symmetry

MDPI

Article

Contextually Enriched Meta-Learning Ensemble Model for Urdu Sentiment Analysis

Kanwal Ahmed [1], Muhammad Imran Nadeem [1], Dun Li [1], Zhiyun Zheng [1,*], Nouf Al-Kahtani [2], Hend Khalid Alkahtani [3], Samih M. Mostafa [4] and Orken Mamyrbayev [5,*]

1 School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China
2 Department of Health Information Management and Technology, College of Public Health, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia
3 Department of Information Systems, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia
4 Computer Science Department, Faculty of Computers and Information, South Valley University, Qena 83523, Egypt
5 Institute of Information and Computational Technologies, Almaty 050010, Kazakhstan
* Correspondence: iezyzheng@zzu.edu.cn (Z.Z.); morkenj@mail.ru (O.M.)

Abstract: The task of analyzing sentiment has been extensively researched for a variety of languages. However, due to a dearth of readily available Natural Language Processing methods, Urdu sentiment analysis still necessitates additional study by academics. When it comes to text processing, Urdu has a lot to offer because of its rich morphological structure. The most difficult aspect is determining the optimal classifier. Several studies have incorporated ensemble learning into their methodology to boost performance by decreasing error rates and preventing overfitting. However, the baseline classifiers and the fusion procedure limit the performance of the ensemble approaches. This research made several contributions to incorporate the symmetries concept into the deep learning model and architecture: firstly, it presents a new meta-learning ensemble method for fusing basic machine learning and deep learning models utilizing two tiers of meta-classifiers for Urdu. The proposed ensemble technique combines the predictions of both the inter- and intra-committee classifiers on two separate levels. Secondly, a comparison is made between the performance of various committees of deep baseline classifiers and the performance of the suggested ensemble Model. Finally, the study's findings are expanded upon by contrasting the proposed ensemble approach efficiency with that of other, more advanced ensemble techniques. Additionally, the proposed model reduces complexity, and overfitting in the training process. The results show that the classification accuracy of the baseline deep models is greatly enhanced by the proposed MLE approach.

Keywords: sentiment analysis; Urdu sentiment analysis (USA); machine learning; deep learning; natural language processing; meta-learning ensemble (MLE)

check for updates

Citation: Ahmed, K.; Nadeem, M.I.; Li, D.; Zheng, Z.; Al-Kahtani, N.; Alkahtani, H.K.; Mostafa, S.M.; Mamyrbayev, O. Contextually Enriched Meta-Learning Ensemble Model for Urdu Sentiment Analysis *Symmetry* **2023**, *15*, 645. https://doi.org/10.3390/sym15030645

Academic Editors: José Carlos R. Alcantud and Silvio Pardi

Received: 17 November 2022
Revised: 16 January 2023
Accepted: 22 February 2023
Published: 3 March 2023

Copyright: © 2023 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 emergence of social media platforms has enabled the dissemination of information and perspectives on numerous global topics. Internet users share ideas, information, and sentiments about products, events, services, and political topics. People can debate a wide variety of topics, issues, and challenges on social media communication platforms like Facebook, Twitter, Instagram, and YouTube, and they can articulate themselves in a different way, including text, photographs, and videos. Such a plethora of freely accessible data has led to the creation of intelligent sentiment analysis technologies to help corporations, institutions, and organizations make better decisions [1]. The proposed algorithm that we developed serves a variety of functions, all of which will be discussed in further depth in the following sections.

To date, most research has been carried out on English language for sentiment analysis (SA) [2]. However, since the rise of social media, people are more likely to talk about how they feel in their own languages. So, SA needs to be used in other languages as well, so that important information that might be presented in different languages and ways does not get missed. Pakistan's official national language is Urdu, and it is more widely spoken in South Asia [3–9]. Urdu is spoken by more than 100 million people worldwide. It is an Indo-Aryan language. It employs the segmental writing technique and Arabic script in cursive format (Nastaliq style). Urdu's advanced vocabulary is derived from Persian and Arabic, while its everyday vocabulary is derived from the native languages of South Asia [10]. Urdu lacks capitalization, making it harder to recognize proper nouns, titles, acronyms, and abbreviations. Similar to Arabic and Persian, vowels are rare and optional in written Chinese. [11] Thus, words are frequently predicted using context. Urdu is a language with a free word order (Subject Object Verb) [12]. The boundary between words is not always distinguishable such as (she is very beautiful) وہ بہت خوبصورت ہے is understandable, although it has no space between words. Word order in Urdu sentence may be different, but the meaning would remain the same such as ام میٹھے ہیں and میٹھے ام ہیں have the same sense.

Urdu is based on a system called "abjad." This method dictates that long vowels and consonants must be written, whereas diacritics (short vowels) are optional in the Urdu language. It is a language that can be read in both directions, with left-to-right numbered sequence and characters are written in the opposite direction of the text, going from right to left. When the letters of a word are put together, they take on different shapes based on what the word means. A character can have up to four different shapes, which are called initial, medial, final, and isolated. There are numerous barriers that make SA of the Urdu language difficult, such as the fact that Urdu has both formal and informal verb forms, as well as masculine and feminine genders for each noun. Because of the syntactic and morphological peculiarities of Urdu, the difficulty of executing SA in that language has not been investigated to a great extent. Sentiment analysis implementations in Urdu are limited due to the following issues.

- A disregard for the situation

The resources currently available on the internet are predominantly written in widely spoken languages such as English, Spanish, Chinese, and others. As a result, these widely used languages have emerged as the most important topic of study over the past few decades. In addition, the inherent characteristics of Urdu have contributed to the delay in people's interest in studying its script, which has hampered the progress that has been made.

- Distinctions from a variety of other languages

Due to the various inherent distinctions that exist among Urdu and other languages, the SA methodologies that are now in use are not suitable for Urdu. For example, there is no capitalization, no grammatical or morphological qualities, and the word order is completely arbitrary.

