Article

A Feature-Based Approach for Sentiment Quantification Using Machine Learning

Kashif Ayyub ^{1,*} , Saqib Iqbal ², Muhammad Wasif Nisar ¹, Ehsan Ullah Munir ¹, Fawaz Khaled Alarfaj ³ and Naif Almusallam ³

¹ Department of Computer Science, Wah Campus, COMSATS University Islamabad, Wah Cantt 45550, Islamabad Capital Territory, Pakistan; wasif@ciitwah.edu.pk (M.W.N.); ehsan@ciitwah.edu.pk (E.U.M.)

² College of Engineering, Al Ain University, Al Ain 64141, United Arab Emirates; saqib.iqbal@aau.ac.ae

³ Department of Computer Science and Information, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11564, Saudi Arabia; fkarfaj@imamu.edu.sa (F.K.A.); nyalmusallam@imamu.edu.sa (N.A.)

* Correspondence: kashifayyub@ciitwah.edu.pk

Abstract: Sentiment analysis has been one of the most active research areas in the past decade due to its vast applications. Sentiment quantification, a new research problem in this field, extends sentiment analysis from individual documents to an aggregated collection of documents. Sentiment analysis has been widely researched, but sentiment quantification has drawn less attention despite offering a greater potential to enhance current business intelligence systems. In this research, to perform sentiment quantification, a framework based on feature engineering is proposed to exploit diverse feature sets such as sentiment, content, and part of speech, as well as deep features including word2vec and GloVe. Different machine learning algorithms, including conventional, ensemble learners, and deep learning approaches, have been investigated on standard datasets of SemEval2016, SemEval2017, STS-Gold, and Sanders. The empirical-based results reveal the effectiveness of the proposed feature sets in the process of sentiment quantification when applied to machine learning algorithms. The results also reveal that the ensemble-based algorithm AdaBoost outperforms other conventional machine learning algorithms using a combination of proposed feature sets. The deep learning algorithm RNN, on the other hand, shows optimal results using word embedding-based features. This research has the potential to help diverse applications of sentiment quantification, including polling, trend analysis, automatic summarization, and rumor or fake news detection.

Keywords: sentiment analysis; sentiment quantification; deep features; machine learning; deep learning

Citation: Ayyub, K.; Iqbal, S.; Wasif Nisar, M.; Munir, E.U.; Alarfaj, F.K.; Almusallam, N. A Feature-Based Approach for Sentiment Quantification Using Machine Learning. *Electronics* **2022**, *11*, 846. <https://doi.org/10.3390/electronics11060846>

Academic Editors: Juan M. Corchado and Gemma Piella

Received: 13 January 2022

Accepted: 3 March 2022

Published: 8 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The social web has changed the way people communicate. The emergence of social media channels has resulted in the rapid creation of textual content. People create and post their content using social interaction platforms such as the web, discussion forums, Facebook, Twitter, etc. The rapid growth of content has sentiment information, which offers the potential for researchers to obtain people's opinion through social media about entities including business, academia, products, marketing, etc. To extract meaningful information from raw data, a famous field known as sentiment analysis is in trend [1,2].

Sentiment analysis is an active research area that classifies opinions in negative, positive, and neutral texts. It also finds the grade of polarity (high, moderate, and mild). Sentiment analysis is carried out on three levels: document level, sentence level, and phrase level. Document-level sentiment analysis is the most popular and is followed by numerous opinion mining techniques. Document-level sentiment analysis creates groups of documents and classifies the target documents into the required set of classes. For binary classification, the target documents are classified as positive or negative, while for tertiary classification the required classes include positive, negative, and neutral. Document-level sentiment

analysis does not consider diverse factors for analysis. Apart from document-level analysis, sentence-level analysis considers each sentence and counts its single supposition. Sentence-level sentiment analysis is based on the subjectivity of sentences. Document-level and sentence-level sentiment analysis do not give a clear understanding of the polarity of the text. Sentiment analysis has various research areas, including subjectivity analysis, sentiment polarity detection [3], sentiment quantification, etc. [4].

Sentiment quantification deals with the estimation of class labels of individual content. For sentiment quantification, various methods that include Classify and Count, Adjusted Count, and Instance-based Quantification Trees [5] are commonly used in different studies. However, an analysis of previous classification algorithms shows that for quantification, these standard algorithms are not an optimal solution. In this regard, research suggests that quantification should be considered as a different approach compared to classification and this research problem should be addressed using diverse approaches [6]. Hence, it opens new research opportunities to explore different approaches and develop new methods in this domain.

Consequently, it raises the need to devise sentiment quantification-based methods that deliver high accuracy. To address the issue of accuracy, this research contributes to the field of sentiment quantification as follows:

- Novel feature sets are proposed such as pos, tweet, specific, content, and sentiment, with the ranking of features carried out using feature selection approaches.
- Deep features including word2vec and GloVe are used for sentiment analysis, and these features are considered for sentiment quantification.
- Machine learning approaches have been investigated, including: (1) traditional techniques—Support Vector Machine (SVM), Naïve Bayes (NB), and Decision Tree (DT); (2) ensemble learners—Random Forest (RF) and AdaBoost; (3) deep learning-based—Deep Belief Network (DBN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN).
- The results for sentiment quantification are computed on SemEval2016, SemEval2017, STS-Gold, and Sanders. Standard performance evaluation measures, including Kullback–Leibler divergence (KLD), relative absolute error (RAE), and absolute error (AE), are applied for the evaluation of classifiers.

The remainder of this paper is organized as follows: Section 2 provides a review of existing research studies in the relevant literature. Section 3 provides details of the proposed research methodology. Section 4 provides a comprehensive discussion of the empirical-based results. Section 5 concludes the paper.

2. Related Work

Here, we discuss the existing research work of quantification based on sentiment analysis, which further divides into three main classes: aggregated methods, non-aggregated methods, and ensemble-based methods.

2.1. Sentiment Quantification

Sentiment quantification is the process of detecting frequent data. It is also known as prevalence estimation [7]. Quantification is used in different fields to deal with aggregated data. Sentiment quantification has various research applications, with some of them discussed here. The sentiment quantification method is used to detect communities [8], quantification of cross-lingual language [9], public health monitoring [10], and for tweet classification [11].

2.1.1. Aggregated Methods

The quantification approach is preferred to predict the class prior probabilities. Classify and Count (CC) is a famous technique used for quantification. However, CC is lacking in the estimation of the class distribution. Newer approaches are presented to overcome the processing limitation of CC based on Sample Means Matching (SMM). SMM is very

effective in quantifying a large amount of data per second. Twenty-five datasets are taken to perform the experiments. The proposed technique has outperformed the existing methods of quantification [12]. Further, a model titled Ordinal Quantification based on trees is proposed. The purpose of the proposed technique is to accurately count the frequency of each class of unlabeled items in text. In ordinal quantification, the order of the class is defined. The same approach is utilized to find the highest stars in products' reviews by analyzing their class prevalence over time. The proposed approach is evaluated on the SemEval2016 dataset and outperforms state-of-the-art methods [13].

