ISSN 2723-6471

Aspect-Based Sentiment Analysis of Healthcare Reviews from Indonesian Hospitals based on Weighted Average Ensemble

Esther Irawati Setiawan[1,*], Patrick Tjendika[2], Joan Santoso[3], FX Ferdinandus[4], Gunawan[5], Kimiya Fujisawa[6]

[1,3,5] *Information Technology Department, Institut Sains dan Teknologi Terpadu Surabaya, Indonesia*
[2,4] *Informatics Department, Institut Sains dan Teknologi Terpadu Surabaya, Indonesia*
[6] *School of Media Science, Tokyo University of Technology, Japan*

(Received: July 20, 2024; Revised: July 30, 2024; Accepted: August 17, 2024; Available online: October 15, 2024)

Abstract

Public assessments are essential for evaluating hospital quality and meeting patient demand for superior medical treatment. This study offers a novel approach to aspect-based sentiment analysis (ABSA), which consists of aspect extraction, emotion categorization, and aspect classification. The goal is to examine patient reviews (6,711 reviews) from Google assessments of 20 Indonesian hospitals, broken down by categories including cost, doctor, nurse, and other categories. For example, there are 469 good, 66 negative, and 7 neutral ratings for cleanliness and 93 positive, 125 negative, and 19 neutral reviews for pricing in the sample, which covers a range of attitudes. Using the Conditional Random Field (CRF) approach, aspect phrase extraction was refined and word characteristics and positional tags were adjusted, resulting in an improvement in the F1-score from 0.9447 to 0.9578. The Support Vector Machine (SVM) model had the greatest F1-score of 0.8424 out of two strategies used for aspect categorization. With the addition of sentiment words, sentiment classification improved and led by SVM to an ideal F1-score of 0.7913. For aspect and sentiment classification, a Weighted Average Ensemble approach incorporating SVM, Naïve Bayes, and K-Nearest Neighbors was employed, yielding F1-scores of 0.7881 and 0.8413, respectively. The use of an ensemble technique for sentiment and aspect classification and the incorporation of hyperparameter optimization in CRF for aspect term extraction, which led to notable performance gains, are the innovative aspects of this work

Keywords: Aspect-Based Sentiment Analysis, Aspect Term Extraction, Aspect Classification, Sentiment Classification, Conditional Random Field, Weighted Average Ensemble, Support Vector Machine, Naïve Bayes, K-Nearest Neighbors

1. Introduction

Due to the inevitability of getting sick, the hospital is a public facility that everyone will frequent. The hospital, a highly essential public facility, must account for many lives. Consequently, it is believed that hospitals can provide the finest care for their patients, particularly those who require urgent care. Reviews are the societal standard for everything from movies and songs to food and hospitals [1]. This review has a significant impact on how the public perceives something. People can form inferences about something based on the opinions or experiences of others. A positive rating will encourage the public to utilize the facilities. Therefore, it is hoped that the hospital will receive a positive evaluation so people will feel confident and secure in seeking care there [2].

Currently, sentiment analysis in healthcare reviews largely relies on conventional methods, which focus on identifying the overall sentiment of a review as positive, negative, or neutral. However, these traditional approaches have limitations, particularly in their ability to handle hospital reviews' complexity and multifaceted nature. For instance, a single review might simultaneously praise the cleanliness of a facility while criticizing the waiting time. Standard sentiment analysis techniques would struggle to capture these nuanced opinions, often providing a single, overarching sentiment that fails to represent the detailed feedback given by patients accurately.

Aspect-Based Sentiment Analysis significantly improves by addressing these limitations. Aspect Based Sentiment Analysis's allows for detecting sentiments associated with specific aspects or features of the service being reviewed.

***Corresponding author: Esther Irawati Setiawan (esther@istts.ac.id)**
DOI: https://doi.org/10.47738/jads.v5i4.328

This granularity is crucial in healthcare, where patient feedback can cover a broad range of topics such as cleanliness, cost, physician care, food, nursing staff, parking, receptionist and billing, safety, tests and examinations, waiting time, and more. By categorizing and analyzing sentiments related to these specific aspects, Aspect Based Sentiment Analysis's provides a more comprehensive and accurate understanding of patient experiences and perceptions.

With this propose approach, these challenges can be alleviated by categorizing the comments in the review on several crucial features of the hospital. These factors include cleanliness, cost, physician, food, nurse, parking, receptionist and billing, safety, test and examination, waiting time, and no factor. Aspect-based sentiment analysis is a method for detecting positive, neutral, and negative sentiment toward a language aspect [3], [4]. Additionally, Aspect Based Sentiment Analysis [5], [6] has an edge over conventional sentiment analysis. For instance, a single statement may include multiple facets, yet standard sentiment analysis [7] would only identify a single sentiment. Aspect Based Sentiment Analysis can detect sentiment from multiple phrase components simultaneously [8]. Aspect Based Sentiment Analysis will be broken into three primary steps for this research: aspect term extraction, aspect classification, and sentiment classification. Aspect Classification and Sentiment Classification are performed using the Weighted Average Ensemble approach with models of Support Vector Machine, Naïve Bayes, and K-Nearest Neighbors. Aspect Term Extraction is performed using the Conditional Random Field model.

2. Supporting Theory

2.1. Conditional Random Field

The Conditional Random Field (CRF) is a class of discriminative models that excel in producing predictions in situations where contextual information or nearby conditions influence the current forecast. Unlike generative models, which model the joint probability distribution of the observed and target variables, CRFs approach the conditional probability of the target variables given the observed data, making them highly suitable for structured prediction tasks. [8], [9] CRFs are applied to a wide range of problems, including named entity recognition (NER), where the goal is to locate and classify named entities in text; part-of-speech (POS) tagging, which involves assigning parts of speech to each word in a sentence.

Xia et al. in[10] investigated sentiment analysis for online reviews by integrating Conditional Random Fields and Support Vector Machines (SVMs). The authors designed a hybrid model that leverages the sequence labelling capabilities of CRFs and the classification strength of SVMs. By combining these two techniques, the study aimed to improve the accuracy of sentiment classification for textual reviews. The experimental results showed that their approach outperformed traditional single-model methods, providing more precise and reliable sentiment predictions.

Yao and Zheng in [11] introduced an enhanced sentiment analysis framework based on an improved Transformer model coupled with Conditional Random Fields. The authors improved the Transformer model's architecture to better capture contextual information and semantic nuances in the text. They then used CRFs to handle the sequential nature of sentiment labels. This combination aimed to boost the accuracy and robustness of sentiment analysis tasks.