There have only been a handful of research studies conducted on the use of the Urdu language to undertake sentiment analysis. Despite its widespread use, Urdu sentiment analysis has yet to be thoroughly investigated; the majority of existing literature studies are focused on different aspects of language processing [13,14]. This is because those in charge of the restoration of the Urdu language have shown little enthusiasm in the development, and there are not enough linguistic resources available. Various Deep Learning(DL) techniques have been successfully used for different natural language processing (NLP) tasks [15–17]. A lot of research has been conducted on SA [4–9], where information only gathered from written text. This text could be a review of a movie, a tweet, or an update or comment on Facebook. The context has a considerable impact on the meaning of a phrase; for instance, sardonic and other forms of mocking language are difficult to identify. Awais and Shoaib [18] investigated the use of several techniques used for sentiment analysis for other languages including English and reported that these techniques fail to replicate the

same results for Urdu. Recent research has highlighted the importance of conducting a thorough investigation into machine and deep learning-based methods for Urdu SA [19]. Our primary goal in carrying out this research is to evaluate the efficacy of meta learning ensemble models (MLE) for Urdu language as no previous study have tested its effectiveness for the task of sentiment analysis in Urdu language. The contribution of this research are as follows:

i. A novel Urdu SA mechanism is proposed that systematically combines ML&DL baseline models with 2-tiers of shallow meta-learners to produce an ensemble of models.
ii. For the purpose of text classification, we train many deep learning models utilizing public benchmark datasets of varying network architectures.
iii. Experiments are run to compare the proposed ensemble technique to single deep learning models.
iv. We further the experiments by contrasting the suggested ensemble method's results with those of other, more conventional ensemble methods.
v. We look into how the different kinds of predictions made by deep learning models affect the suggested ensemble approach.
vi. We investigated the efficacy of MLE model in low-resource languages like Urdu because, to the best of our knowledge, no prior study demonstrates its application for Urdu sentiment analysis.

This paper continues in the following way: Section 2 presents relevant literature. Section 3 explains Urdu SA's methodology and framework. Section 4 describes the experimental dataset and describes the outcomes. Section 5 summarizes and suggests further research subjects.

2. Literature Review

Sentiment analysis and the use of ensemble learning for classification purposes using either machine or deep learning are highlighted here. Additionally, a basic overview of sentiment analysis in Urdu will be provided.

2.1. Methods for Sentiment Analysis

Many prior studies of sentiment analysis relied on supervised machine learning techniques [20]. Support Vector Machine (SVM), Latent Dirichlet Allocation (LDA), Naive Bayes (NB), and artificial neural networks (ANN) are utilized to determine user sentiment from text [21,22]. Supervised methods are time-consuming and demanding of enormous amounts of training data. Unsupervised lexicon-based approaches were proposed [23,24] and are easy to implement and can be scaled quickly and easily. They rely extensively on the lexicon, making them less accurate [24,25]. Domain dependency makes lexicon-based approaches less suitable to domains without specialized lexicons.

Few researchers integrated supervised and lexicon-based approaches [26,27]. Zhang et al. [28] suggested a two-stage method for entity-level SA of tweets, with the first phase being a lexicon-based algorithm with high precision. A combination of lexicon-based and machine learning techniques has also been proposed, by Mudians et al. [29]. When compared to lexicon-based approaches, their strategy outperformed its competitors in detecting polarity and the intensity of sentiment and provided more precise justification and explanation than statistical methods. Ghiassi and Lee [30] recently suggested a new hybrid method for sentiment categorization by discovering and reducing a Twitter-specific vocabulary set. Chikersal et al. [31] presented a mix of ML and lexicon-based sentiment polarity approaches. In classifying user reviews, the hybrid strategy outperformed statistics and lexicon-based methods. Many studies have proposed using ensemble learning for sentiment analysis. Ensemble learning methods outperformed baseline classifiers. Research [32,33] used ensemble bagging to classify sentiment. Another study [34] suggested two ensemble approaches for sentiment analysis classification: majority voting and stacking. In [35], researchers

employed bagging and boosting to analyze tweet sentiment. KNN, SVM, and logistic regression were applied to sentiment140 [36].

Most recent DNN-based sentiment analysis studies have focused on word embedding or using DNNs for classification or clustering. Word embeddings capture similarity and lexical links [37]. Few researchers presented sentiment-aware word vectors to convey context. Large sentiment lexicons and supervised algorithms [38–40] are used to build these vectors. CNNs extract local features for sentiment analysis. These models are helpful when local patterns like n-grams are important in a long text. Many different languages and dialects use deep learning classification methods to categorize texts. The Russian encoder with transformers was used by Smetanin and Komarov [41]. CNN models to Roman, Spanish, and English have only been utilized in a small number of studies [42,43]. Arab Egyptian, Chinese, Emirati dialects, and Bengali have all had LSTM models applied to them [44,45].

Recent deep learning work has incorporated ensemble learning to deep learning classifiers. Ensemble approaches have improved deep learning performance in several disciplines. Akhtar et al. [46] suggested a multi-task ensemble architecture for sentiment, emotion, and intensity prediction. They used voting and stacking on LSTM, CNN, and GRU. Heikal et al. [47] implemented voting ensemble to CNN and LSTM model outputs using ASTD [48]. Minaee et al. [49] presented a voting ensemble by averaging CNN and LSTM predictions on IMDB reviews [49,50].

2.2. URDU Sentiment Analysis

Some research studies have been conducted in the domain of Urdu sentiment analysis. SentiUnits [51] were generated by detecting sentiment words/sentences. Along the word, this comprised orthographic, phonological, syntactic, and morphological aspects. The polarity of the SentiUnits was added to compute the polarity of the sentence. Later in the work Syed et al. [52], used shallow parsing chunking to correlate SentiUnits with their objectives. Nominal assessment head words and modifiers were added to the lexicon. By locating all of the intended SentiUnits, the polarity of the sentences may be calculated. syed et al. [53] presented SenitUnits in JSON format and a two-step categorization method. They included context-dependent phrases, intensifiers, and verbs for lexicon-based SA. Lexicon-based approaches for sentiment categorization beat supervised machine learning [54].

As sentiment carrier words, they used adjectives, verbs, and nouns, as well as negation intensifiers and context dependant words. SentiUrduNet is an Urdu sentiment lexicon collection established by Asghar et al. [5] through the process of translating English opinion expressions into their Urdu equivalents. In a similar manner, Urdu language modifiers were rendered into their English equivalents in order to compute emotion scores. For those phrases for which the sentiment score was either absent or incorrect, manual scoring was carried out. Hassan and Shoaib [55] presented a SEGMODEL that studied how the mood of a sentence changed depending on whether a sub-opinion was considered. After adding up the polarities of each clause in the sentence, we were able to determine the overall polarity of the statement. According to the results of their tests, their method performed far better than the BOW method, with an accuracy of 75.8.