Classifying data in deep layers is a complex task. Various techniques such as Neural and Statistical Machine Translation are used for this purpose, but are lacking in encoding and decoding while learning data from deep layers. To address these issues, Expectation Maximization (EM) is a quantification technique used for the automatic detection of errors in Arabic text to overcome the shortcomings of neural and statistical translation methods. EM dynamically combines information around layers. Moreover, during training, Kullback–Leibler divergence (KLD) is used to improve the model's performance. The proposed approach is evaluated on two standard datasets, namely QALB-2014 and QALB-2015. The experiments showed that the proposed approach outperformed the previous techniques in terms of F1 score [14].

EM is also applied in the field of rumor detection for Arabic tweets. The proposed method is based on a semi-supervised (EM) method to extract the user and content-based features from tweets. Both feature sets are tested to check their significance. The proposed feature sets are trained through a semi-supervised (EM) method with a small base of labelled data. The proposed method was compared with the Gaussian method and outperformed the baseline with 78.6% accuracy [15].

An estimation of class proportions based on counting the classification errors is also used for classification purposes. A new method is proposed to adjust the classification errors by building confidence intervals. The model is introduced for the quantification of social media. The proposed approach is better than previous approaches that used the accurate estimation intervals [16].

The CC method has given rise to many other derived methods. “QuaNet”, another derived method, is introduced using Recurrent Neural Network (RNN) to learn “quantification embeddings”. These embeddings are firstly learned by a model then elaborated by CC. This approach is tested on Kindle, IMDb, and HP datasets. The results have shown the effectiveness of this model over existing quantification techniques [17].

2.1.2. Non-Aggregated Methods

González-Castro et al. [18] developed a model to quantify data based on a divergence measure. Hellinger distance is used for data distribution, validation, and to find the mismatch between test and validation. Prior probability estimation is used to minimize divergence. HDx and HDy are two types of Hellinger distance, where HDy needs output from the classifier and HDx does not need input from the classifier. Hopkins et al. introduced a non-parametric technique to quantify data [19]. The proposed approach quantifies data without any need of classification. American presidential blogs were selected as a dataset, with the proposed method decreasing the unbiased estimation. Software application was developed to quantify thousands of opinions about the US presidency.

2.1.3. Ensemble-Based Methods

Ensemble learners combine some weak learners. Some aggregated methods are combined to address the data distribution issues in sentiment analysis. Count HDy and Adjusted Classify are combined to make an ensemble model. CC, AC, PCC, PAC, and HDy are applied for learning the proposed ensemble model. Two schemes are presented to learn and predict. All learners are used to give prediction, then four sets of measures are applied to select the best model [20].

Ensemble methods give optimal results by building various training sets. Each model is then trained using data distribution techniques for quantification. The proposed methods categorize the errors of data distribution to enhance the performance of ensemble learners. The model explicitly addresses the binary quantification problem by focusing on the change in the expected distribution for each class. The results have shown that ensemble-based method have outperformed prior techniques [21].

The ensemble method is also explored in the field of soundscape ecology. A new approach is introduced which combines quantification and classification to train the CNN to classify classes of birds. The experiments show that the quantification performed better than the classification for the classification of bird species [22].

To obtain the optimal accuracy for sentiment quantification, machine learning techniques reported promising results. However, due to the sensitive nature of sentiments in the opinion-seeking process, there is a need to achieve more optimal results. Non-lexicon-based approaches are not widely applied for sentiment quantification. The role of diverse features can be exploited to find their role in improving the classification accuracy of sentiment quantification along with non-lexicon approaches. Some of these studies are summarized in Table 1.

Table 1. Quantification Techniques.

Method	Approach	Year	Feature(s)	Dataset
Aggregated	Expectation Minimization (EM) [15]	2022	User and content-based features.	Tweets
	Expectation Minimization (EM) [14]	2022	Quantifying errors in text	QALB-2014, QALB-2015
	Sample Means Matching (SMM) [12]	2020	Quantifying billions of data elements in seconds.	25 benchmark datasets
	QuaNet [17]	2018	Quantification of data.	Kindle, IMDB, HP (Harry Potter)
	OQT [13]	2016	Quantifying Tweets	SemEval2016
Non-Aggregated	Automated Nonparametric Content Analysis [18]	2010	Quantification of data.	Blogs
	HDy and HDx [18]	2013	Quantification of data.	UCI datasets
Ensemble-based	Ensembles for Quantification [21]	2017	Data distribution and quantification.	UCI datasets, Sentiment140
	Dynamic Ensembles [20]	2019	Quantification of data. Applied techniques based on ensemble learners.	UCI datasets

2.2. Problem Statement and Formulation

Accuracy is an important parameter in the field of sentiment analysis. In the literature, various feature sets have been exploited using machine learning techniques to improve the results. However, there is still a need to investigate those feature sets for the emerging domain of sentiment quantification and to further improve accuracy because of the sensitive nature of sentiment in the opinion-seeking process. There is a need to contribute to the field of sentiment quantification to inquire about the impact of feature sets for sentiment quantification. In addition, as existing research studies only focus on machine learning, there is a need to explore deep learning approaches.

Formally, the research problem is to estimate the dispersion of a set $D = \{d_1, d_2, \dots, d_q\}A$ of unlabeled documents across a set $C = \{c_1, c_2, \dots, c_p\}S$ of classes. In our research, the relevant literature deals with $|C| = 3$. There are three classes: positive, neutral, and neg-

ative. As our focus is the SLMC quantification task, we consider the measures that have been proposed for evaluation. Some of the notations include quantification loss denoted by $\nabla(\hat{p}, p, D, C)$, error estimation denoted by ∇ , and distribution denoted by p for set D and class C by another distribution $\hat{p}$.

3. Proposed Research Methodology

This segment describes the approach used to quantify tweets based on sentiment analysis. A framework is proposed to give insights into the steps followed for sentiment quantification. A detail discussion follows on the feature engineering, algorithms applied, dataset considered, and performance evaluation measure used in this research.

3.1. Framework for Sentiment Quantification

The proposed model demonstrates the procedure carried out for sentiment quantification, as shown in Figure 1. In the first step, cleansing of the standard datasets (SemEval2016, SemEval2017, STS-Gold, and Sanders) is performed using data preprocessing techniques. Data preprocessing includes spaces removal, tokenization achievement, stop-words removal, case conversion, removal of words of less than three letters, and lemmatization for content feature extraction. In the second step, features based on content, POS (part of speech), tweet specific, and sentiment features are extracted using libraries of Python. Parameter settings for optimizing all classifiers are shown in Table 7. Further, the traditional machine learning approaches and deep learning approaches such as NB, AdaBoost, DT, RF, SVM and RNN, CNN_LSTM, and DBN are applied for sentiment quantification. Afterward, to count and classify the instance of data, the Classify and Count (CC) method is applied. Next, to evaluate the performance of machine learning classifiers, performance evaluation measures are applied for sentiment quantification.

Figure 1. The proposed model for sentiment quantification.