The power of CRFs as seen in Algorithm 1 lies in their ability to model the dependencies between output variables, allowing for more accurate predictions in structured tasks. CRFs use conditional probabilistic models to account for the influence of neighboring elements in the prediction process [12]. The formula for conditional probability is as in equation 1.

$$P(y|x,\lambda) = \frac{1}{Z(x)} \exp \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y_{i-1}, y_i, x, i) \quad (1)$$

Where: y: label, x: text inputs, λ: feature weight, Z: normalization function, n: number of words, f: node dan edge feature

There are various important tasks that are usually involved in the process of implementing a Conditional Random Field (CRF) model. To represent pertinent patterns, features are first taken from the training data. Subsequently, potential data sequences are developed, and the model is optimized for parameter estimation. Accurate predictions are ensured by using the learnt parameters to extract features from the test data and decode the best label sequence for each sample.

 ISSN 2723-6471


```
crf = CRF(
    algorithm='lbfgs',
    c1=0.001,
    c2=0.001,
    max_iterations=100,
    all_possible_transitions=True
   )
print("Training phase!")
crf.fit(X_train, y_train)
```

Figure 1. Training Algorithm

Where: c1: coefficient for L1 regularization, c2: The coefficient for L2 regularization, all_possible_transitions: When True, generates transition features that associate all of possible label pairs (L * L transition features), lbfgs: Gradient descent using the L-BFGS method

2.2. Weighted Average Ensemble

Weighted Average Ensemble (WAE) [13] is an approach comprised of multiple models that are averaged based on their weights to reduce the overall error[14].

The study [15] explored the effectiveness of ensemble learning techniques in sentiment analysis within their study. The authors employed various ensemble methods to combine the predictions of multiple machine learning models, aiming to enhance the accuracy of sentiment classification tasks. Their approach capitalized on the strengths of individual classifiers while mitigating their weaknesses through a voting mechanism. The results demonstrated a significant improvement in performance, with their ensemble model achieving higher accuracy and robustness compared to single classifiers.

Aziz and Dimililer in [16] presented their research on Twitter sentiment analysis using an ensemble weighted majority vote classifier. The study involved collecting a large dataset of tweets and applying various preprocessing techniques to prepare the data for analysis. The authors then implemented an ensemble classifier that combined the outputs of several base models, with each model's contribution weighted according to its accuracy.

The formula for the ensemble is in equation 2.

$$\bar{V}(X) = \sum W_{\alpha} V^{\alpha}(X) \tag{2}$$

Where: $\bar{V}$= ensemble output, X = input, W_{α} = weight of model α, V^{α} = output of model α, with the condition of weight:

$$\sum_{\alpha=1}^{p} W_{\alpha} = 1 \tag{3}$$

Where: p = number of models, W_{α} = weight of model α

Each method produces a different error as in equation 3. For instance, in the first scenario, model 1 creates a small error while model 2 produces a huge error; yet, in other cases, model 1 could provide a significant error while model 2 produces a tiny error. Consequently, integrating a number of these classifiers will minimize the overall error rate. The formula is equation 4.

$$\text{error} = \sum_{i=0}^{n} C(n,i) * e^{i} * (1-e)^{n-i} \tag{4}$$

Where: n = number of models, e = error classifier (assumed that each classifier error is the same), C(n,i) = Combination of i from n

Figure 2 depicts the WAE procedure for this research. The procedure begins with the training conducted by the three models with the identical training set. Then, the outcomes of the three models are divided by the predetermined weight.

 ISSN 2723-6471

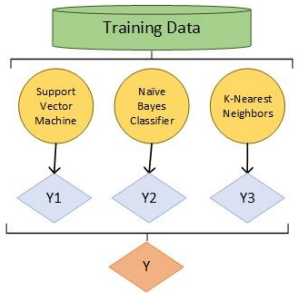

Figure 2. Weighted Average Ensemble workflow

2.3. Support Vector Machine

In this study [17], Bourequat and Mourad proposed a sentiment analysis approach to evaluate public reactions to iPhone releases using SVM techniques. The research aimed to classify sentiments expressed in social media posts and reviews, focusing on features such as design, performance, and user experience. By employing SVM, the authors were able to effectively distinguish between positive and negative sentiments, achieving promising results in classification accuracy. The findings underscored the utility of SVM in sentiment analysis, particularly in capturing the nuanced opinions surrounding technology product launches.

Figure 3 demonstrates that the optimal separating line is the H2 line, which has the greatest margin and splits into two classes. Although H3 does not divide into two classes, whereas H1 does, the difference is relatively small. The hyperplane representing the highest separation or margin between two classes to maximize the distance between the nearest data points on each side is known as the maximum margin hyperplane, and the linear classifier it defines is known as the maximal margin classifier. Maximum–margin hyperplane and margin for SVM learned using support vectors from two classes.

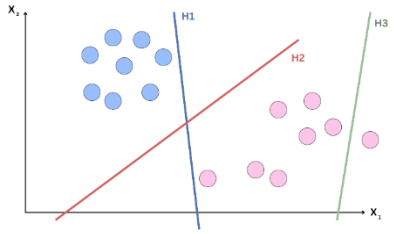

Figure 3. Examples of Some Hyperlanes

Training on the SVM classification will generate a value or pattern that will be employed in the testing procedure designed to assign sentiment labels. Then, an evaluation is conducted by evaluating the score corresponding to the document's side. Figure 4 depicts the decision-making process with SVM as well as the examination of the level of accuracy and the number of documents in each positive, negative, and neutral class [8].

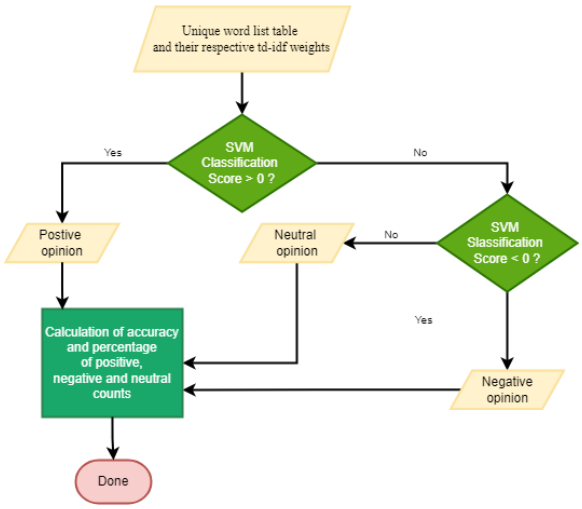

Figure 4. Classification Flowchart with SVM and Analysis

2.4. Naïve Bayes

Naïve Bayes (NB) is a probabilistic machine learning algorithm that can be used in various categories. Typical applications consist of spam filtering, document classification, and sentiment prediction. Based on the work of Rev. Thomas Bayes, NB was developed. Changes in a feature's value do not immediately affect or alter the value of other features. The benefit of NB is that the algorithm can be easily implemented, and because the model is probabilistic, predictions may be generated extremely quickly. Therefore, NB is easily scalable because the algorithms typically employed by real-world applications must reply promptly to user requests. The NB method is implemented using conditional probabilities. Bayes's rule for NB can be deduced from the two notations listed equation 5 and 6 [18].