Mukhtar et al. [56] demonstrated that the top classifiers for classifying Urdu text sentiment were Lib SVM, J48, and IBK. For sentence level SA, Mukhtar et al. used supervised ML algorithms in [57]. They used SVM, KNN, and decision trees to extract 154 features, such as positive, negative, neutral, intensifier, and negation to improve classification. In terms of accuracy, the authors reported that KNN outperformed the other classifiers. Awais and Shoaib [18] used conversation information to identify sub-opinions, that they subsequently fed into supervised and rule-based techniques. According to their findings, the rule-based classifier outperformed BOW, and the ML model containing discourse features outperformed the one without. When it comes to classification of sentiments, ML approaches perform better than rule-based methods when training data is available. Nasim

and Ghani [58] used Markov Chains to classify tweets as neutral, positive or negative. Their methodology was more accurate than lexicon-based and other Machine Learning methods.

Previous research [59–61], shows that lexicon-based sentiment classification performed better than ML approaches for Urdu text sentiment analysis, although its construction requires many human operations and is highly dependent on vocabulary size. Domain-specific criteria are designed for better classification, however social media data deviates greatly from linguistic standards, making them ineffective. On the other hand, machine learning algorithms demand labeled data but lack the necessary domain knowledge. It would be possible to make this better by including features such as positive words, negative words, and negation.

Based on a comprehensive literature review, we note the following gaps in the current body of knowledge. It should be noted that Urdu is a language with few publicly available corpora and lexicons. It has morphological complexity, which makes SA for Urdu more difficult. According to [62], very few studies in the Information Retrieval (IR) field have focused on Urdu stemming challenges, and many of the strategies developed for SA in other languages are not applicable to Urdu, due to its complex morphology [10–12]. There is a need for further research into concept-level sentiment analysis that considers communication context. Almost all of the previous research on Urdu employed either a lexicon-based approach or machine learning to determine the polarity of a sentence. Due to the absence of a good corpus and stable vocabulary, it is required to employ a hybrid strategy that combines the characteristics of a lexicon with the supervised and unsupervised machine learning approaches [63]. Earlier research noted the differences in performance between Roman Urdu datasets and Urdu corpora, owing to the morphological complexity of the latter [64]. The majority of earlier work involving the verification of Urdu phrases was accomplished using a lexicon-based method. Therefore, we attempted to build a system that combines machine and deep learning techniques to get more precise findings. No previous research in Urdu attempted to extract contextual semantic for the task of sentiment analysis. Although DL approaches have not been thoroughly researched for use with Urdu text, we chose to use them because they have been shown to be effective in sentiment categorization. Keeping in view the success of ensemble methods in improving the accuracy of baseline machine learning models [65,66], we consider it for our research as no previous research in Urdu SA explored Ensemble methods [56,61]. We have used deep models [61,67–70] as baseline for meta-learning and ML models as shallow meta classifier due to their proven supremacy for Urdu language [18,56–58]. It's essential to mention that the ensemble model's prediction power is limited by the dataset's size and the performance of the baseline classifiers.

3. Methodology

In this section, a Meta learning ensemble model is proposed for Urdu Sentiment Analysis. Five baseline classifiers are trained in tier-0 of 3-tier meta learning architectures. Committees are formed in tier-0. The output from tier-0 is fed intro tier-1. Output of tier-1 is used as input to tier-2 where final predictions are made. By utilizing the idea of meta-classifiers or meta-learners, the proposed ensemble technique combines the predictions of both the inter- and intra-committee classifiers on two separate levels. By utilizing a shallow meta-classifier, committee members can combine their individual baseline classifiers to form a single, more accurate classification. In order to generate meta-classifiers or Tier-1 models, learning algorithms are implemented across all committees. These models attempt to foretell how meta-data should be generated by combining the results of committee-level baseline classifiers (Tier-0 models). In addition, a Tier-2 model, also known as a meta-learner, is created by combining the outputs of the Tier-1 models with a state-of-the-art learning algorithm. We begin with a discussion of the processes involved in cleaning the data and standardizing the text. Later, we discuss the architecture of proposed ensemble model. Last, we discuss the ML/DL models used in ensemble and their parameter values.

3.1. Preprocessing Data

Preprocessing is carried out to eliminate data inconsistencies. The first thing that is performed to clean up the data is to get rid of any characters that are not related to Urdu, such as punctuation marks, digits, alphabetic characters, and characters from other languages. The next step is to eliminate inconsequential and undesired data items such as stop words. There is a list that contains 254 stop words of Urdu. On white spaces, text is tokenized. Research [71], described a method for segmenting Urdu words that is based on Conditional Random Fields (CRF). This method is used for word segmentation. The Assas-band stemmer [72] is used for stemming. Following the process of stemming, word tokens in the text that include fewer than two characters are deleted. Stemming consists of reducing a given word to its stem, base, or root, e.g., the stem of دردمند ("dardmand", "sorrowful") is درد ("dard", "pain").

3.2. Normalization

Social media and user-generated content text is analyzed and used for decision-making. This textual data often uses informal language because users can express their opinions without using basic grammar and lexical rules. NLP analysis tools help convert such texts into more advanced grammar. As per Wikipedia, "text normalization is the particular type of process in which text is transferred into a single canonical form that it might not have had before". Normalization is performed prior to applying a model to ensure that the text is consistent before processing. Alam and ul Hussain [73] presented findings that included a normalization process for the Roman Urdu and Urdu that was built independently using tokenization and the frequency of each word. In their study, Khan and Malik [74] defined normalization as follows: all string attributes were transformed into a set of attributes depending on the word tokenizer using the StringToWordVector prior to the classification phase. Attributes may be gleaned from the training data. Good, bad, positive, and negative categories were used to classify feelings. Prior to testing, a classifier must be trained using the training corpus's rules.

Normalizing Urdu text helps with NLP tasks. This step corrects Urdu encoding. Normalization is performed to get all Urdu unicode characters (0600-06FF). This step prevents Urdu word concatenation. خوشبخت is a unigram with two strings. Syntactically and semantically, khush and hal are the same word. If the space between two strings is missing, we get خوش بخت, which is improper Urdu. Normalization reduces this effect. This task is performed by utilizing UrduHack .