3.2. Feature Engineering

Feature engineering consists of feature extraction and selection to achieve optimal accuracy. The selection of features has a major impact in achieving the desired results. Here, the discussion is divided into subparts such as proposed features, baseline features, and deep features to elaborate on the features' impact on quantification accuracy. It also presents the selection and ranking of the features.

3.2.1. Proposed Feature Sets

To perform sentiment quantification, sentiment-based features are extracted through a sentiment-based lexicon, Vader. Vader is well known for the computation of sentiment features and is also used in various research studies [23,24].

POS tagging is applied to understand the nature of content-based features. The verb feature is used to obtain the action of an entity. The adjective count is considered because adjectives show the negative and positive characteristics of a tweet. The content-based feature exploits the diverse characteristics of text in tweets. Question marks, exclamation marks, and special characters are counted to check whether a person is asking a question or trying to attract attention. The retweet feature is also important to check if a tweet contains facts. If a tweet is frequently retweeted, it potentially relates to a sensitive topic containing more sentiments. The mention feature is considered to check if another person is added to the discussion. Moreover, the URL feature is used to obtain the number of URLs shared by users to support their point of view. The hashtag contains the content topic; therefore, this feature is also considered. A list of proposed features is shown in Table 2.

Table 2. Proposed feature sets for tweets.

Sr#	Categories	Description	Symbol
1	Sentiment	Sentiment score of the tweet	S_{sent}^T
2		Sentiment—Number of positive words	S_{PW}^T
3		Sentiment—Number of negative words	S_{NW}^T
4		Count of positive emoticons	S_{PE}^T
5		Count of negative emoticons	S_{NE}^T
6	POS	Number of nouns in a tweet	P_N^T
7		Number of pronouns in a tweet	P_P^T
8		Verbs frequency in a tweet	P_V^T
9		Adjectives frequency in a tweet	P_A^T
10		Number of special symbols	C_{SS}^T
11	Content	Number of WH words in a tweet	C_{WH}^T
12		Number of question marks in a tweet	C_{QM}^T
13		Number of exclamation marks	C_{EM}^T
14		Number of capitalized words	C_{RW}^T
15		Number of quoted words	C_{QW}^T
16	Tweet Specific	Number of retweets	T_{RT}^T
17		Number of mentions	T_M^T
18		Number of URLs	T_{URL}^T
19		Hashtag length	T_{HL}^T
20		Is it a tweet or retweet?	C_{RT}^T
21		Number of hashtags	T_{HT}^T
22		Number of capitalized hashtags	T_{CH}^T

Features such as n -gram, word2vec and Bag-of-Words (BoW) are also used for sentiment quantification. The n -gram is a technique of word embedding, while BoW is used in natural language processing (NLP). BoW takes frequencies of each word to train a classifier. Term frequency-inverse document frequency (TF-IDF) finds the frequency of a particular word in text. Word2vec and GloVe are used to represent words into vectors. The feature word2vec also finds the syntactic and semantics' similarity between words, while GloVe divides words into clusters to find similar and dissimilar words.

3.2.2. Feature Selection and Ranking

Optimal features increase the performance of classifiers. To find the optimal feature sets, three widely used feature selection methods are applied: Information Gain (IG), Gain Ratio (GR), and Relief-f. IG is suitable for biased data and decreases due to its mutual information formula. GR computes the difference of attributes and considers the features with small difference. Relief-F computes the closest neighbors for all attributes.

Optimal features are selected through feature selection techniques, with those features having less importance and a negative influence on the target class omitted. Features ranked by feature selection algorithms according to their importance are shown in Table 3.

Table 3. Feature engineering for tweets.

Sentiment	Parts of Speech	Content	Tweet Specific	Baseline	Deep
Sentiment score of the tweet	Verbs frequency in a tweet	Number of WH words in a tweet	Number of mentions	TF-IDF	GloVe
Sentiment—Number of negative words	Adjectives frequency in a tweet	Number of question marks in a tweet	Number of retweets	n -gram	Word2vec
Sentiment—Number of positive words	Number of nouns in a tweet	Number of quoted words	Number of URLs	BoW	
Count of negative emoticons	Number of pronouns in a tweet	Number of repetitive words	Number of hashtags		
Count of positive emoticons		Number of special symbols	Number of capitalized hashtags		
		Number of exclamation marks	Hashtag length		
Is it a tweet or retweet?					

Table 3 shows that the sentiment features have a greater impact than other features. Negative sentiments and negative emoticons have a greater impact and are ranked higher than positive sentiments, with their importance consistent with existing research studies [25]. The adjective and adverbs features of POS have high scores and show the action of an object. Verbs have a greater impact than nouns due to their nature and show the attributes of any entity. POS features have a greater impact for predicting the target class.

Content features such as WH, quoted, and repetitive content-based features have high scores due to the subjective and opinionative nature of features, respectively [26]. Then, special characters followed by exclamation marks are ranked higher, showing the discussion within content. Another content-based feature, URL, has a high score as this focuses on the opinion of the subject. Hashtags are ranked low as they relate to topics that are important both in objective and opinionative content. Then follows hashtags that contain topics of the content which can be retweets and contain no sentiments.

Among the baseline features, TF-IDF and n -gram have higher scores than BoW. Among deep features, GloVe has a higher score than word2vec due to the fast processing of training. In addition, GloVe combines the benefits of the word2vec-based skip-gram model in word analogy tasks such as sentiment analysis and stance classification.

3.3. Classification Algorithms Applied

This subsection discusses the machine learning technique applied for sentiment quantification. Machine learning techniques are divided into three categories: traditional algorithms, ensemble learners, and deep learners.

3.3.1. Machine Learning Techniques

Some machine learning approaches applied on tweets for sentiment quantification are discussed.

Support Vector Machine (SVM)

SVM is based on linear regression. It uses high margins for high-dimensional data to classify negative and positive features. SVM optimization is calculated through the formula shown in Equations (1) and (2).

$$\max_{f(i_1, i_2, \dots, i_x)} = \sum_{x=1}^n i_k - \frac{1}{2} \sum_{y=1}^n q_x i_x (c_x c_y) q_x i_x \quad (1)$$

$$\sum_{x=1}^n q_x i_x = 0, 0 \leq i_x \leq C \quad (2)$$

where n is the number of trainings, i is a linear combination of training inputs, q is the training output, m is the cost function, and x and y measure the similarity of the dot product of c . SVM is not suitable for noisy and large datasets due to more execution time being required for the training process.

Naïve Bayes (NB)

NB is a conditional probability-based classifier. The NB probability estimation formula is given in Equation (3), where X is an event, Z is the evidence, $P(X)$ is the probability of an event before the evidence is seen, and $P(X|Z)$ is the probability of an event after the evidence is seen.

$$P(X|Z) = \frac{P(X)P(Z|X)}{P(X)} \quad (3)$$

NB gives better performance for categorical data than numerical data.

Decision Tree (DT)

DT is based on the rule of data decision and works on the principal of entropy and Information Gain (IG) techniques. DT helps to reduce the execution time of preprocessing for missing attributes. The entropy is calculated using the formula in Equation (4).

$$En(X) = \sum_{h=1}^m P_h \log_2 P_h \quad (4)$$

where P_h is the probability that an attribute belongs to class m . To process the information in bits, the $\log_2$ function is used. While $En(X)$ is the required entropy for the class label, it is also known as Entropy.

