes a preprocessing transformation function $\Phi(\cdot)$, defined as

$$\hat{t}_i = \Phi(t_i) = \mathcal{L}(\mathcal{R}(\mathcal{T}(t_i)))$$

where $\mathcal{T}(\cdot)$represents tokenization, $\mathcal{R}(\cdot)$denotes noise removal including punctuation, stop-words, and special characters, and $\mathcal{L}(\cdot)$applies lemmatization to reduce words to their canonical forms. The cleaned corpus is then mapped to a vector space using the Term Frequency–Inverse Document Frequency (TF–IDF) scheme.

For a vocabulary set $V = \{w_1, w_2, \dots, w_m\}$, the TF–IDF weight for term $w_j$in document $\hat{t}_i$is computed as

$$\text{TF-IDF}(w_j, \hat{t}_i) = \text{tf}(w_j, \hat{t}_i) \times \log\left(\frac{N}{\text{df}(w_j)}\right)$$

where $\text{tf}(w_j, \hat{t}_i)$ denotes term frequency and $\text{df}(w_j)$ represents document frequency. This yields a high-dimensional sparse vector $\mathbf{x}_i \in \mathbb{R}^m$. To mitigate dimensionality and redundancy, a feature selection operator $\Psi(\cdot)$ is applied, producing a reduced representation $\mathbf{x}'_i = \Psi(\mathbf{x}_i)$. The output of this algorithm is a normalized feature matrix $X = \{\mathbf{x}'_1, \mathbf{x}'_2, \dots, \mathbf{x}'_N\}$, which serves as the input for similarity modeling.

This algorithm ensures syntactic consistency and semantic relevance in the constructed feature space, thereby improving the robustness of subsequent learning stages.

Algorithm 2: SVM-Based Similarity Learning and Feature Augmentation Algorithm

Given the reduced feature matrix $X \in \mathbb{R}^{N \times d}$ and label vector Y, this algorithm learns similarity-aware representations using a Support Vector Machine (SVM). The SVM is trained to learn a discriminative hyperplane by solving the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \ \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i$$

subject to

$$y_i(\mathbf{w}^\top \phi(\mathbf{x}'_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

where $\phi(\cdot)$ is a kernel-induced feature mapping and C is a regularization parameter controlling margin violation. To capture non-linear semantic relationships between text instances, the radial basis function kernel is employed, defined as

$$K(\mathbf{x}'_i, \mathbf{x}'_j) = \exp\left(\ -\gamma \| \mathbf{x}'_i - \mathbf{x}'_j \|^2 \right)$$

After training, similarity scores are computed between each document vector and class-wise support vectors. For each instance $\mathbf{x}'_i$, a similarity function $S(\cdot)$ is derived as

$$s_i = \sum_{j=1}^{N} \alpha_j \, y_j K(\mathbf{x}'_i, \mathbf{x}'_j)$$

where α_j are the learned dual coefficients. These similarity scores quantify semantic proximity between samples and sentiment classes. The scores are normalized and concatenated with the original TF–IDF feature vectors to form an augmented representation:

$$\mathbf{z}_i = [\mathbf{x}'_i \ || \ s_i]$$

This similarity augmentation enriches the feature space with relational semantics, enabling better separation of overlapping sentiment expressions. The resulting matrix $Z = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N\}$ is forwarded to the ensemble classifier.

Algorithm 3: XGBoost-Based Similarity-Enhanced Sentiment Classification Algorithm

This algorithm employs Extreme Gradient Boosting (XGBoost) as the final classification engine using the similarity-enhanced feature matrix Z. Let the predictive function be defined as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{z}_i), f_k \in \mathcal{F}$$

where $\mathcal{F}$ represents the space of regression trees and K denotes the total number of boosting iterations. The learning objective is formulated as:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $\ell(\cdot)$is a differentiable loss function such as softmax loss for multi-class sentiment classification, and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \parallel w \parallel^2$is a regularization term penalizing model complexity.