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)} \tag{5}$$

$$P(Y|X) = \frac{P(X \cap Y)}{P(X)} \tag{6}$$

The Bayes rule is a method for determining P(Y|X) at the moment of prediction using the training dataset's P(X|Y). If Y has more than two classes, the probability of each class will be determined, and the class with the highest probability will be selected. The formula for the Bayes rule is as equation 7.

$$P(Y|X) = \frac{P(X|Y) \cdot P(Y)}{P(X)} \tag{7}$$

Bayes's rule provides the formula for determining the probability of Y given X. However, in real-world problems, X typically contains multiple variables. Bayes rule can be extended to Naïve Bayes when the features are independent. The formula for NB is as equation 8, 9, and 10, where Y is the target class, k is the label, X is the feature and n is the number of features. The formula can be translated as Equation 8.

$$P(Y = k|X_1 \ldots X_n) = \frac{P(X_1|Y = k) \cdot \ldots \cdot P(X_n|Y = k) \cdot P(Y = k)}{P(X_1) \cdot P(X_n)} \tag{8}$$

2.5. K-Nearest Neighbors

The KNN algorithm is a supervised algorithm-based technique. The objective of the supervised learning algorithm is to discover new patterns, whereas the objective of the unsupervised learning algorithm is to discover patterns in data. KNN Regression is an algorithm that introduces the K-nearest neighbor regression, which is the foundation of the Unsupervised K Nearest Neighbor or UNN approach, which is used to predict the output value in regression [19]. KNN is predicated on the premise of locality in the data space. In the local environment, it is anticipated that pattern x will have the same output value y (or class label) as pattern f. (x). For x' , it is therefore known that the label must be comparable to the label of the nearest pattern, which is represented by the mean of the output values of the nearest sample K. K-Nearest Neighbor (KNN) is based on the notion of finding the shortest distance between the data to be evaluated and the K closest neighbors in the training data. The training data is projected onto a multidimensional space, where each dimension corresponds to a data characteristic. This area is partitioned based on the classification of training data. A point in this space is classified as class c if class c is the most prevalent classification among the point's k nearest neighbors. The equation 9 and 10 is the distance and KNN formula.

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{9}$$

In equation 9, x is the sampling data, y is the testing data, and D is the distance.

$$fknn(x') = \frac{1}{K} \sum_{i \in N_{k(x')}} y_i \tag{10}$$

In equation 10, x' is the estimation, K is the n-neighbor, $N_{(k(x'))}$ is the neighborhood, and y_i is the nearest neighbor output

2.6. K-Fold

K-Fold Cross Validation is a model evaluation method used to evaluate the performance of machine learning models. This method is done by dividing the training data into several parts, then studying the model on some of these parts and measuring the model's performance on the other parts. However, it should be noted that K-Fold Cross Validation

 ISSN 2723-6471

is not a training method but only an evaluation method. K-Fold Cross Validation works by dividing the training data into several parts, which are referred to as "folds." The number of folds made depends on the value of K. For example, if K = 5, the training data will be divided into 5 parts.

2.7. Aspect Based Sentiment Analysis

In this survey [20], Liu et al. reviewed various deep learning methods for aspect-based sentiment analysis, highlighting their effectiveness in extracting nuanced sentiments from text. The study focused on the advancements in model architectures, including convolutional neural networks and recurrent neural networks, which have been applied to diverse datasets across different domains. The authors emphasized the importance of attention mechanisms in improving sentiment classification by enabling models to weigh relevant features more effectively. The findings indicated that deep learning approaches significantly enhance sentiment analysis performance, with notable improvements in accuracy and interpretability compared to traditional methods.

Bahri and Suadaa [21] conducted an aspect-based sentiment analysis on user reviews of Bromo Tengger Semeru National Park, sourced from Google Maps, to evaluate various aspects such as natural beauty, accessibility, and visitor facilities. Their methodology included data preprocessing, aspect extraction, and sentiment classification using a combination of SVM and a lexicon-based approach. The findings revealed that the model effectively identified and analyzed sentiments, providing detailed insights into visitor experiences and satisfaction levels.

3. Aspect-Based Sentiment Analysis

3.1. Aspect Term Extraction

Aspect Term Extraction (ATE) involves identifying and extracting aspect terms from a document. This process uses the IOB (Inside-Outside-Beginning) tagging scheme plus aspect tags. Label B (beginning) indicates the start of an aspect term, label I (inside) indicates the continuation of an aspect term, and label O (outside) indicates that the term is not part of any aspect being considered. For instance, in the sentence "The doctor is good but the place for parking is very narrow," "doctor" and "place for parking" would be considered aspect terms for the 'doctor' and 'parking' aspects, respectively. The labels would be {B-Doctor, good: O, but: O, place: B-Parking, parking: I-Parking, very: O, narrow: O}. Preprocessing includes translating non-Indonesian text, cleaning (removing symbols, emails, phone numbers, URLs, and repetitive characters), converting to lowercase, normalizing text, removing stopwords, tokenizing, performing POS tagging, and stemming. Feature extraction uses a Multi-Label Binarizer (MLB) to convert identified aspect terms into a binary format suitable for machine learning models. Figure 5 illustrates the input and output of the Aspect Term Extraction process.

Figure 5 starts with the review's input in the form of textual remarks. Then, text preprocessing is performed by translating text outside of Indonesian, cleaning the text by removing symbols, emails, phone numbers, URLs, and consecutive same characters with more than two occurrences, lowering all letters, normalizing, and tokenizing. Then, POS tagging is performed for CRF features. Following this, the application will extract features. For the current word, the following features are utilized: bias (1.0), word.lower(), the last three letters of the word, the last two letters of the word, word.isupper(), word.istitle(), word.isdigit(), POS tag, and the first two letters POS tag. For words before and after, word.lower(), word.istitle(), word.isupper(), POS tag, and the first two letters of the POS tag are utilized. Then, CRF will Aspect Term Extraction these features and generate IOB labels as output.