3.3.2. Ensemble Learning Techniques

AdaBoost

AdaBoost is an ensemble method that aggregates the strong and weak learners. This technique helps to give more accurate decisions for predicting the target class. This method is also favorable for attributes that are misclassified during prediction. While training, each element is assigned a weight. The weight assignment calculation is shown in Equation (5).

$$\text{Weight}(e_i) = \frac{1}{y} \quad (5)$$

where e is the number of elements to be trained. Misclassified instances are computed as shown in Equation (6).

$$\text{Err} = \frac{(\text{Corr} - X)}{X}. \quad (6)$$

Random Forest (RF)

RF is based on the technique of regression models. RF is suitable for high-variation data and calculate average to compute. RF works on a strategy of votes to calculate responses on data attributes. This approach uses a bagging method k times. For $Q = 1 \dots m$, RF trains its regression tree by the formula given in Equation (7).

$$\frac{1}{Q} \sum_{i=1}^m f_x(R) \quad (7)$$

3.3.3. Deep Learning Techniques

Deep Belief Networks (DBN)

DBN is a deep learning technique that follows the method of probability and statistics. DBN architecture contains hidden layers and blocks, with layers interconnected but blocks separated from each other. For sentiment analysis, DBN is in trend to exploit its efficiency for prediction [27].

CNN-LSTM

CNN (Convolutional Neural Network) is a deep learner but is not capable of calculating long-distance dependencies in data. LSTM (Long Short-Term Memory) can work well with long-distance dependencies and is combined with CNN to achieve the desired result for any biased datasets. CNN, along with LSTM, is applied for sentiment analysis and sequence-based text processing [28].

Recurrent Neural Network (RNN)

RNN is also a deep learner and is preferable for text processing and language translation. It works on the rule of memory. The previous output is saved and fed as input for the next phase. This strategy helps its sequential processing. RNN is applied for sentiment analysis [29].

3.4. Datasets

This subsection discusses the details of datasets selected for experimentation.

3.4.1. SemEval2016

SemEval2016 is a widely used dataset for quantification. SemEval2016 includes five tasks and contains tweets such as Feminist Movement (949 tweets), Abortion (933 tweets), Atheism (733 tweets), Hillary Clinton (984 tweets), and Climate Change (564 tweets). The details of the dataset are shown in Table 4. This dataset has been used in earlier studies [30,31].

Table 4. Tweet statistics of investigated datasets.

Dataset	Total Tweets	Testing Tweets	Training Tweets
SemEval2016	68,197	51,851	16,346
SemEval2017	62,617	12,284	50,333
STS-Gold	1,600,000	320,000	1,280,000
Sanders	5512	1102	4410

3.4.2. SemEval2017

SemEval2017 is a famous multilingual dataset. This dataset consists of tweets in two languages: Arabic and English. English tweets are higher in number than Arabic tweets, which are only 19% of the dataset. The dataset contains 6100 testing and 3355 training tweets in Arabic, with 12,284 testing and 50,333 training tweets are in English. The details of the dataset, which has been used in earlier studies [32,33], are shown in Table 4.

3.4.3. STS-Gold

STS-Gold may present different sentiment labels because tweets and targets (entities) are annotated individually [34]. This dataset contains 1.6 million manually classified tweets. There were 1.28 million tweets used for training and 3.2 million tweets used for testing purposes.

3.4.4. Sanders

The Sanders [35,36] [N2-N4] dataset is manually labelled by one annotator and consists of 5512 tweets. We have used 4410 tweets for training and 1102 tweets for testing purposes.

3.5. Performance Evaluation Measures

This subsection describes the performance evaluation measures used for sentiment quantification.

3.5.1. Absolute Error (AE)

This measure corresponds to the average absolute difference between the predicted class prevalence and the true class prevalence, using Equation (8).

$$AE(\hat{p}, p) = \frac{1}{|C|} \sum_{c_j \in C} |\hat{p}(c_j) - p(c_j)| \quad (8)$$

3.5.2. Relative Absolute Error (RAE)

Relative absolute error (RAE) addresses the trouble that occurred in normalized absolute error by scaling the value $|\hat{p}(c_j) - p(c_j)|$ in Equation (9) with the true class prevalence.

$$RAE(\hat{p}, p) = \frac{1}{|C|} \sum_{c_j \in C} \frac{|\hat{p}(c_j) - p(c_j)|}{p(c_j)} \quad (9)$$

3.5.3. Kullback–Leibler Divergence (KLD)

Another measure that has become the standard metric of quantification is normalized cross-entropy, better known as Kullback–Leibler divergence (KLD), which is used as a quantification measure and is defined in Equation (10).

$$KLD(\hat{p}, p) = \sum_{c_j \in C} p(c_j) \log \frac{p(c_j)}{\hat{p}(c_j)} \quad (10)$$

4. Results and Discussion

According to the literature, there is room to improve the accuracy for sentiment-based quantification. Sentiment quantification is not addressed with feature-based approaches to achieve the desired accuracy. To address this problem, we have proposed various feature sets to reach the optimal accuracy for quantification of tweets based on sentiment analysis. To evaluate our feature-based framework, machine learning approaches which are subdivided into three levels, conventional algorithms, ensemble learners, and deep learning approaches, are applied on the SemEval2016, SemEval2017, STS-Gold, and Sanders datasets. To evaluate the performance of classifiers, performance evaluation metrics are applied.

4.1. Single Feature Sets

Detailed experiments are performed to evaluate the effectiveness of our proposed features for the sentiment quantification task. To achieve this aim, each proposed feature set is tested on both datasets to obtain detailed analysis. To evaluate their impact, NB, SVM, and DT conventional algorithms and AdaBoost and RF ensemble learners are applied on each set including POS, content, sentiment, and tweet specific. The results suggest that POS features show more effective results when applied with AdaBoost than when evaluated through performance evaluation metrics. AdaBoost dominated the other classifiers in terms of a lower error rate (KLD = 0.0213) for SemEval2016, SemEval2017 (KLD = 0.0214), STS-Gold (KLD = 0.0129), and Sanders (KLD = 0.0169), as shown in Table 5.

Table 5. Comparison of single feature sets using ML classifiers.