3.3. N-Gram Model

Shannon first proposed the information theoretic concept of N-grams in 1948. [75]. N-grams, as defined by [76], are word sequence within a document with a fixed window size of N. N-grams provide insight into the corpus that can be put to various uses [77,78]. In addition, we used N-grams based on the characters in a text in this study. The values for N could be between 2 and 10. We use N-Gram with ML model NBSVM.

3.4. Pre-Trained Word Embedding

In numerous recent NLP tasks, pre-trained word vector models have achieved state-of-the-art performance. These models have already been trained for a variety of purposes using massive data sets. FastText [69] is a word vector model that learns from the English Wikipedia and other popular crawl datasets. Included in the 157 languages used to train this model is Urdu. This is why we employ deep learning models trained with the fastText word embedding model for this job. For fastText, we used skip-gram and continuous bag of words (CBOW) for model training [79,80]. By decomposing the unigram (words) into bags of character n-grams (sub-words) and assigning a vector value to each character n-gram, the fastText model expands upon the skip-gram approach. As a result, we can represent any given word by adding together its n-gram vectors. For CNN we exploited

Word2Vec [61], to extract deep contextual representations. For Bi-GRU we used pretrained WORD2VEC(CONLL) embedding.

3.5. Proposed Ensemble Scheme

The proposed MLE method's primary focus is on combining five distinct base-classifiers into a hierarchy of ascending levels of accuracy. The so-called committees are formed in the initial tier, which is referred to as Tier-0, where training dataset is partitioned into distinct group to train the base learners. A set of shallow meta-classifiers are trained on the predictions made by the committees in Tier-0 prior to moving on to Tier-1. The findings are created in the final layer (Tier-2) by merging the predictions of the previous set of classifiers (Tier 2). For this strategy, the name "committee" serves as an analogy for a group of different classifiers working together. The number of classifiers on the committee is referred to as its size. Using the concept of meta-classifiers or meta-learners, the suggested ensemble technique combines inter-committee and intra-committee predictions at two layers. Inter-committee fusion involves employing a shallow meta-classifier to join the baseline classifiers within the committee. The idea behind intra-committees is to use a top-level meta-classifier to integrate the predictions of the several committees. To build meta-classifiers or Tier-1 models, each committee use learning techniques. Tier-0 models seek to foresee how the outputs of a committee's baseline classifiers should be integrated in order to generate meta-data. By integrating Tier-1 model results with a superior learning technique, a meta-learner or Tier-2 model can be constructed as well.

3.5.1. The Proposed Architecture

Figure 1 depicts the proposed ensemble's overall learning architecture. Each committee is trained individually utilizing distinctive training data and base models in the architecture's three distinct layers of classifiers. Once the baseline learners in each committee have finished their work, all committee outputs are combined using a top-level meta-classifier.

The suggested design has an input layer, a hidden layer, and an output layer represented as tier-0, tier-1, tier-2, respectively. This architecture is analogous to that of a multi-layer perceptron. Meta-classifiers in Tier 1 perform the role of activation functions, taking input from Tier 0 and producing output for Tier 2 using meta-classifiers.

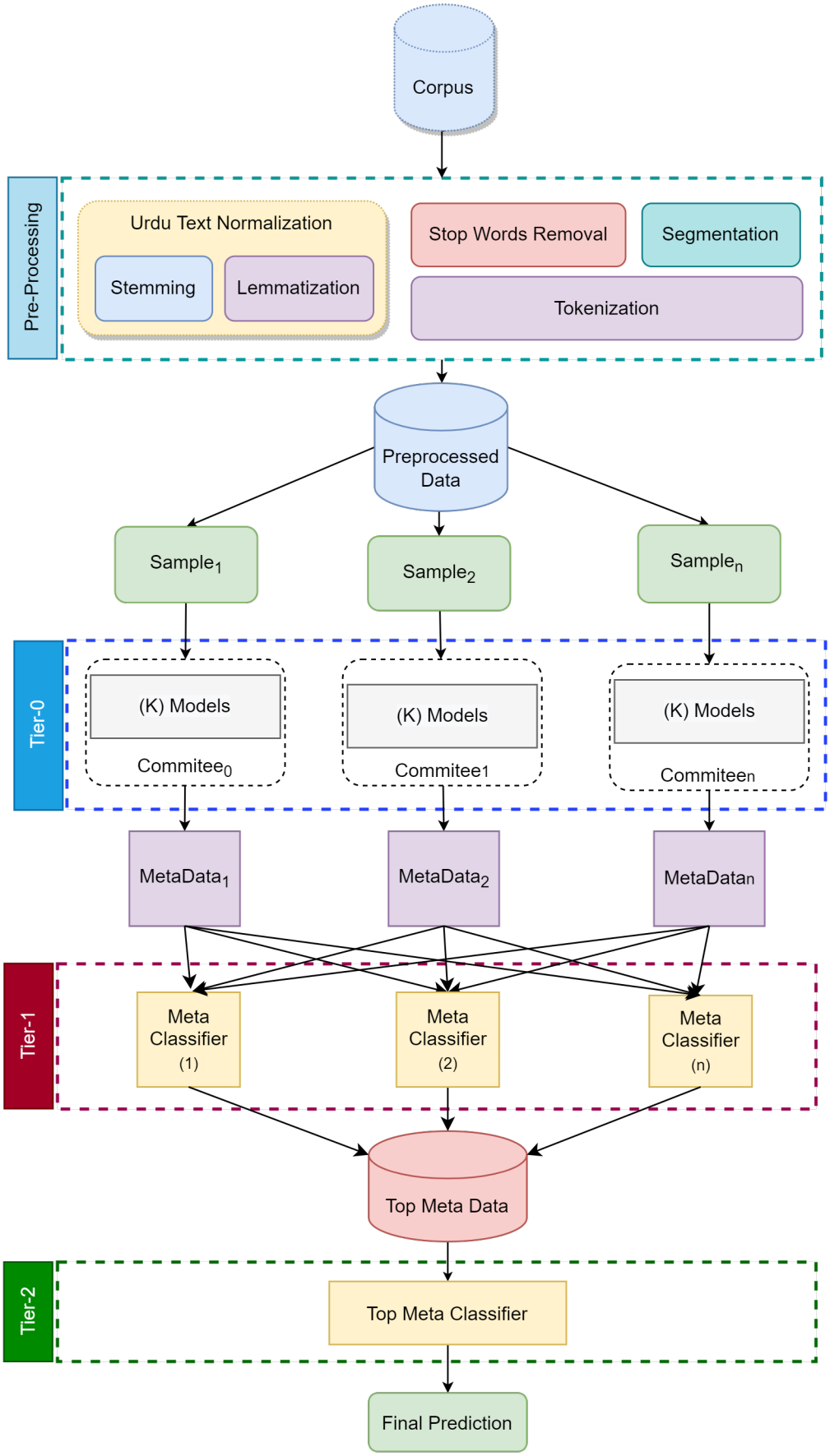

Figure 1. The proposed meta-learning ensemble model.