 ISSN 2723-6471

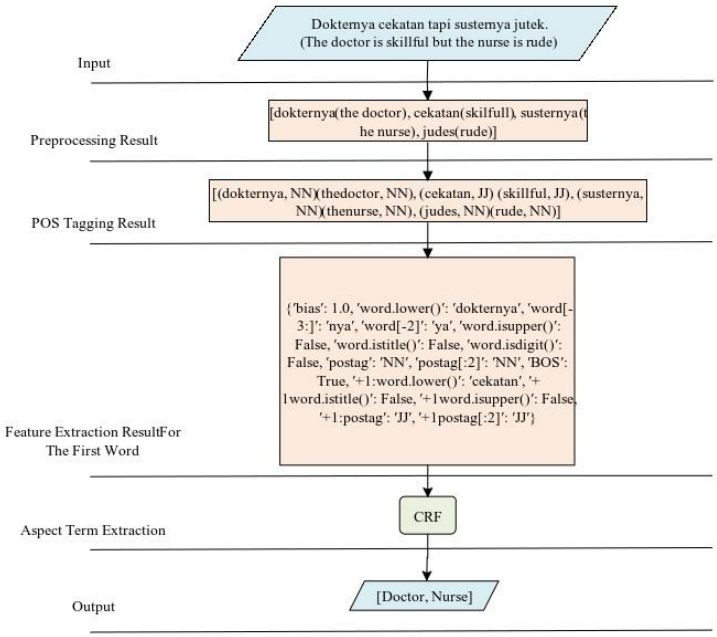

Figure 5. Aspect Term Extraction Input and Output

3.2. Aspect Classification

Aspect Classification (AC) involves classifying textual aspects identified in the reviews. In this study, Aspect Classification follows the methodology described in publication [22]. The WAE models, including Naive Bayes (NB), Support Vector Machine, and KNN, are utilized for this classification task. The output from Aspect Term Extraction enhances the Aspect Classification process by providing additional context. Preprocessing steps are similar to those in Aspect Term Extraction: translating, cleaning, converting to lowercase, normalizing, removing stopwords, tokenizing, POS tagging, and stemming. Features are extracted using a MLB for aspect terms and a tf-idf Vectorizer for the text. The ensemble models then classify the identified aspects, and figure 6 shows the Aspect Classification input and output.

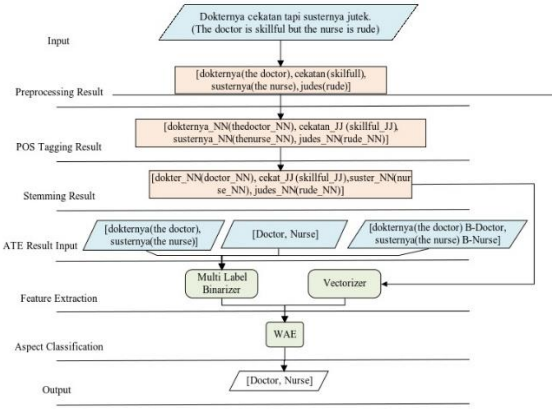

Figure 6. Aspect Classification Input and Output

The operation of the Aspect Classification input and output is depicted in Figure 6. The input for Aspect Classification (AC) begins with the textual reviews. The preprocessing steps include translating text not in Indonesian, cleaning by removing symbols, emails, phone numbers, URLs, and characters repeating more than twice. This is followed by lowering the text to lowercase, normalizing, removing stopwords, and tokenizing. Unlike in the Aspect Term Extraction (ATE) step, stopwords are removed because they are not considered aspect terms in Aspect Classification. Part-of-speech aspetagging and stemming are then performed. The aspect words identified by Aspect Term Extraction are incorporated as features. The features are extracted using a MLB for Aspect Term Extraction features and a tf-idf Vectorizer for the text. The WAE models, including Naive Bayes, Support Vector Machine, and KNN, classify the aspects, and the output is a set of identified aspects.

 ISSN 2723-6471

3.3. Sentiment Classification

Sentiment Classification (SC) involves determining the sentiment of the text, specifically in the context of identified aspects. This study's SC methodology is based on previous research. In addition to using the same WAE models (NB, SVM, and KNN) as in Aspect Classification, SC incorporates output from both Aspect Term Extraction and Aspect Classification to improve accuracy. The preprocessing steps are identical to those used in Aspect Term Extraction and Aspect Classification: translation, cleaning, lowercase conversion, normalization, stopword removal, tokenization, POS tagging, and stemming. Emotional terms are identified using a sentiment glossary, distinguishing between positive and negative sentiments. Features are extracted using a MLB for sentiment keywords and aspect terms, and a tf-idf Vectorizer for the text. The WAE models classify the sentiment of each aspect, providing an output that includes both the aspects and their corresponding sentiments. Figure 7 depicts the SC input and output.

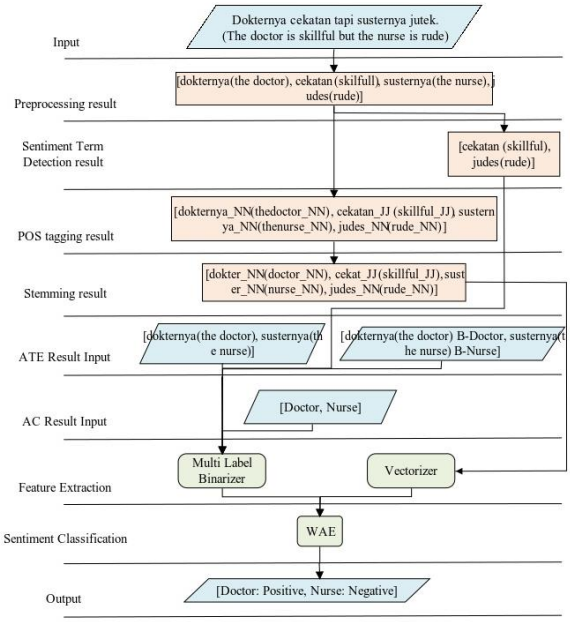

Figure 7. Sentiment Classification Input and Output

Figure 7 illustrates the input and output operations of the SC. The process for SC starts similarly with the textual reviews as input. The text undergoes the same preprocessing steps as in Aspect Classification, including translation, cleaning, normalization, stopword removal, and tokenization. Emotional terms are identified using a sentiment glossary, which separates positive and negative sentiments. POS tagging and stemming follow. The aspect terms from Aspect Term Extraction and the identified aspects from Aspect Classification are added to the features. Feature extraction uses MLB for sentiment keywords and Aspect Term Extraction/Aspect Classification features, and tf-idf Vectorizer for the text. The WAE models classify the sentiment of each aspect, producing an output that includes both the aspects and their corresponding sentiments.