Dataset	Features	NB			DT			SVM			AdaBoost			RF		
		AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
SemEval2016	S	0.033	0.691	0.025	0.040	0.798	0.030	0.030	0.599	0.023	0.029	0.595	0.022	0.030	0.634	0.023
	P	0.044	0.898	0.034	0.037	0.750	0.028	0.028	0.575	0.022	0.028	0.571	0.021	0.042	0.853	0.032
	C	0.039	0.802	0.030	0.035	0.708	0.026	0.029	0.579	0.022	0.036	0.715	0.027	0.037	0.750	0.028
	T	0.055	1.097	0.042	0.029	0.603	0.022	0.029	0.596	0.022	0.029	0.592	0.022	0.053	1.056	0.040
SemEval2017	S	0.030	0.618	0.022	0.041	0.823	0.031	0.029	0.577	0.022	0.028	0.576	0.022	0.035	0.732	0.026
	P	0.049	0.986	0.037	0.035	0.710	0.027	0.028	0.572	0.021	0.027	0.547	0.020	0.030	0.607	0.023
	C	0.043	0.873	0.032	0.032	0.657	0.025	0.028	0.574	0.022	0.033	0.671	0.026	0.032	0.657	0.024
	T	0.072	1.453	0.056	0.027	0.565	0.021	0.028	0.575	0.022	0.028	0.572	0.021	0.029	0.579	0.022
STS-Gold	S	0.021	0.434	0.016	0.033	0.661	0.025	0.019	0.378	0.014	0.019	0.378	0.014	0.027	0.566	0.020
	P	0.042	0.851	0.032	0.026	0.532	0.020	0.018	0.375	0.014	0.017	0.347	0.013	0.020	0.413	0.016
	C	0.035	0.723	0.027	0.023	0.472	0.018	0.019	0.376	0.014	0.024	0.485	0.019	0.023	0.473	0.018
	T	0.069	1.385	0.053	0.018	0.370	0.014	0.019	0.377	0.014	0.018	0.374	0.014	0.019	0.380	0.015
Sanders	S	0.025	0.532	0.019	0.037	0.748	0.028	0.024	0.484	0.018	0.024	0.483	0.018	0.031	0.655	0.023
	P	0.045	0.923	0.035	0.031	0.627	0.024	0.024	0.480	0.018	0.022	0.454	0.017	0.026	0.517	0.020
	C	0.039	0.803	0.030	0.028	0.571	0.021	0.024	0.482	0.018	0.029	0.584	0.022	0.028	0.571	0.021
	T	0.071	1.421	0.054	0.023	0.474	0.017	0.024	0.482	0.018	0.024	0.479	0.018	0.024	0.486	0.019

4.2. Combination of Feature Sets

To take the experiments to the next step, the proposed feature sets are combined with each other to determine the optimal pair of feature sets. The proposed features are combined in groups such as “sentiment + content” (SC), “sentiment + tweet specific” (ST), “sentiment + POS” (SP), “POS + tweet specific” (PT), “POS + content” (PC), “content + tweet specific” (CT), “sentiment + POS + content” (SPC), “sentiment + content + tweet specific” (SCT), “sentiment + POS + tweet specific” (SPT), “POS + content + tweet specific” (PCT), and all feature sets. The results have shown that when all proposed features are combined, they outperform all single feature sets. SVM outperformed other classifiers when applied with all feature sets “sentiment + POS + content + tweet specific” (SPCT). SVM has more promising results with a lower error rate for all four datasets, with KLD = 0.014 for SemEval2016, 0.013 for SemEval2017, 0.0051 for STS-Gold, and 0.0092 for Sanders, as shown in Tables 6–9, respectively.

Table 6. Comparison of feature sets' combinations using ML classifiers on SemEval2016.

Features	NB			DT			SVM			AdaBoost			RF		
	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
SP	0.036	0.736	0.027	0.035	0.713	0.027	0.025	0.519	0.019	0.025	0.512	0.019	0.033	0.680	0.025
SC	0.032	0.671	0.024	0.033	0.679	0.025	0.025	0.506	0.019	0.028	0.575	0.022	0.029	0.613	0.022
ST	0.040	0.824	0.031	0.030	0.611	0.022	0.024	0.499	0.018	0.024	0.494	0.018	0.038	0.773	0.029
PC	0.037	0.758	0.028	0.031	0.630	0.023	0.023	0.467	0.017	0.026	0.537	0.020	0.034	0.705	0.026
PT	0.045	0.916	0.035	0.027	0.558	0.020	0.022	0.461	0.017	0.022	0.454	0.016	0.043	0.869	0.033
CT	0.042	0.850	0.032	0.025	0.523	0.019	0.022	0.448	0.016	0.025	0.520	0.019	0.039	0.801	0.030
SPC	0.032	0.664	0.024	0.030	0.617	0.023	0.021	0.431	0.016	0.023	0.477	0.017	0.029	0.606	0.022
SPT	0.037	0.769	0.028	0.027	0.565	0.021	0.020	0.423	0.015	0.020	0.418	0.015	0.035	0.716	0.026
SCT	0.035	0.719	0.026	0.026	0.538	0.019	0.020	0.410	0.015	0.022	0.457	0.017	0.032	0.663	0.024
PCT	0.038	0.788	0.029	0.024	0.505	0.018	0.019	0.389	0.014	0.021	0.436	0.016	0.036	0.736	0.027
SPCT	0.034	0.706	0.025	0.025	0.524	0.019	0.018	0.379	0.014	0.020	0.412	0.015	0.031	0.649	0.023

Table 7. Comparison of feature sets combination using ML classifiers on SemEval2017.

Features	NB			DT			SVM			AdaBoost			RF		
	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
SP	0.036	0.742	0.027	0.035	0.705	0.027	0.025	0.505	0.019	0.024	0.489	0.018	0.029	0.606	0.022
SC	0.032	0.669	0.024	0.033	0.665	0.025	0.024	0.491	0.018	0.027	0.542	0.020	0.029	0.617	0.022
ST	0.048	0.971	0.036	0.029	0.604	0.022	0.023	0.476	0.018	0.023	0.473	0.017	0.028	0.575	0.022
PC	0.041	0.843	0.031	0.028	0.581	0.022	0.022	0.462	0.017	0.024	0.501	0.019	0.026	0.525	0.019
PT	0.057	1.157	0.044	0.025	0.517	0.019	0.022	0.448	0.016	0.021	0.431	0.016	0.024	0.475	0.018
CT	0.053	1.083	0.041	0.023	0.474	0.017	0.021	0.434	0.016	0.024	0.486	0.018	0.024	0.489	0.018
SPC	0.033	0.695	0.025	0.029	0.593	0.022	0.020	0.420	0.015	0.022	0.445	0.016	0.025	0.521	0.019
SPT	0.044	0.903	0.033	0.026	0.546	0.020	0.020	0.406	0.015	0.019	0.395	0.014	0.024	0.488	0.018
SCT	0.041	0.848	0.031	0.025	0.514	0.019	0.019	0.392	0.014	0.021	0.427	0.016	0.024	0.492	0.018
PCT	0.048	0.981	0.036	0.022	0.457	0.016	0.018	0.378	0.014	0.019	0.403	0.015	0.021	0.430	0.016
SPCT	0.040	0.829	0.030	0.024	0.495	0.018	0.017	0.364	0.013	0.018	0.383	0.014	0.022	0.450	0.016

Table 8. Comparison of feature sets combination using ML classifiers on STS-Gold.