3.5.2. Formal Description

The algorithm of the proposed model is presented in Algorithm 1.

Algorithm 1 Proposed Training Algorithm

Require: Input data
Ensure: Sampling
1: $Da(0)$
2: $D_a^{(0)} = Train_a^{(0)} U\left(Test_a^{(0)} = \left(X_a^{(0)}, Y_a^{(0)}\right)\right), 1 \leq a \leq n$
3: Tier-0
4: $DL = DL1, DL2, \ldots, DLk$, a set of baseline Algorithm
5: For each $Train_a^{(0)}, 1 \leq a \leq n$
6: For each $DL_j \in DL, 1 \leq b \leq k$
7: $M_{ab} \leftarrow \left(DL_b,\ Train_a^{(0)}\right), 1 \leq b \leq k$
8: $G_a \leftarrow \{M_{a1},\ M_{a2},\ \ldots.., M_{ak}\},\ 1 \leq a \leq n$
9: For each $M_{ab} \in G_a,\ 1 \leq a \leq n,\ 1 \leq b \leq k$
10: $y_{ab}^{(1)} \leftarrow predictions\ of\ M_{ab}\left(X_a^{(0)}\right), 1 \leq b \leq k$
11: $Data_a^{(1)} \leftarrow stack\left(\left[y_{a1}^{(1)}, y_{a2}^{(1)}, \ldots.,\ y_{ak}^{(1)}, Y_a^{(0)}\right]\right), 1 \leq a \leq n$
12: Tier-1
13: $SplitD_a^{(1)} = Train_a^{(1)} U\left(Test_a^{(1)} = \left(X_a^{(1)}, Y_a^{(1)}\right)\right),\ 1 \leq a \leq n$
14: $F = f1, f2, \ldots.fn$ a set of n shallow classifiers
15: For each $Train_a^{(1)}, 1 \leq a \leq n$
16: $D_{fb} \leftarrow fit\left(f_a,\ Train_a^{(1)}\right), 1 \leq b \leq k$
17: $df \leftarrow (df_1, df_2, \ldots., df_n)$
18: TIER-2
19: For each $Test_a^{(1)} = \left(X_a^{(1)}, Y_a^{(1)}\right), 1 \leq a \leq n$
20: For each $df_a \in df$
21: $y_{ab}^{(2)} \leftarrow predictionofdf_b(X_a^{(1)})$
22: $Data_a^{(2)} \leftarrow stack\left[y_{a1}^{(2)}, y_{a2}^{(2)}, \ldots.., y_{an}^{(2)}, Y_a^{(2)}\right], 1 \leq a \leq n$
23: $FinalMetaData = stack([D1(2), D2(2), \ldots., Dn2]T)$
24: $Top \leftarrow a\ shallow\ classifier$
25: $Model \leftarrow fit(Top, FinalMetaData)$

3.5.3. Classification Using MLE

Figure 2 depicts the categorization operation carried out by the trained ensemble model on previously unseen data. When a fresh, previously unseen sample, denoted by the letter x, is introduced to the proposed ensemble technique, a copy of it is first sent across all the committees that make up Tier-0. After that, the committees construct a two-dimensional tensor that contains K columns and n rows of predictions. This tensor serves as the input to the Tier-1 models. These models make a forecast in the form of a vector with n length predictions. The final prediction will be generated using the Tier-2.

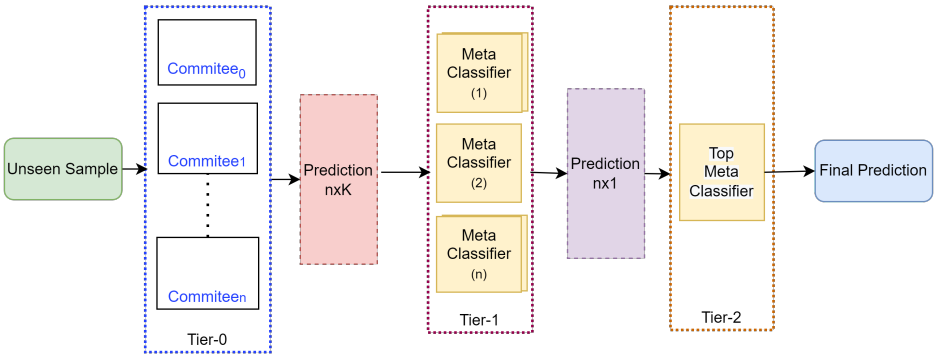

Figure 2. Classification of unseen data.

3.6. Generating Base Models for Tier-0

Textual input must be preprocessed before training baseline classifiers. Therefore, word embedding is utilized as the initial layer prior to training the network [61]. For each baseline model we have used embedding accordingly. To evaluate our suggested ensemble technique, we first need to develop a group of classifiers that represent the baseline models.