4. Experiment and Results

The dataset used in this study was gathered from Google reviews of 20 hospitals in Indonesia. 5 K-fold cross-validation will be used for the assessment, with the micro F1-score being the main focus. The F1-score is chosen to be highlighted because it offers a more nuanced assessment of model performance than depending only on accuracy, recall, or precision alone [23]. It does this by comprehensively balancing precision and recall. Table 1 displays the statistics of the dataset.

Table 1. Dataset Statistics

Aspect	Positive	Negative	Neutral	Total
Cleanliness	469	66	7	542
Cost	93	125	19	125

Doctor	586	254	20	254
Food	120	37	9	37
Nurse	490	275	21	275
Parking	139	129	13	129
Receptionist and Billing	154	430	11	430
Safety	25	14	1	14
Test and Examination	282	142	13	142
Waiting Time	286	675	23	675
No Aspect	1178	386	219	1783
Total	3822	2533	356	6711

CRF is the model utilized for Aspect Term Extraction. For the current word, the following features are utilized: bias (1.0), word.lower(), the final three letters of the word, the final two letters of the word, word.isupper(), word.istitle(), word.isdigit(), heading tags, and the initial two letters of the post tag. For words before and following, word.lower(), word.istitle(), word.isupper(), heading tags, and the first two letters of heading tags are utilized. The hyperparameter algorithms are lbfgs, c1 and c2 0.001, max iterations 100, and all possible transations. True. The Aspect Term Extraction results are presented in table 2.

Table 2. Aspect Term Extraction

Iteration	Result			After Optimization			Result from First Approaches		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
1	0.94	0.92	0.9333	0.95	0.94	0.9484	0.84	0.79	0.8131
2	0.94	0.92	0.9255	0.94	0.93	0.9365	0.81	0.77	0.7873
3	0.96	0.94	0.9464	0.96	0.96	0.9586	0.84	0.79	0.8148
4	0.96	0.94	0.9526	0.97	0.97	0.9686	0.83	0.79	0.8070
5	0.97	0.96	0.9656	0.98	0.97	0.9768	0.83	0.78	0.8013
Average	0.954	0.936	0.9447	0.96	0.954	0.9578	0.83	0.784	0.8047

The CRF's hyperparameters are optimized to generate superior performance. Following optimization, the parameters c1 and c2 were determined to be 0.3998 and 0.0088. The successful hyperparameter optimization of the CRF is shown in table 2 column 5,6,7. Before optimization, the mean f1 score was only 0.9447. When optimized, its value climbs to 0.9578. For Aspect Classification, I employ two strategies. The first method utilizes elements derived from the outcomes of word annotation. The outcomes of the initial strategy are shown in table 2 column 5,6,7. The second method employs three WAE models: NB, SVM, and KNN. C=1 is the SVM hyperparameter, while k=5 is the KNN hyperparameter. Word embedding (WE) using tf-idf and aspect term (AT) derived from Aspect Term Extraction is used as the feature. Table 3 display the second strategy's model-specific outcomes.

Table 3. Result on Aspect Classification

Feature	Iteration	NB Result			SVM Result			KNN Result		
		Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
WE	1	0.98	0.10	0.1737	0.87	0.76	0.8121	0.46	0.31	0.3686
	2	0.98	0.10	0.1803	0.86	0.76	0.8088	0.44	0.30	0.3558
	3	0.93	0.11	0.1951	0.88	0.78	0.8280	0.45	0.31	0.3704
	4	0.98	0.12	0.2197	0.87	0.77	0.8156	0.44	0.30	0.3544
	5	0.95	0.09	0.1627	0.88	0.75	0.8095	0.42	0.27	0.3317
	Average	0.964	0.104	0.1863	0.872	0.764	0.8148	0.442	0.298	0.3562
WE+AT	1	0.82	0.75	0.7820	0.88	0.81	0.8444	0.85	0.71	0.7727

ISSN 2723-6471

2	0.78	0.74	0.7632	0.86	0.79	0.8239	0.83	0.69	0.7499
3	0.81	0.77	0.7908	0.89	0.83	0.8605	0.85	0.72	0.7766
4	0.81	0.76	0.7880	0.88	0.81	0.8461	0.85	0.72	0.7767
5	0.79	0.74	0.7628	0.88	0.80	0.8369	0.84	0.67	0.7458
Average	0.802	0.752	0.7774	0.878	0.808	0.8424	0.844	0.702	0.7643

Table 3 column 3,4,5 demonstrates that AT significantly impacts the outcomes of NB. Without AT, the average f1-score is only 0.1863, but with AT, the average f1-score rises to 0.7774. The average f1-score grew by fourfold. Table 3 column 6,7,8 demonstrates that SVM has been able to categorize aspects fairly successfully without the use of AT. Without AT, the average f1-score is 0.8148; with AT, the average f1-score rises to 0.8424. Table 3 column 9,10,11 demonstrates that the results of KNN without AT are unsatisfactory. But still superior to the NB result. Without AT, the average f1-score is 0.3562; with AT, the average f1-score rises to 0.7643. The average f1 score grew by than fourfold. Each Weighted Average Ensemble model is required to have a weight. In this study, the weight of each model is decided by the average of the top 5 k-fold cross-validation combinations. Table 4 displays the search results for the optimal weight combination in Aspect Classification.

Table 4. Weight Combination Search Result on Aspect Classification

Feature	Iteration	Weight (NB, SVM, KNN)
WE	1	(0.00, 0.98, 0.02)
	2	(0.00, 0.98, 0.02)
	3	(0.01, 0.97, 0.02)
	4	(0.00, 0.99, 0.01)
	5	(0.00, 0.98, 0.02)
	Average	(0.00, 0.98, 0.02)
WE+AT	1	(0.05, 0.95, 0.00)
	2	(0.01, 0.98, 0.01)
	3	(0.00, 1.00, 0.00)
	4	(0.02, 0.98, 0.00)
	5	(0.03, 0.97, 0.00)
	Average	(0.02, 0.98, 0.00)

Table 4 demonstrates that SVM gains more weight than the other two models. This occurs because, as seen in the previous table, SVM produces the best results.