Features	NB			DT			SVM			AdaBoost			RF		
	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
SP	0.028	0.575	0.021	0.026	0.528	0.020	0.015	0.299	0.011	0.014	0.283	0.010	0.020	0.419	0.015
SC	0.024	0.494	0.018	0.024	0.483	0.018	0.014	0.283	0.010	0.017	0.340	0.013	0.021	0.434	0.015
ST	0.041	0.835	0.031	0.020	0.414	0.015	0.013	0.267	0.010	0.013	0.265	0.010	0.019	0.379	0.014
PC	0.033	0.691	0.025	0.019	0.387	0.014	0.012	0.252	0.009	0.014	0.294	0.011	0.016	0.323	0.012
PT	0.052	1.046	0.040	0.015	0.316	0.011	0.011	0.236	0.009	0.010	0.219	0.008	0.013	0.261	0.010
CT	0.047	0.963	0.036	0.013	0.270	0.010	0.011	0.220	0.008	0.014	0.278	0.010	0.014	0.279	0.010
SPC	0.025	0.524	0.019	0.020	0.402	0.015	0.010	0.205	0.007	0.011	0.234	0.009	0.015	0.325	0.012
SPT	0.037	0.759	0.028	0.017	0.349	0.013	0.009	0.190	0.007	0.009	0.178	0.006	0.014	0.281	0.010
SCT	0.034	0.698	0.025	0.015	0.314	0.011	0.008	0.174	0.006	0.010	0.213	0.008	0.014	0.288	0.010
PCT	0.041	0.847	0.031	0.012	0.251	0.009	0.008	0.159	0.006	0.009	0.188	0.007	0.010	0.214	0.008
SPCT	0.032	0.676	0.024	0.014	0.294	0.011	0.007	0.144	0.005	0.008	0.165	0.006	0.012	0.241	0.009

Table 9. Comparison of feature sets combination using ML classifiers on Sanders.

Features	NB			DT			SVM			AdaBoost			RF		
	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
SP	0.032	0.665	0.024	0.031	0.622	0.023	0.020	0.409	0.015	0.019	0.393	0.014	0.025	0.519	0.019
SC	0.028	0.588	0.021	0.028	0.580	0.022	0.019	0.394	0.015	0.022	0.447	0.017	0.025	0.532	0.019
ST	0.044	0.908	0.034	0.025	0.515	0.019	0.018	0.379	0.014	0.018	0.376	0.014	0.024	0.484	0.018
PC	0.037	0.772	0.028	0.024	0.491	0.018	0.018	0.364	0.013	0.020	0.404	0.015	0.021	0.431	0.016
PT	0.055	1.105	0.042	0.020	0.423	0.015	0.017	0.349	0.013	0.016	0.332	0.012	0.019	0.375	0.014
CT	0.050	1.027	0.038	0.018	0.379	0.014	0.016	0.334	0.012	0.019	0.389	0.014	0.019	0.391	0.015
SPC	0.029	0.615	0.022	0.025	0.504	0.019	0.015	0.320	0.012	0.017	0.347	0.013	0.020	0.430	0.015
SPT	0.041	0.836	0.031	0.022	0.454	0.017	0.015	0.305	0.011	0.014	0.294	0.010	0.019	0.392	0.014
SCT	0.037	0.778	0.028	0.020	0.420	0.015	0.014	0.290	0.010	0.016	0.327	0.012	0.019	0.397	0.014
PCT	0.045	0.918	0.034	0.017	0.361	0.013	0.013	0.276	0.010	0.014	0.303	0.011	0.016	0.329	0.012
SPCT	0.036	0.758	0.027	0.019	0.401	0.014	0.012	0.262	0.009	0.013	0.281	0.010	0.017	0.352	0.013

4.3. Optimal Feature Sets

The results analysis is also represented in Figures 2 and 3. Results have shown the impact of POS as a single feature set. POS contains the action of an object and also important information. Thus, when applied with machine learning algorithms, it has shown promising results and a lower error rate for sentiment quantification, as shown in Figure 2 for all four datasets. When the feature sets are combined their effectiveness is increased, which shows the usefulness of these features that contain meaningful information and outperformed other approaches when applied with SVM for all four datasets, as shown in Figure 3.

Figure 2. Comparison of single feature set performances.

Figure 3. Comparison of combined feature sets performances.

4.4. Results of Deep Features

Some of the deep features are also exploited to find out their impact on sentiment quantification. Deep features, including GloVe, BoW, word2vec, and n -gram, are extracted from all four datasets, SemEval2016, SemEval2017, STS-Gold, and Sanders. The deep features are tested with deep learning approaches such as DBN, RNN, and CNN-LSTM. The deep learning approaches are chosen due to their scalability and efficiency. Deep learning approaches do not require feature engineering and are suitable to achieve desired results. The results suggest that RNN is the best approach when applied with GloVe, which had a lower error rate (KLD = 0.009) for SemEval2016 and SemEval2017 (KLD = 0.011), and lower error rate (KLD = 0.004) for STS-Gold and Sanders (KLD = 0.008) when applied with word2vec among other deep learning approaches, as shown in Table 10. Deep learning approaches outperformed the conventional and ensemble-based machine learning approaches due to their high efficacy.

Table 10. Sentiment quantification based on deep features.

Algorithm	Features	SemEval2016			SemEval2017			STS-Gold			Sanders		
		AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD	AE	RAE	KLD
DBN	GloVe	0.019	0.394	0.014	0.021	0.431	0.015	0.011	0.222	0.008	0.016	0.334	0.012
	Word2vec	0.020	0.423	0.015	0.024	0.496	0.018	0.014	0.296	0.011	0.019	0.403	0.014
	n -Gram	0.033	0.676	0.025	0.034	0.695	0.026	0.025	0.516	0.019	0.030	0.612	0.023
	BoW	0.036	0.717	0.027	0.034	0.686	0.026	0.025	0.503	0.019	0.030	0.601	0.023
CNN-LSTM	GloVe	0.014	0.298	0.011	0.016	0.345	0.012	0.006	0.122	0.004	0.011	0.241	0.009
	Word2vec	0.016	0.329	0.012	0.019	0.393	0.014	0.037	0.772	0.027	0.037	0.772	0.027
	n -Gram	0.030	0.602	0.023	0.031	0.624	0.023	0.021	0.435	0.016	0.026	0.536	0.020
	BoW	0.030	0.606	0.023	0.030	0.602	0.023	0.020	0.409	0.015	0.025	0.512	0.019
RNN	GloVe	0.012	0.256	0.009	0.015	0.308	0.011	0.037	0.772	0.027	0.037	0.772	0.027
	Word2vec	0.015	0.306	0.011	0.016	0.338	0.012	0.005	0.114	0.004	0.011	0.234	0.008
	n -Gram	0.029	0.587	0.022	0.029	0.599	0.022	0.046	0.935	0.035	0.046	0.935	0.035
	BoW	0.027	0.565	0.021	0.028	0.568	0.021	0.045	0.928	0.034	0.045	0.928	0.034

4.5. Comparison of Proposed Technique with Existing Techniques

The proposed framework is effective and has achieved the desired results of accuracy for sentiment quantification. The proposed framework outperformed the baseline

approaches for SemEval2016, SemEval2017, STS-Gold, and Sanders, as shown in Table 11. Parameter settings for optimizing machine learning algorithms are shown in Table 12. The settings for deep learning algorithms DBN, CNN-LSTM, and RNN are CNN_Layers = “3”, Activation_Function = “tanh”, MaxPooling = 3, parameter values for hidden_layers = (300,300,300), learning_rate = “adaptive”, alpha = 0.001, regularizes L2(overfitting) = (0.01), loss = “categorical_crossentropy”, and optimizer = “Rmsprop”.