3.6.1. Naïve Bayes Support Vector Machines (NBSVM)

Combining the traditional support vector machine (SVM) with Bayesian probabilities is the premise of Wang and Manning's [67] model for classifying texts. In this model, word count attributes are substituted with the following Naive Bayes log-count ratios:

$$h = \log\left(\frac{\frac{x}{|x|}}{\frac{y}{|y|}}\right) \tag{1}$$

where, x and y represent word count vectors. These are used for the binary classification problem with the label $z(a) \in \{-1, 1\}$ as described below:

$$x = \alpha + \sum_{a:z^{(a)}=1} f^{(a)} \tag{2}$$

$$y = \alpha + \sum_{a:z^{(a)}=-1} f^{(a)} \tag{3}$$

In this scenario, the collection of features is denoted by V, and the feature count vector for training sample and is represented by $f^{(a)} \in F \mid V$. It has been shown that this method is effective for a number of text classification jobs, and not just in terms of speed. In this study, we constructed a basic neural NBSVM by stacking two embedding layers on top of one another and adding a sigmoid activation layer. This model's input document consists of word IDs. This model trains more quickly than one that uses a term-document matrix because it makes use of a look-up strategy that is embedded inside the layer. The first embedding layer of the NBSVM model is responsible for storing the Naive Bayes log-count ratios. These ratios represent the probability that a given word would appear in a document belonging to one class as opposed to the other. The second layer is responsible for storing the learned coefficients for every word contained in the document. The conclusion that can be drawn from this model is represented by the simple dot product of these two vectors. The primary advantage is that its training process can be completed very quickly. We have made use of character N-grams, where N is between 2 and 10. From our corpus, we have retrieved character N-grams. Using a NBSVM classifier, we compare the effectiveness of a variety of values for N when applied to character N-grams. so, NBSVM benefits more from the 2-Gram features for Urdu datasets in terms of accuracy, precision, recall, and F1-score.

3.6.2. Convolutional Neural Network (CNN)

This section will examine the features that were extracted from the text. We propose utilizing deep contextual semantic characteristics in conjunction with other standard natural language processing features to determine the sentiment. Word2vec [61] generates word embeddings so comparable words share a vector space. Word2vec's 300-dimensional skip-gram model produces word vectors. Minimum dataset word count is 5, and window size is 20. Algorithms are designed to optimize average log probabilities given the set of training words $x_1, x_2, \ldots, x_T$.

$$E = \frac{1}{C} + \sum_{c=1}^{C} \sum_{-w \le j \le w, j \neq 0} \log p\left(x_{c+j} \mid x_c\right) \tag{4}$$

where w is the size of the window, and C signifies the size of the corpus. Using the $SoftMax$ function, the probability $p\left(x_{c+j} \mid x_c\right)$ may be defined as:

$$p(x_0 \mid x_I) = \frac{\exp\left(v'^{C_{vx_I}}_{x_0}\right)}{\sum_{x=1}^{x} \exp\left(v'^{C_{vx_I}}_{x_0}\right)} \tag{5}$$

Word2vec is effective in capturing the semantics of the words; nevertheless, it does not consider the order of words or the context of a word. Contextual information and semantics can be extracted using CNN. Word vectors that have been created using *word2vec* are input into the embedding layer of CNN. The embedding matrix M maps words to vectors.

$$M = \begin{bmatrix} j_{00} & j_{01} & j_{02} & j_{03} & \cdots, & j_{0i} & \cdots, & j_{0q} \\ j_{10} & j_{11} & j_{12} & j_{13} & \cdots, & j_{1i} & \cdots, & j_{1q} \\ \vdots & \vdots & \vdots & \vdots & \cdots, & \vdots & \cdots, & \vdots \\ j_{i0} & j_{i1} & j_{i2} & j_{i3} & \cdots, & j_{ii} & \cdots, & j_{iq} \\ \vdots & \vdots & \vdots & \vdots & \cdots, & \vdots & \vdots & \vdots \\ j_{p0} & j_{p1} & j_{p2} & j_{p3} & \cdots, & j_{pi} & \cdots, & j_{pq} \end{bmatrix}^{px}$$

where j_{pi} represents the embedding of the ith letter of the word p and q is the dimension of the vector. Let $ji \in Rq$ represent the q-dimensional word vector of j word i in a sentence. The following can be used to represent a sentence with n words:

$$j_{1:n} = j_1, \oplus j_2, \oplus j_3, \oplus \ldots\ldots, \oplus j_n \tag{6}$$

where $\oplus$ denotes the concatenation operator. Sentence embeddings $j_i, j_{i+1}, \ldots, j_{i+j}$ are represented as e. Convolving $X \in R_h q$ over h words generates new features. Convolution with $j_{i:i+h-1} h$ yields v_i as:

$$V = [v_1, v_2, \ldots, v_{n-h+1}] \tag{7}$$

where $b \in R$ is the bias term, X is the h-sized kernel, h and f is the rectified linear unit nonlinear activation function. To build a feature map, the kernel is convolved with each window of words $j_{1:h}, j_{2:h+1}, \ldots j_{n-h+1:n}$.

$$V = [v_1, v_2, \ldots, v_{n-h+1}] \tag{8}$$

where $V \in R^{n-h+1}$. Max pooling extracts the most important features from feature maps.

$$\hat{V} = [\max(v_1), \max(v_2), \ldots + \max(v_{n-h+1})] \tag{9}$$

A fully connected tanh layer translates salient features into k-dimensional vectors.

$$S_f = \tanh\left(X_f \cdot \hat{V} + b_f\right) \tag{10}$$

Sf signifies deep contextual semantic features, X denotes the fully connected layer's weights, and bf signifies the layer's bias term. Finally, classification is performed using the output layer, which consists of a dense layer with two SoftMax cells. Table 1 displays the CNN model's parameter settings.

Table 1. CNN hyperparameter values.

Parameter Name	Value
Number of filters	250
Embedding Dim	50
Max features	20,000
Drop out	0.2
Kernel size	3
Activation Function	Relu
Dense	250
Loss Function	Categorical Cross Entropy
Optimizer	Adam

3.6.3. Bidirectional Gated Recurrent Network (BiGRU)

To acquire Urdu embeddings for Bi-GRU, we consulted the NLPL word embeddings repository [10]. Using the Word2Vec Continuous Skipgram model, it provides 100-dimensional vectors trained on more than 108310 words from the CoNLL17 Urdu corpus. Input data shape, embedding matrix, and maximum sentence length are the parameters for the embedding layer. These embeddings are input into Bi-GRU model. To capture long dependencies in text, Bi-GRU is used [68]. A unique kind of GRU known as a bidirectional GRU (BiGRU) is one that can determine sequential relationships in both the forward and the backward directions. The BiGRU makes it possible for the model to consider both the previous and the subsequent contexts. This is an important characteristic since considering the context of sentiment words is a significant challenge in relation to applications of sentiment analysis [81]. A dropout layer follows the embedding layer in the BiGRU model, just like it does in the CNN model discussed in Section 3.6.2. The subsequent layer maps vocabulary indices to an embedding space.