Table 5. Weighted Average Ensemble Result on Aspect Classification

Feature	Iteration	Precision	Recall	F1-score
WE	1	0.87	0.76	0.8137
	2	0.86	0.76	0.8096
	3	0.88	0.79	0.8302
	4	0.87	0.77	0.8146
	5	0.88	0.75	0.8106
	Average	0.872	0.766	0.8157
WE+AT	1	0.88	0.81	0.8445
	2	0.86	0.79	0.8231
	3	0.89	0.83	0.8592
	4	0.88	0.81	0.8465
	5	0.88	0.80	0.8372

ISSN 2723-6471

	Average	0.878	0.808	0.8421

As shown on table 5 WAE results on Aspect Classification, it can be shown that the WAE results (0.8421) with the weight of the average best combination of 5 k-fold cross-validation still cannot match the SVM results (0.8424). The Aspect Classification model is optimized with respect to its hyperparameters to improve performance. Following optimization, it was determined that alpha=1.0 is the optimal hyperparameter for NB, C=1.5 is the optimal hyperparameter for SVM, and k=3 is the optimal hyperparameter for KNN.

Table 6. Model Optimization Result on Aspect Classification

Feature	Model	F1-score
WE+AT	NB	0.7774
	SVM	0.8425
	KNN	0.7665

Table 6 demonstrates that SVM is still the best model. After optimization, there is no difference in NB results. The SVM yield rose a little from 0.8424 to 0.8425. The KNN result became 0.7665, up from 0.7643.

Table 7. The Best Weight Combination Search Result After Aspect Classification Optimization

Feature	Iteration	Weight (NB, SVM, KNN)
WE+AT	1	(0.00, 0.99, 0.01)
	2	(0.00, 0.95, 0.05)
	3	(0.04, 0.95, 0.01)
	4	(0.13, 0.87, 0.00)
	5	(0.02, 0.81, 0.17)
	Average	(0.04, 0.91, 0.05)

After optimization, the search for the optimal weight combination was conducted once again. Table 7 demonstrates that SVM still gains more weight than the other two models. The weight of NB and KNN is significantly raised during optimization.

Table 8. Weighted Average Ensemble Result After Optimization on Aspect Classification

Feature	Iteration	Precision	Recall	F1-score
WE+AT	1	0.88	0.81	0.8443
	2	0.86	0.79	0.8222
	3	0.89	0.83	0.8570
	4	0.89	0.81	0.8472
	5	0.88	0.80	0.8356
	Average	0.88	0.808	0.8413

As shown in table 8, the average f1-score on WAE decreases due to the model's optimization results. Before optimization, the average f1-score was 0.8421; after optimization, it was 0.8413. SVM results (0.8425) continue to be superior to WAE results (0.8413). We used the same model for SC as for Aspect Classification. The employed features are WE and Sentiment Term (ST). Table 5, table 6, and table 7 display the model-specific outcomes of the second methodology. And table 9 shows the performance of Gemini 1.5 Pro results for Aspect Classification.

Table 9. Comparison of proposed model with Gemini 1.5

Aspect	Precision	Recall	F1-score
Food	0.91	0.69	0.78
Test and Examination	0.83	0.4	0.54
Cost	0.68	0.92	0.78

 ISSN 2723-6471

Nurse	0.6	0.99	0.75
Doctor	0.77	0.95	0.85
Waiting Time	0.83	0.64	0.72
Safety	0.12	0.82	0.21
Receptionist and Billing	0.63	0.76	0.69
Cleanliness	0.75	0.99	0.85
Parking	0.94	0.98	0.96
No Aspect	0.88	0.8	0.84
Average	0.721818182	0.812727273	0.724545455

Table 10. Feature NB Result on Sentiment Classification

Feature	Iteration	Precision	Recall	F1-score
WE	Cost	0.72	0.608	0.6591
	Nurse	0.692	0.672	0.6815
	Doctor	0.696	0.682	0.6899
	Cleanliness	0.892	0.886	0.8899
	Receptionist and Billing	0.75	0.734	0.7423
	Food	0.726	0.726	0.7232
	No Aspect	0.672	0.664	0.6666
	Parking	0.666	0.486	0.5622
	Test and Examination	0.726	0.684	0.7052
	Waiting Time	0.844	0.822	0.8318
	Safety	0.606	0.6	0.6067
	Average	0.7264	0.6876	0.7053

Table 11. Feature NB Result on Sentiment Classification

Feature	Iteration	F1-score
WE+ST	Cost	0.6985
	Nurse	0.8144
	Doctor	0.7800
	Cleanliness	0.8953
	Receptionist and Billing	0.7714
	Food	0.7232
	No Aspect	0.7205
	Parking	0.6387
	Test and Examination	0.7961
	Waiting Time	0.8593
	Safety	0.6250
	Average	0.7566

Table 10 and table 11 demonstrates that the outcomes of NB with the assistance of ST improve the performance of nearly all models, with the exception of the Food element. Without ST, the average f1-score was just 0.7053, however with ST, the average f1-score climbed to 0.7566.

 ISSN 2723-6471

Table 12. SVM on Sentiment Classification

Feature	Iteration	F1-score
WE	Nurse	0.8424
	Doctor	0.8351
	Cleanliness	0.9306
	Receptionist and Billing	0.8201
	Food	0.7158
	No Aspect	0.7581
	Parking	0.7063
	Test and Examination	0.7577
	Waiting Time	0.8641
	Safety	0.6400
	Average	0.7794
WE+ST	Cost	0.6943
	Nurse	0.8394
	Doctor	0.8573
	Cleanliness	0.9370
	Receptionist and Billing	0.8244
	Food	0.7362
	No Aspect	0.7895
	Parking	0.7027
	Test and Examination	0.7924
	Waiting Time	0.8721
	Safety	0.6588
	Average	0.7913

Table 12 demonstrates that the results of SVM with the ST feature enhance the performance of many models. Several models perform better without the ST feature, including the Cost, Nurse, and Parking aspects. Without ST, the average f1-score was 0.7794, however with ST, the average f1-score increased to 0.7913.

Table 13. KNN Result on Sentiment Classification

Feature	Iteration	F1-score
WE	Cost	0.6340
	Nurse	0.7234
	Doctor	0.7360
	Cleanliness	0.9197
	Receptionist and Billing	0.7419
	Food	0.6679
	No Aspect	0.6698
	Parking	0.6164
	Test and Examination	0.7321
	Waiting Time	0.8409
	Safety	0.6500
	Average	0.7211
WE+ST	Cost	0.6690

ISSN 2723-6471

	Nurse	0.7164
	Doctor	0.7215
	Cleanliness	0.9133
	Receptionist and Billing	0.7744
	Food	0.6522
	No Aspect	0.6837
	Parking	0.5633
	Test and Examination	0.7016
	Waiting Time	0.8384
	Safety	0.7000
	Average	0.7213

Table 13 demonstrates that the outcomes of KNN with the ST feature only raise the f1-score in a few models, such as Cost, Receptionist and Billing, No Aspect, and Safety. Over fifty percent of models function better without the ST. Still, the average f1-score with the ST feature is superior. Without ST, the average f1-score was 0.7211, however in the presence of ST, the average f1-score increased to 0.7213. The weights of the three models are then obtained for WAE in the same manner as for Aspect Classification. Table 14 displays the search results for the optimal weight combination in SC.