Table 11. Comparison of proposed framework and baseline approaches.

Sr. No	Dataset	Proposed Method	Baseline	Reference
1	SemEval2016	KLD = 0.013	KLD = 0.034	[37]
2	SemEval2017	KLD = 0.012	KLD = 0.036	[6]
3	STS-Gold	AE = 0.007	AE = 0.008	[38]
4	Sanders	KLD = 0.010	KLD = 0.009	[39]

Table 12. Parameter settings for applied machine learning algorithms.

Algorithms	Values
NB	priors = None, var_smoothing = 1e-09
Decision Tree	(loss = “deviance”, learning_rate = 0.01, n_estimators = 100, subsample = 1.0, criterion = “friedman_mse”, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_depth = 3, min_impurity_decrease = 0.0, min_impurity_split = None, init = None, random_state = None, max_features = None, verbose = 0, max_leaf_nodes = None, warm_start = False, presort = “auto”, validation_fraction = 0.1, n_iter_no_change = None, tol = 0.0001)
SVM	C = 1.0, kernel = “rbf”, degree = 3, gamma = “auto_deprecated”, coef0 = 0.0, shrinking = True, probability = False, tol = 0.001, cache_size = 200, class_weight = None, verbose = False, max_iter = 1, decision_function_shape = “ovr”, random_state = None
AdaBoost	(base_estimator = None, n_estimators = 10, max_samples = 1.0, max_features = 1.0, bootstrap = True, bootstrap_features = False, oob_score = False, warm_start = False, n_jobs = None, random_state = None,
RF	(n_estimators = 10, criterion = “gini”, max_depth = None, min_samples_split = 2, min_samples_leaf = 1, min_weight_fraction_leaf = 0.0, max_features = “auto”, max_leaf_nodes = None, min_impurity_decrease = 0.0, min_impurity_split = None, bootstrap = True, oob_score = False, n_jobs = None, random_state = None, verbose = 0, warm_start = False, class_weight = None)

5. Conclusions

This study contributes to the field of quantification based on sentiment analysis. The study exploits the diverse feature sets and explores the performance of machine learning approaches for the quantification of tweets. The proposed feature sets, such as POS, tweet specific, and sentiment- and content-based, increase the performance of classifiers. When the proposed feature sets are combined, they demonstrate efficient results in terms of quantification accuracy.

Three conventional machine learning approaches, namely Naïve Bayes (NB), Decision Tree (DT), and Support Vector Machine (SVM), are used in the proposed framework. AdaBoost and Random Forest are used in the case of ensemble-based approaches. Recurrent Neural Network (RNN), Deep Belief Network (DBN), and Convolutional Neural Network (CNN-LSTM) are exploited in the deep learning category of approaches. Ensemble approach AdaBoost dominated the other classifiers when applied using a single feature set, in terms of a lower error rate (KLD = 0.0213) for SemEval2016, SemEval2017

(KLD = 0.0214), STS-Gold (KLD = 0.0129), and Sanders (KLD = 0.0169). When the feature sets are combined, SVM has more promising results with a lower error rate for all four datasets, with KLD = 0.014 for SemEval2016, 0.013 for SemEval2017, 0.0051 for STS-Gold, and 0.0092 for Sanders. The computed results show that RNN with GloVe performed best for SemEval2016 and SemEval2017, and RNN with word2vec performed best for STS-Gold and Sanders.

Future work directions are as follows:

- As the social web channels provide a facility to add multilingual content, it raises diverse research issues for natural language processing and context understanding. In the case of multilingual content, especially where the diversity in different languages' structure presents issues such as sentence structure, stemming, parsing, tagging, etc., more research is needed.
- Each language has its own syntax and vocabulary. Text-based features of each language provide different research challenges. Therefore, applying the proposed features and algorithms on languages such as Arabic, Persian, and Urdu will be an interesting research work, as these languages are written from right to left.
- The analysis and learning carried out using one language can be applied to another language using cross-lingual analysis. Thus, the cross-lingual sentiment quantification task can also be a potential research area, especially in languages that lack annotated datasets.

Author Contributions: Conceptualization, K.A.; Formal analysis, K.A. and S.I.; Funding acquisition, F.K.A. and N.A.; Investigation, E.U.M.; Methodology, K.A.; Resources, F.K.A. and N.A.; Software, K.A.; Supervision, M.W.N. and E.U.M.; Validation, M.W.N.; Writing—original draft, K.A.; Writing—review & editing, S.I., F.K.A. and N.A. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia, for funding this work through Research Group no. RG-21-51-01.

Data Availability Statement: All the data used in this research study is publicly available for download and use for any research purpose.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Zamir, A.; Khan, H.U.; Mehmood, W.; Iqbal, T.; Akram, A.U. A feature-centric spam email detection model using diverse supervised machine learning algorithms. *Electron. Libr.* **2020**, *38*, 633–657. [\[CrossRef\]](#)
2. Mahmood, A.; Khan, H.U.; Ramzan, M.J. On Modelling for Bias-Aware Sentiment Analysis and Its Impact in Twitter. *J. Web Eng.* **2020**, *1*–28, 21–28.
3. Jabreel, M.; Moreno, A.J.A.S. A deep learning-based approach for multi-label emotion classification in tweets. *Appl. Sci.* **2019**, *9*, 1123. [\[CrossRef\]](#)
4. Chen, C.Y.-H.; Hafner, C.M.J. Sentiment-induced bubbles in the cryptocurrency market. *J. Risk Insur.* **2019**, *12*, 53. [\[CrossRef\]](#)
5. Jungherr, A.; Schoen, H.; Posegga, O.; Jürgens, P. Digital trace data in the study of public opinion: An indicator of attention toward politics rather than political support. *Soc. Sci. Comput. Rev.* **2017**, *35*, 336–356. [\[CrossRef\]](#)
6. Rosenthal, S.; Farra, N.; Nakov, P. SemEval-2017 task 4: Sentiment analysis in Twitter. In Proceedings of the Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), Vancouver, BC, Canada, 4 August 2017; pp. 502–518.
7. Gao, W.; Sebastiani, F. From classification to quantification in tweet sentiment analysis. *Soc. Net. Anal. Min.* **2016**, *6*, 1–22. [\[CrossRef\]](#)
8. Moradi-Jamei, B.; Shakeri, H.; Poggi-Corradini, P.; Higgins, M.J. A new method for quantifying network cyclic structure to improve community detection. *Physica A* **2021**, *561*, 125116. [\[CrossRef\]](#)
9. Esuli, A.; Moreo, A.; Sebastiani, F. Cross-lingual sentiment quantification. *IEEE Intell. Syst.* **2020**, *35*, 106–114. [\[CrossRef\]](#)
10. Faryal, M.; Iqbal, M.; Tahreem, H. Mental health diseases analysis on Twitter using machine learning. *IKSP J. Comput. Sci. Eng.* **2021**, *1*, 16–25.
11. Samuel, J.; Ali, G.; Rahman, M.; Esawi, E.; Samuel, Y. COVID-19 public sentiment insights and machine learning for tweets classification. *Information* **2020**, *11*, 314. [\[CrossRef\]](#)