On the other hand, this model makes use of pre-trained word vectors that were trained on "common crawl" and "Wikipedia" through the use of the fastText model [82]. The CBOW algorithm with position-weighting is used to train this word vector. After the dropout layer, the BiGRU layer extracts forward and backward contexts. This layer is made up of GRU cells, each of which makes use of two gates: an update gate r, which combines the forget and input gates found in ordinary Long Short-Term Memory cells (LSTMS), and a reset gate z. Together, these gates make up this layer. The following functions are the foundation on which the update and reset procedures are built:

$$r_t = \delta(W_r h_{t-1} + U_r x_t + b_r) \tag{11}$$

$$z_t = \delta(W_z h_{t-1} + U_z x_t + b_z) \tag{12}$$

Assuming that U and W are the weight matrices of gates, x_t and h_t are the input and hidden states, and b is the bias vector, then represents the logarithmic-sigmoid function. Cell output is computed utilizing input and hidden state. The following functions are used to determine the cell's hidden state:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{13}$$

$$\tilde{h}_t = \tanh\left(W_{\tilde{h}_t}(h_{t-1} \odot r_t) + U_{\tilde{h}_t} x_t\right) \tag{14}$$

As part of the BiGRU layer, two hidden layers are integrated to extract both the forward and backward contexts. As a result, there is a two-way flow of temporal information. The output of the BiGRU model is subjected to both a global max pooling and a global

average pooling layer in order to produce various feature maps. The output of pooling layers is then combined and routed to a fully connected layer. Table 2 depicts the BiGRU model's parameter configuration.

Table 2. BiGRU hyperparameter values.

Parameter Name	Value
Embedding Dim	300
Max features	20,000
Drop out	0.2
Number of Cells	80
Activation Function	Relu
Dense	2
Loss Function	Categorical Cross Entropy
Optimizer	Adam

3.6.4. FastText

To represent each word in the fasttext model, a bag of character n-grams is used. It allows embeddings to be constructed utilizing data at the subword level. The word "بہترین" can be divided into "بہتر" "بہت" "بہتری" and "بہترین". As a result, it is able to deal with words that are not present in the dataset/dictionary. With no space between words and a high number of compound nouns, the Urdu language benefits greatly from this.

The fastText model employed in this investigation is comparable to the original fastText model [69]. It contains the same embedding, spatial dropout, and global max-pooling layers as Sections 3.6.2 and 3.6.3. Additionally, the fastText model takes advantage of a batch normalization layer to speed up training and increase the model's performance. A representation of the text as a bag-of-words is what is fed into the fastText model as its input. After that, this representation is passed on to a lookup layer, which is responsible for computing the embeddings for every word. The subsequent phase involves averaging the word embeddings in order to arrive at a single embedding for the entire body of text. Max features x Embedding dim parameters are used in the hidden layer. Finally, a linear classifier using a SoftMax function is fed with the averaged vector. Table 3 displays the parameter settings that are used for the fastText model.

Table 3. FastText hyperparameter values.

Parameter Name	Value
Embedding Dim	64
Max features	20,000
Spatial Dropout	0.25
Activation Function	Relu
Dense 1	64
Dropout	0.5
Dense 2	2
Loss Function	Categorical Cross Entropy
Optimizer	Adam

3.6.5. DistilBERT Base Multilingual Model

This model is a simplified version of the BERT base multilingual model. The concatenation of Wikipedia in 104 distinct languages was used to train the model. There are six layers, 768 dimensions, and 12 heads in the model, which totals 134 M parameters (compared to 177 M parameters for mBERT-base), while DistilBERT(base M) is faster than mBERT-base, it is only by a factor of two. According to [70], "DistilBERT was trained on 8 16 GB V100 GPUs for roughly 90 h." This architecture considerably outperforms similar ones, such as the RoBERTa [83] model, which requires one day of training on 1024 32 GB V100, in terms of speed and memory requirements.

3.7. The Combiner Shallow Meta-Classifiers

We use a few different shallow meta-classifiers as top surface meta-learners in order to combine the baseline models that were trained within the committees. To be more specific, we made use of a collection of highly effective algorithms for shallow learning. These algorithms include Naive Bayes (NB) [84], Random Forest (RF) [85], Gradient Boosting (GB) [86], Logistic Regression (LR) [87], and Support Vector Machines (SVM) [88]. The predictions of committees can be combined in Tier-1 using any shallow classifier. In the literature we have seen that ML models performed best on Urdu Language. We have evaluated the proposed model with hard and soft predictions. Voting is typically used to average the predictions of baseline classifiers. Hard voting is used to determine the final prediction results, which are usually determined by a majority vote on the predictions of many classifiers. The mathematical definition of hard voting is Equation (15), which specifies the statistical mode of the classifiers' predictions.

$$y_i = mode(c_1, c_2, \ldots, c_k) \tag{15}$$

While hard voting is simple to implement and produces better results than baseline classifiers, it does not account for the probability of minor predicated classes. For instance, if we have three classifiers with prediction probabilities of (0.49, 0.48, and 63), hard voting will result in the probabilities being predicted as (0,0,1) In this case, the final hard vote prediction based on the votes of the three classifiers is 0. When the probabilities of classifiers are averaged, however, the weighted average becomes 0:526, which implies 1. Therefore, Soft voting takes into account the probabilities value of each classifier rather than its prediction labels. Using Equation (16), soft voting prediction can be formalized.

$$y = argmax_i \frac{1}{n} \sum_{j=1}^{n} w_{ij} \tag{16}$$

where w_{ij} represents the likelihood of the I_{th} class label for the j_{th} classifier. Voting is modified by weighting each classifier proportionally to its accuracy performance on a validation set [89].

4. Experiments and Results

In this section, we will discuss the experimental setup, datasets, evaluation metrics and results.

4.1. Experimental Setup

This section describes the conditions under which the proposed ensemble scheme will be evaluated. The MLE Model was implemented in Python using Keras with the TensorFlow backend. Python's scikit-learn library was used to implement ensemble learning.