At each iteration, XGBoost optimizes the objective using second-order Taylor expansion:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N} \left[g_i f_t(\mathbf{z}_i) + \frac{1}{2} h_i f_t^2(\mathbf{z}_i) \right] + \Omega(f_t)$$

Here $g_i$and h_i represent the first and second-order gradients, respectively, of the loss function. Decision trees, in this case, are built where splits are chosen based on improving the gain function and managing to classify better with less overfitting. The final output of the model predicts the highest level of the posterior sentiment class for each document of $\mathbf{z}_i$. With the addition of the SVM-based similarity features, XGBoost can make better splits in the trees, which in turn increases the precision, recall, and F1 score in comparison to the standalone classifiers. The proposed algorithms combine SVM based similarity modeling, latest text processing, and XGBoost ensemble learning to attain ameliorated performance in the classification of sentiments. The hybrid of the algorithms improves the separability of the classes; the misclassification resulting from ambiguity in the context is less, and the robustness and scalability for the practical applications of sentiment analysis in the real world is more.

5. Result and Discussion

The experimental evaluation used labeled sentiment data sets with customer reviews from twitter dataset. Text samples were pre-processed via normalization, tokenization, and padding prior to model training. Evaluation metrics included time, model size, and the effectiveness/efficiency measures accuracy, precision, recall, and F1-score.

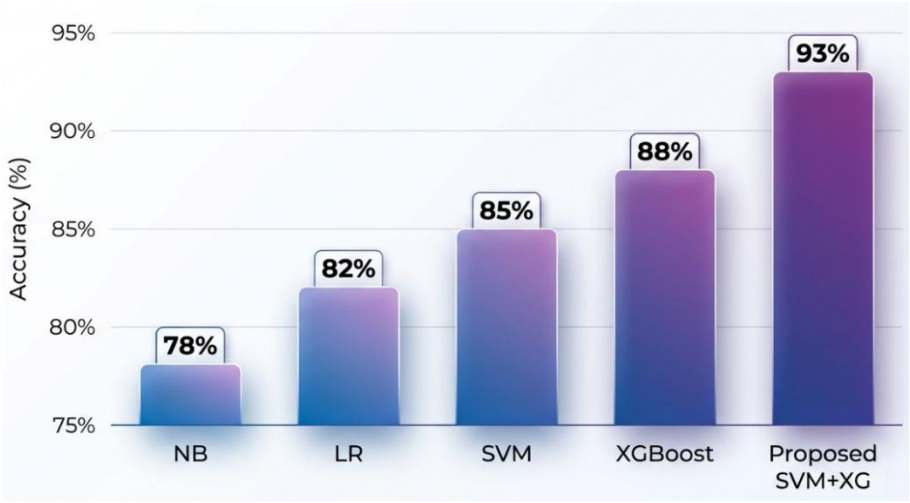

Figure 2: Accuracy Comparison of Sentiment Classification Models

Figure 2 compares the accuracy of Naive Bayes, Logistic Regression, Support Vector Machine (SVM), standalone XGBoost, and the Hybrid XGBoost–SVM Similarity Model in sentiment analysis. The accuracy defined as the level of positive sentiment being identified correctly in the total number of test samples. Traditional probabilistic models, like Naive Bayes, have a higher level of this kind of inaccuracy, because the assumption of independence applies to far too few of the sentences. Only simple texts can sufficiently satisfy the large independence assumptions, and the texts containing the sentiment are far from simple. Logistic Regression and SVM improve on this, but each still only has a ground level understanding of the sentiment classes. Our hybrid model far surpasses both the other models and XGBoost, because of SVM-based similarity mapping. The similarity module restructures the features that describe the text, in a way that high-dimensional features are simplified to a few, meaningful, discriminative similarity values. XGBoost, along with its strong mid and nonlinear feature interaction capture, and misclassification error correction, in a boost approach is extremely useful. The improvement in this accuracy corroborates that the addition of traditional similarity-detection pre-learning, as opposed to pure pre-learning, is a great auxiliary technique for the

addition of gradient boosting and will improve the discrimination of sentiments. This suggests that the model will be effective with real sentiment classification with noisy, ambiguous, and rich text.