12. Hassan, W.; Maletzke, A.; Batista, G. Accurately quantifying a billion instances per second. In Proceedings of the 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), Sydney, Australia, 6 October 2020; pp. 1–10.
13. Da San Martino, G.; Gao, W.; Sebastiani, F. Ordinal text quantification. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, Pisa, Italy, 7 July 2016; pp. 937–940.
14. Solyman, A.; Zhenyu, W.; Qian, T.; Elhag, A.A.M.; Rui, Z.; Mahmoud, Z. Automatic Arabic Grammatical Error Correction based on Expectation Maximization routing and target-bidirectional agreement. *Know.-Based Syst.* **2022**, *241*, 108180. [\[CrossRef\]](#)
15. Alzanin, S.M.; Azmi, A.M. Rumor detection in Arabic tweets using semi-supervised and unsupervised expectation–maximization. *Know.-Based Syst.* **2019**, *185*, 104945. [\[CrossRef\]](#)
16. Daughton, A.R.; Paul, M. A bootstrapping approach to social media quantification. *Soc. Net. Anal. Min.* **2021**, *11*, 1–14. [\[CrossRef\]](#)
17. Esuli, A.; Moreo Fernández, A.; Sebastiani, F. A recurrent neural network for sentiment quantification. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, 17 October 2018; pp. 1775–1778.
18. González-Castro, V.; Alaiz-Rodríguez, R.; Alegre, E. Class distribution estimation based on the Hellinger distance. *Inf. Sci.* **2013**, *218*, 146–164. [\[CrossRef\]](#)
19. Hopkins, D.J.; King, G.J.A.J.o.P.S. A method of automated nonparametric content analysis for social science. *Am. J. Political Sci.* **2010**, *54*, 229–247. [\[CrossRef\]](#)
20. Pérez-Gállego, P.; Castano, A.; Quevedo, J.R.; del Coz, J. Dynamic ensemble selection for quantification tasks. *Inf. Fus.* **2019**, *45*, 1–15. [\[CrossRef\]](#)
21. Pérez-Gállego, P.; Quevedo, J.R.; del Coz, J. Using ensembles for problems with characterizable changes in data distribution: A case study on quantification. *Inf. Fus.* **2017**, *34*, 87–100. [\[CrossRef\]](#)
22. Dias, F.F.; Ponti, M.A.; Minghim, R. A classification and quantification approach to generate features in soundscape ecology using neural networks. *Neur. Comput. Appl.* **2021**, *34*, 1–15. [\[CrossRef\]](#)
23. Adarsh, R.; Patil, A.; Rayar, S.; Veena, K. Comparison of VADER and LSTM for sentiment analysis. *Int. J. Recent Technol. Eng.* **2019**, *7*, 540–543.
24. Alabrah, A.; Alawadh, H.M.; Okon, O.D.; Meraj, T.; Rauf, H.T. Gulf Countries’ Citizens’ Acceptance of COVID-19 Vaccines—A Machine Learning Approach. *Mathematics* **2022**, *10*, 467. [\[CrossRef\]](#)
25. Khan, H.U. Mixed-sentiment classification of web forum posts using lexical and non-lexical features. *J. Web Eng.* **2017**, *16*, 161–176.
26. Khan, H.U.; Daud, A. Using Machine Learning Techniques for Subjectivity Analysis based on Lexical and Nonlexical Features. *J. Web Eng.* **2017**, *14*, 481–487.
27. Almanaseer, W.; Alshraideh, M.; Alkadi, O. A deep belief network classification approach for automatic diacritization of arabic text. *Appl. Sci.* **2021**, *11*, 5228. [\[CrossRef\]](#)
28. Elzayady, H.; Badran, K.M.; Salama, G.I. Arabic Opinion Mining Using Combined CNN-LSTM Models. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 25–36. [\[CrossRef\]](#)
29. Nemes, L.; Kiss, A.J. Social media sentiment analysis based on COVID-19. *J. Inf. Syst. Telecommun.* **2021**, *5*, 1–15. [\[CrossRef\]](#)
30. Zeng, J.; Liu, T.; Jia, W.; Zhou, J. Relation construction for aspect-level sentiment classification. *Inf. Sci.* **2022**, *586*, 209–223. [\[CrossRef\]](#)
31. Wu, C.; Xiong, Q.; Yi, H.; Yu, Y.; Zhu, Q.; Gao, M.; Chen, J. Multiple-element joint detection for Aspect-Based Sentiment Analysis. *Knowl.-Based Syst.* **2021**, *223*, 107073. [\[CrossRef\]](#)
32. Pathak, A.R.; Pandey, M.; Rautaray, S. Topic-level sentiment analysis of social media data using deep learning. *Appl. Soft Comput.* **2021**, *108*, 107440. [\[CrossRef\]](#)
33. Hamraoui, I.; Boubaker, A. Impact of Twitter sentiment on stock price returns. *Soc. Net. Anal. Min.* **2022**, *12*, 1–15. [\[CrossRef\]](#)
34. Saif, H.; Fernandez, M.; He, Y.; Alani, H. Evaluation datasets for Twitter sentiment analysis: A survey and a new dataset, the STS-Gold. In Proceedings of the 1st International Workshop on Emotion and Sentiment in Social and Expressive Media: Approaches and Perspectives from AI (ESSEM 2013), Turin, Italy, 3 December 2013.
35. Wang, D.; Al-Rubaie, A.; Hirsch, B.; Pole, G.C. National happiness index monitoring using Twitter for bilanguages. *Soc. Net. Anal. Min.* **2021**, *11*, 1–18. [\[CrossRef\]](#)
36. Deitrick, W.; Hu, W. Mutually enhancing community detection and sentiment analysis on twitter networks. *J. Data Anal. Inf. Proc.* **2013**, *1*, 19–29. [\[CrossRef\]](#)
37. Nakov, P.; Ritter, A.; Rosenthal, S.; Sebastiani, F.; Stoyanov, V. SemEval-2016 task 4: Sentiment analysis in Twitter. *arXiv* **2019**, arXiv:1912.00741.
38. Ayyub, K.; Iqbal, S.; Munir, E.U.; Nisar, M.W.; Abbasi, M. Exploring diverse features for sentiment quantification using machine learning algorithms. *IEEE Access* **2020**, *8*, 142819–142831. [\[CrossRef\]](#)
39. Labille, K.; Gauch, S. Optimizing Statistical Distance Measures in Multivariate SVM for Sentiment Quantification. In Proceedings of the Thirteenth International Conference on Information, Process, and Knowledge Management, Nice, France, 18–22 July 2021; pp. 57–64.