Table 14. Weight Combination Search Result on Sentiment Classification

Feature	Aspect	Weight (NB, SVM, KNN)
WE	Cost	(0.13, 0.61, 0.27)
	Nurse	(0.10, 0.74, 0.16)
	Doctor	(0.01, 0.85, 0.14)
	Cleanliness	(0.06, 0.57, 0.37)
	Receptionist and Billing	(0.13, 0.66, 0.21)
	Food	(0.07, 0.60, 0.33)
	No Aspect	(0.02, 0.90, 0.07)
	Parking	(0.16, 0.71, 0.13)
	Test and Examination	(0.10, 0.66, 0.24)
	Waiting Time	(0.15, 0.41, 0.44)
	Safety	(0.00, 0.33, 0.67)
WE+ST	Cost	(0.15, 0.44, 0.41)
	Nurse	(0.10, 0.72, 0.18)
	Doctor	(0.09, 0.81, 0.10)
	Cleanliness	(0.00, 0.70, 0.30)
	Receptionist and Billing	(0.07, 0.64, 0.29)
	Food	(0.00, 0.68, 0.32)
	No Aspect	(0.06, 0.86, 0.08)
	Parking	(0.19, 0.70, 0.11)
	Test and Examination	(0.37, 0.60, 0.03)
	Waiting Time	(0.25, 0.58, 0.17)
	Safety	(0.10, 0.10, 0.80)

Table 14 demonstrates that the combination of weights generated by SC is more uniformly distributed than that of Aspect Classification, which is dominated by SVM. Despite the fact that most models continue to utilize SVM.

ISSN 2723-6471

Table 15. Weighted Average Ensemble Result on Sentiment Classification

Feature	Iteration	F1-score
WE	Cost	0.6858
	Nurse	0.8331
	Doctor	0.8192
	Cleanliness	0.9289
	Receptionist and Billing	0.8065
	Food	0.7198
	No Aspect	0.7549
	Parking	0.6911
	Test and Examination	0.7620
	Waiting Time	0.8680
	Safety	0.6500
	Average	0.7745
WE+ST	Cost	0.6962
	Nurse	0.8395
	Doctor	0.7152
	Cleanliness	0.7791
	Receptionist and Billing	0.7087
	Food	0.7905
	No Aspect	0.8771
	Parking	0.6250
	Test and Examination	0.7850
	Waiting Time	0.8475
	Safety	0.9334
	Average	0.8227

Table 15 demonstrates that almost all models that employ the WE+ST feature are superior to those that use the WE feature alone, except the Food and Safety elements. Without ST, the average f1-score was 0.7745, however with ST, the average f1-score climbed to 0.7850. SC models are optimized with respect to their hyperparameters in an effort to enhance performance.

Table 16. Model Optimization on Sentiment Classification Result

Feature	Model	F1-score
WE+AT	NB	0.7630
	SVM	0.7909
	KNN	0.7061

As with Aspect Classification, table 16 demonstrates that SVM is still the best model after optimization. The NB yield rose from 0.7566 to 0.7330. The output of the SVM decreased somewhat from 0.7913 to 0.7909. Similarly, KNN results declined from 0.7213 to 0.7061. After optimization, the search for the optimal weight combination was conducted once again.

Table 17 displays D the optimal weight combination after optimization. The SVM continues to gain more weight than the other two models. However, the average weight of SVM reduced from 0.62 to 0.58 after optimization. This suggests that the importance of NB and KNN has grown.

 ISSN 2723-6471

Table 17. Weight Combination Search Result After Sentiment Classification Optimization

Feature	Aspect	Weight (NB, SVM, KNN)
WE+ST	Cost	(0.34, 0.39, 0.27)
	Nurse	(0.21, 0.69, 0.10)
	Doctor	(0.17, 0.78, 0.04)
	Cleanliness	(0.24, 0.47, 0.29)
	Receptionist and Billing	(0.17, 0.45, 0.38)
	Food	(0.00, 0.73, 0.27)
	No Aspect	(0.18, 0.79, 0.03)
	Parking	(0.20, 0.69, 0.11)
	Test and Examination	(0.38, 0.55, 0.08)
	Waiting Time	(0.28, 0.56, 0.16)
	Safety	(0.19, 0.32, 0.50)

The optimized WAE findings are shown in table 18. Table 18 demonstrates that the optimization results have a significant impact on WAE. The average f1 score went from 0.7851 to 0.7871. However, the most excellent result on Sentiment Classification still belongs to SVM with (0.7909).

Table 18. Weighted Average Ensemble Result After Sentiment Classification Optimization

Feature	Iteration	F1-score
WE+ST	Cost	0.7183
	Nurse	0.8429
	Doctor	0.8559
	Cleanliness	0.9303
	Receptionist and Billing	0.8238
	Food	0.7068
	No Aspect	0.7796
	Parking	0.7177
	Test and Examination	0.7976
	Waiting Time	0.8769
	Safety	0.6083
	Average	0.7871

We also did an experiment with Gemini 1.5 Pro and by prompting, we evaluated the Aspect Classification and our model resulted a comparable performance as Gemini and shown on table 19, with the details of each aspect displayed on table 9.

Table 19. Performance Evaluation with other models

Model	F1
Proposed Approach	0.8413
Gemini [24]	0.7245

5. Conclusions

This research highlights key advancements in enhancing sentiment analysis for healthcare reviews. It demonstrates the efficiency and ease of implementing Flask and MongoDB, specifically MongoDB Atlas, for managing automatically hosted databases. A notable contribution is the refinement of the Aspect-Based Sentiment Analysis procedure, which now includes Aspect Classification and a Sentiment Term function in Sentiment Classification. The enhancement of

the Aspect Term Extraction label to incorporate both IOB and Aspect tags significantly improved the classification model's performance.

The study found that, despite advancements, the WAE approach was unsuccessful. The results of the SVM model differed significantly from those of the NB and KNN models. However, the classification model's performance was improved by adding features such as aspect term and sentiment term.