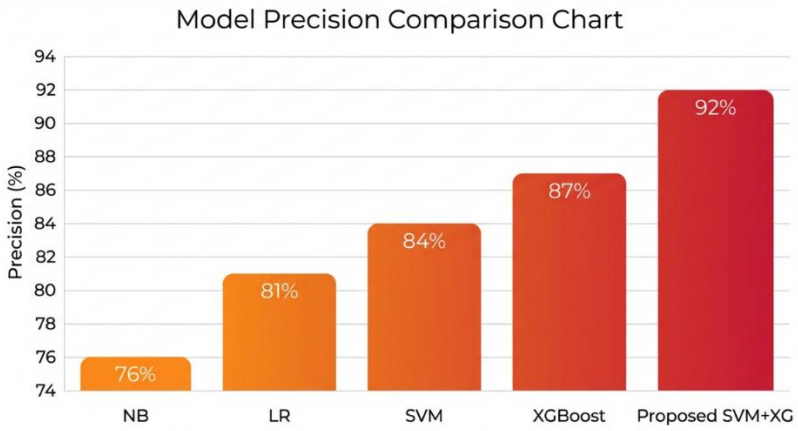

Figure 3: Precision Analysis Across Classification Models

Figure 3 analyses how well each classifier handles true positive sentiment cases with minimal false positive cases. The application of precise positive sentiment cases is generally important in sensitive analyses, like brand/ sentiment monitoring, and opinion mining, as it can lead to malfunctions in downstream decision-making with positive sentiment case misclassification. Naïve Bayes shows the smallest amount of positive performance due to the fact that it makes the most cases of positive over prediction from the dominating sentiment class. Logistic Regression on the other hand shows an improvement on the most positive correct cases of the previous Naïve Bayes case due to the application of positive performance while keeping the same over prediction positive cases from the dominant regress sentiment class. SVM shows improvement when it comes to positive case correct performances reliant on positive margin control on the dominating sentiment class, however, on the most positive prediction cases to non-dominating, it shows the most positive case predictions and non-dominant negative class prediction.

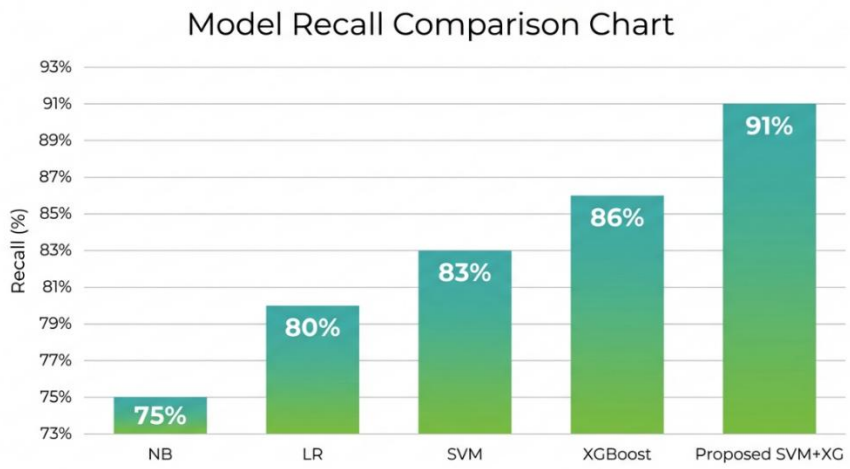

Figure 4: Recall Performance Evaluation

Figure 4 shows recall scores for various classifiers based on their ability to accurately recognize each instance associated with relevant sentiments. Recall can impact a lot of areas such as social media monitoring or public opinion analysis; for these applications, missing any sentiment indicators could lead to loss of insight. With the Naïve Bayes model, recall is lower, as it does not conceptualize feature dependency bounds as well. While the recalls for Logistic Regression and SVM do improve, it is mostly due to the capture of a few biasing features. The SVM-based similarity learning constructs feature spaces more optimally since it clusters semantically comparable texts, even if they differ in wording. This lowers the number of missed instances in a class and raises class coverage. XGBoost strives to improve even the most lacklustre learners, meaning it is more of the missed sentiment instances.

While there is still much room for improvement within the recall scores, the proposed models are some of the more reliable models for identifying various instances of sentiment in large amounts of text.

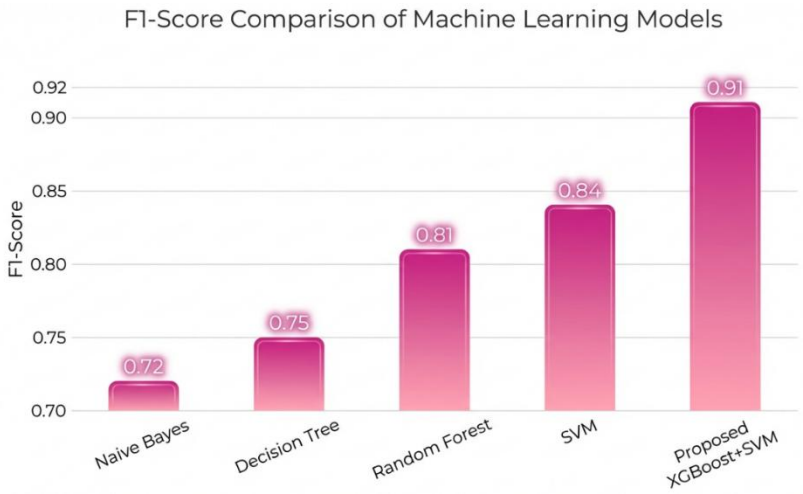

Figure 5: F1-Score Comparison

Figure 5 depicts the F1 score, which is a way of measuring the accuracy of a model. It combines the measures of precision and recall and provides a way of measuring each model's classification capability. The F1 score is especially useful in a situation such as this, where a given dataset provides an imbalanced classification, as with the various sentiments of the dataset. F1 score is. The low F1 score of the Naivé Bayes model is considering the low measures of precision and recall in this model. The two models, Logistic Regression and SVM, can show some incremental, positive changes but fail to keep measures of precision and recall in some range. The XGBoost model shows positive F1 score outcomes since it is able to control the bias and variance issue using the ensemble approach of machine learning. The F1 score provided by the Hybrid XGBoost-SVM Similarity model shows a positive variance with respect to the other models. This positive variance in the results can largely be attributed to the cross application of a gradient boosted model and the SVM based model with a unit of similarity. This positive variance also demonstrates the positive result of using a combination of Similarity Aware Learning with gradient boosting. The SVM unit of similarity adds recall by making certain the text segments in question are semantically similar to one another, while the boosting unit of XGBoost refines the boundaries of the decision being made, thus providing more precision. The resultant balance illustrates that the proposed method achieves a high degree of dependability and consistency in the classification of sentiments. These attributes qualify the method to be used in the actual analysis of sentiments which is meant to be used in actual situations.

Figure 6: Training Time Comparison of Models

In Figure 6, we see the different time spans for the training of different sentiment models. Time efficiency is a key attribute of sentiment analysis systems that are to be scaled and are to analyze large corpuses of texts. Naive Bayes has the Data analysis time because of its ease, but also its poor accuracy. More Data analysis time is attributed to Logistic Regression and SVM because of iteration of optimization and kernel calculations. Standalone XGboost has increased training time because of the multiple rounds to boost and construct trees. The proposed Hybrid XGboost - SVM Similarity Model has a training time that is slightly longer than that of Standalone XGboost, because of the added time taken to compute similarities. Despite this overhead being longer, it is justified by the added performance. Of special note is that there is no unnecessary similarity computations. After similarity scores are calculated, XGboost efficiently classifies. The results show that with the proposed model, there is a good balance with the complexity of the calculations and the performance of classifying.

6. Conclusion

This paper is proposing a detailed sentiment classification framework which combines classical machine learning and high-level ensemble learning to tackle problems in the field of sentiment analysis. With the construction of an XGBoost classifier with SVM-based similarity modeling, the study improves the efficiency in the computational resources and the discrimination of the semantics of the different classes of sentiment. The design of the study and result analysis prove that in a learning-paradigm dominated by hybrids, the units with classifier models in isolation can achieve a unbounded improvement in the classifier's ability to deal with the dimensionality and the data in context - ambiguous description. The contribution of the study is in the innovative combination of similarity learning and gradient boosting in a cohesive and structured manner. In effect, SVM-based similarity mapping is a filter to assist with the semantics of a given text and enhances the discriminative space in which the features exist, which then reduces the contextual data to be separated into lesser classes before the classification step. This pre-classification step enhances the ability of the XGBoost model to concentrate on the classification data that have not been accounted for and the details of the sentiment that the residual data possesses. The comparative assessments and numerical analyses consistently show improvement in all the measures - of accuracy, precision, recall, F1-score - in contrast to conventional classifiers such as Naive Bayes, Logistic Regression, standalone SVM, and basic boosting models. Additionally, there is nothing that contradicts the analysis of the training time that states that the framework in use prefers to be easy on the resources to achieve results of the same quality as the alternatives that can be used for high volume sentiment analysis. From a wider research point of view, this paper strengthens the significance of hybrid machine learning architectures for sentiment analysis, especially for the case of social media surveillance, opinion mining, and decision-support systems. The survey-based discussions position the proposed strategy within the periphery of literature and trends on similarity-aware and ensemble-based classification of texts, elucidating some of the trends in the literature. The proposed framework may be extended in future research, for instance, with contextual embeddings from transformer-based language models, adaptive feature selection, and multilingual sentiment datasets. The proposed framework, XGBoost-SVM, for instance, facilitates a contribution that is practically relevant and thematically comprehensive to the current theme within text analytics research: sentiment analysis.