For future research, expanding the dataset, experimenting with text preprocessing and feature extraction techniques (such as Word2Vec, FastText, and BERT), and optimizing hyperparameters are recommended. The study also suggests exploring advanced models like BERT or GPT for sentiment analysis to achieve substantial improvements. These findings underscore the importance of careful model selection and feature enhancement, providing a solid foundation for advancing sentiment analysis in healthcare reviews.

6. Declarations

6.1. Author Contributions

Conceptualization: E.I.S. and P.T.; methodology: J.S. and F.X.F.; software: G.; validation: K.F. and E.I.S.; formal analysis: P.T.; investigation: J.S.; resources: F.X.F.; data curation: G.; writing—original draft preparation: E.I.S.; writing—review and editing: P.T. and K.F.; visualization: G.; supervision: K.F.; project administration: E.I.S.; funding acquisition: F.X.F. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This work was partially funded by Institut Sains dan Teknologi Terpadu Surabaya (ISTTS) under Institute for Research and Community Services or Lembaga Penelitian dan Pengabdian kepada Masyarakat (LPPM).

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Afzaal, M. Usman, and A. Fong, "Tourism mobile app with aspect-based sentiment classification framework for tourist reviews," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 2, pp. 233–242, 2019.

[2] E. I. Setiawan, W. Dharmawan, K. J. Halim, J. Santoso, F. X. Ferdinandus, K. Fujisawa, M. H. Purnomo, "Indonesian News Stance Classification Based on Hybrid Bidirectional LSTM and Transformer Based Embedding," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 5, pp. 517–537, 2024.

[3] B. Ray, A. Garain, and R. Sarkar, "An ensemble-based hotel recommender system using sentiment analysis and aspect categorization of hotel reviews," *Appl Soft Comput*, vol. 98, no. 1, pp. 1-18, 2021.

[4] B. Kane, A . Essebbar, O . Guinaudeau, V . Chiesa, I . Quénel and S. Chau, "CNN-LSTM-CRF for Aspect-Based Sentiment Analysis: A Joint Method Applied to French Reviews.," *in ICAART (1)*, vol. 2021, no. 1, pp. 498–505.

[5] E. I. Setiawan, F. Ferry, J. Santoso, S. Sumpeno, K. Fujisawa, and M. H. Purnomo, "Bidirectional GRU for Targeted Aspect-Based Sentiment Analysis Based on Character-Enhanced Token-Embedding and Multi-Level Attention.," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 392-407, 2020.

 ISSN 2723-6471

[6] S. Imron, E. I. Setiawan, J. Santoso, M. H. Purnomo, and others, "Aspect Based Sentiment Analysis Marketplace Product Reviews Using BERT, LSTM, and CNN," *Journal of Systems Engineering and Information Technology*, vol. 7, no. 3, pp. 586–591, 2023.

[7] S. Nuhmana, "Sentiment Analysis of Related Public Opinion Smart City Pasuruan Regency on Media Social Using Svm Algorithm," *Journal of Social Research*, vol. 2, no. 4, pp. 1305–1310, 2023.

[8] C. R. Aydin and T. Güngör, "Combination of recursive and recurrent neural networks for aspect-based sentiment analysis using inter-aspect relations," *IEEE Access*, vol. 8, no. 4, pp. 77820–77832, 2020.

[9] C. Sutton, A. McCallum, and others, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[10] S. Zheng et al., "Conditional random fields as recurrent neural networks," *in Proceedings of the IEEE international conference on computer vision*, vol. 2015, no. 12, pp. 1529–1537.

[11] H. Xia, Y. Yang, X. Pan, Z. Zhang, and W. An, "Sentiment analysis for online reviews using conditional random fields and support vector machines," *Electronic Commerce Research*, vol. 20, no. 2, pp. 343–360, 2020.

[12] L. Yao and N. Zheng, "Sentiment Analysis Based on Improved Transformer Model and Conditional Random Fields," *IEEE Access*, vol. 12, no. 6, pp. 90145-90157, 2024.

[13] W. Gong, C. Zhao, C. H. Juang, Y. Zhang, H. Tang, and Y. Lu, "Coupled characterization of stratigraphic and geo-properties uncertainties–a conditional random field approach," *Eng Geol*, vol. 294, no. 12, pp. 1-16, 2021.

[14] S. Han, Y. Liu, and J. Yan, "Neural network ensemble method study for wind power prediction," *in 2011 asia-pacific power and energy engineering conference*, vol. 2011, no. 5, 2011, pp. 1–4.

[15] M. S. H. Talukder and A. K. Sarkar, "Nutrients deficiency diagnosis of rice crop by weighted average ensemble learning," *Smart Agricultural Technology*, vol. 4, no. 8, pp. 13, 2023.

[16] J. Kazmaier and J. H. Van Vuuren, "The power of ensemble learning in sentiment analysis," *Expert Syst Appl*, vol. 187, no. 1, pp. 1-16, 2022.

[17] R. H. H. Aziz and N. Dimililer, "Twitter sentiment analysis using an ensemble weighted majority vote classifier," *in 2020 International Conference on Advanced Science and Engineering (ICOASE)*, vol. 2020, no. 12, pp. 103–109.

[18] W. Bourequat and H. Mourad, "Sentiment analysis approach for analyzing iPhone release using support vector machine," *International Journal of Advances in Data and Information Systems*, vol. 2, no. 1, pp. 36–44, 2021.

[19] I. Wickramasinghe and H. Kalutarage, "Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation," *Soft comput*, vol. 25, no. 3, pp. 2277–2293, 2021.

[20] M. Nanja and P. Purwanto, "Forward Selection-Based k-Nearest Neighbor Method for Commodity Price Prediction of Pepper," *Pseudocode*, vol. 2, no. 1, pp. 53–64, 2015.

[21] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Trans Comput Soc Syst*, vol. 7, no. 6, pp. 1358–1375, 2020.

[22] C. A. Bahri and L. H. Suadaa, "Aspect-based sentiment analysis in bromo tengger semeru national park indonesia based on google maps user reviews," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 17, no. 1, pp. 79–90, 2023.

[23] D. Ekawati and M. L. Khodra, "Aspect-based sentiment analysis for Indonesian restaurant reviews," *in 2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*, vol. 2017, no. 8, pp. 1–6, 2017.

[24] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," *in Proceedings of the first workshop on evaluation and comparison of NLP systems*, vol. 2020, no. 11, pp. 79–91.

[25] J. Luzano, "Pedagogical Influence of an AI Chatbot Gemini in Mathematics Education," *International Journal of Academic Pedagogical Research*, vol. 8, no. 4, pp. 107–112, 2024.