References

1. Dogra, V.; Verma, S.; Kavita; Chatterjee, P.; Shafi, J.; Choi, J.; Ijaz, M.F. A Complete Process of Text Classification System Using State-of-the-Art NLP Models. Comput. Intell. Neurosci. 2022, 2022, 1883698. [PubMed]
2. Hachiya, H.; Yoshida, H.; Shimada, U.; Ueda, N. Multi-class AUC maximization for imbalanced ordinal multi-stage tropical cyclone intensity change forecast. Mach. Learn. Appl. 2024, 17, 100569.
3. Liu, Y.; Loh, H.T.; Sun, A. Imbalanced text classification: A term weighting approach. Expert Syst. Appl. 2009, 36, 690–701.
4. Nagy, G.; Zhang, X. Simple statistics for complex feature spaces. In Data Complexity in Pattern Recognition; Springer: Berlin/Heidelberg, Germany, 2006; pp. 173–195.
5. Le, P.Q.; Iliyasu, A.M.; Garcia, J.; Dong, F.; Hirota, K. Representing visual complexity of images using a 3d feature space based on structure, noise, and diversity. J. Adv. Comput. Intell. Intell. Inform. 2012, 16, 631–640.
6. Mars, M. From word embeddings to pre-trained language models: A state-of-the-art walkthrough. Appl. Sci. 2022, 12, 8805.
7. Sinjanka, Y.; Musa, U.I.; Malate, F.M. Text Analytics and Natural Language Processing for Business Insights: A Comprehensive Review. Int. J. Res. Appl. Sci. Eng. Technol. 2023, 11.
8. Bashiri, H.; Naderi, H. Comprehensive review and comparative analysis of transformer models in sentiment analysis. Knowl. Inf. Syst. 2024, 66, 7305–7361.
9. Yadav, A.; Patel, A.; Shah, M. A comprehensive review on resolving ambiguities in natural language processing. AI Open 2021, 2, 85–92.

10. Seneviratne, I.S. Text Simplification Using Natural Language Processing and Machine Learning for Better Language Understandability. Ph.D. Thesis, The Australian National University, Canberra, Australia, 2024.
11. Garg, R.; Kiwelekar, A.W.; Netak, L.D.; Bhate, S.S. Potential use-cases of natural language processing for a logistics organization. In Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough: Latest Trends in AI; Springer: Berlin/Heidelberg, Germany, 2021; Volume 2, pp. 157–191.
12. Kim, Y. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
13. Johnson, R.; Zhang, T. Effective use of word order for text categorization with convolutional neural networks. arXiv 2014, arXiv:1412.1058.
14. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
15. Schmidt, R.M. Recurrent neural networks (rnns): A gentle introduction and overview. arXiv 2019, arXiv:1912.05911.
16. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv 2018, arXiv:1810.04805.
17. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. OpenAI Blog 2019, 1, 9.
18. Brown, T.B. Language models are few-shot learners. arXiv 2020, arXiv:2005.14165.
19. Azevedo, B.F.; Rocha, A.M.A.; Pereira, A.I. Hybrid approaches to optimization and machine learning methods: A systematic literature review. Mach. Learn. 2024, 113, 4055–4097.
20. Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; Kumar, V. Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Comput. Surv. 2022, 55, 1–37.
21. Banu, S.; Ummayhani, S. Text summarization and translation across multiple languages. J. Sci. Res. Technol. 2023, 1, 242–247.
22. Orosoo, M.; Goswami, I.; Alphonse, F.R.; Fatma, G.; Rengarajan, M.; Bala, B.K. Enhancing Natural Language Processing in Multilingual Chatbots for Cross-Cultural Communication. In Proceedings of the 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 11–12 March 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 127–133.
23. Liang, L.; Wang, S. Spanish Emotion Recognition Method Based on Cross-Cultural Perspective. Front. Psychol. 2022, 13, 849083.
24. Ali, S.I.M.; Nihad, M.; Sharaf, H.M.; Farouk, H. Machine learning for text document classification-efficient classification approach. IAES Int. J. Artif. Intell. 2024, 13, 703–710.
25. Valluri, D.; Manne, S.; Tripuraneni, N. Custom Dataset Text Classification: An Ensemble Approach with Machine Learning and Deep Learning Models. In Proceedings of the 2023 3rd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bengaluru, India, 21–23 December 2023.