4.2. Dataset

Due to lack of availability of larges annotated Urdu dataset, we have used some small dataset publicly available. In the machine and deep learning field, it is widely accepted that data is the most critical component of any task. For Urdu no standard dataset is currently large enough to be used for sentiment Analysis. Consequently, the data that was analyzed came from three different sources. The concat function in the pandas library was used to merge the above-mentioned datasets. There are a total of 28,921 reviews in the final consolidated dataset utilized in this study. This consolidated data will further be referred as UCD. Statistics of used datasets are presented in Table 4 below:

Table 4. Statistics of datasets being used for evaluation.

Dataset	Total Reviews	Classes
SAU-18 [90]	10,008	Pos,neg,neu
UCSA [91]	9601	Pos,neg
UCSA-21 [92]	9312	Pos,neg,neu
UCD	28,921	Pos,neg,neu

4.3. Evaluation Measures

We used several different algorithms, including NBSVM, fastText, CNN, BiGRU, and DistilBERT(base M), in addition to the suggested fusion model. Recall (R), Precision (P), F1-measure, and Accuracy (A) are used to assess the efficacy of our sentiment analysis models. The following are the mathematical Equations (17)–(20):

$$P = \frac{TP}{TP + FP} \tag{17}$$

$$R = \frac{TP}{TP + FN} \tag{18}$$

$$F1 - score = 2 \times \frac{PxR}{P + R} \tag{19}$$

$$A = \frac{TP + TN}{TP + FP + TN + FN} \tag{20}$$

4.4. Experimental Results and Comparative Analysis

In order to analyze the impact of the proposed ensemble approach on the predictions, we ran several tests on the preceding Urdu datasets to compare the performance of the ensemble to that of best individual baseline models. In addition, we analyze the proposed ensemble by making both hard and soft predictions based on the baseline models. A summary of comparisons with other common ensemble approaches, like stacking, bagging, and voting is also presented.

For each dataset, we used the Pareto principle to divide it into training and test sets, with an 80/20 split between the two [93]. A data partitioning approach is required to divide the training data fed into baseline classifiers in committees. The following partitioning approaches are technically considered to be part of ensemble methods [94]: fold partitioning, random selection of data with replacement, disjunct, and random-size sampling. In the course of our research, we utilized the disjunct partitioning approach, which involves arbitrarily slicing the training set up into k different partitions of equal size. When training each committee, a unique subset of the dataset is utilized as the resource. In the corpus, we chose 5 partitions. A committee of trained five baseline classifiers is formed for each split.

For Tier-1, LR is the most common optimal combiner of the committee predictions from baseline classifiers, according to 5-fold cross-validation. Nevertheless, several shallow top-meta classifiers do significantly better in the overall prediction. A comparison of baseline models is shown in Table 5.

As a starting point, we develop five separate models. All splits' average accuracy is a measure of robust cross-validation for the baseline models. On all datasets, Distilbert(M) outperformed other deep baseline models in terms of accuracy, precision, recall, and F1 score.

Table 5. The comparison of baseline classifiers.

Dataset	Classifiers	Accuracy	Precision	Recall	F1 Score
SAU-18	NBSVM	81.64	80.96	80.94	81.80
	CNN	81.75	81.40	81.48	81.62
	BiGRU	82.35	81.95	82.37	82.90
	DistilBert(M)	**84.98**	**84.56**	**84.40**	**84.65**
	fasttext	80.12	81.29	80.82	80.60
UCSA	NBSVM	77.82	77.26	77.84	77.85
	CNN	78.10	78.43	76.78	77.59
	BiGRU	80.55	80.05	80.15	80.09
	DistilBert(M)	**82.50**	**81.35**	**81.65**	**81.49**
	fasttext	81.10	80.20	80.55	80.37
UCSA-21	NBSVM	76.50	75.01	77.14	76.06
	CNN	72.10	69.79	72.70	71.21
	BiGRU	75.60	73.10	76.70	74.85
	DistilBert(M)	**77.61**	**76.15**	**78.25**	**77.18**
	fasttext	74.57	74.10	74.42	74.60
UCD	NBSVM	83.98	83.56	83.40	83.36
	CNN	84.50	84.35	84.65	84.49
	BiGRU	85.98	85.56	85.40	85.99
	DistilBert(M)	**86.23**	**86.39**	**86.78**	**86.85**
	fasttext	82.10	82.20	82.55	82.37

Bold values represent the best results.

A proposed MLE method is used to aggregate the predictions (soft and hard) from numerous meta-classifiers in two categories of trials on three committees, each with a size five baseline model. The hard predictions of the baseline models are examined in the first category, while the soft predictions are examined in the second. On the basis of a variety of top meta-learners, we can see in Tables 6–9, how accurate the proposed ensemble is. It was found that the suggested ensemble technique greatly outperformed the best-performing baseline model in all shallow meta-learners conducted. With the addition of soft prediction models, the accuracy was even better. The ensemble with SVM as a meta-learner outperforms other meta-classifiers in both hard and soft predication, according to the results. With hard and soft predictions, the suggested ensemble had a higher accuracy than the best individual baseline deep model. Figure 3 show the graphical presentation of hard and soft prediction.

Table 6. Hard and soft prediction comparison of proposed MLE on SAU-18.

Classifiers	Type	Accuracy	Precision	Recall	F1 Score
NB	Hard prediction	80.41	79.32	81.01	80.15
	Soft prediction	**86.42**	**85.21**	**87.62**	**86.39**
GB	Hard prediction	82.03	81.42	82.72	82.06
	Soft prediction	**83.02**	**82.02**	**81.25**	**81.13**
RF	Hard prediction	80.22	77.32	83.31	80.20
	Soft prediction	**85.61**	**84.32**	**86.36**	**85.32**
LR	Hard prediction	80.45	79.21	79.61	79.40
	Soft prediction	**83.81**	**82.41**	**82.25**	**82.32**
SVM	Hard prediction	**86.01**	**85.62**	**85.91**	**85.76**
	Soft prediction	83.11	82.85	83.02	82.93

Bold values represent the best results.

Table 7. Hard and soft prediction comparison of proposed MLE on UCSA.

Classifiers	Type	Accuracy	Precision	Recall	F1 Score
NB	Hard prediction	68.01	67.72	66.22	66.96
	Soft prediction	**84.81**	**82.32**	**83.73**	**83.01**
GB	Hard prediction	**82.72**	**81.33**	**81.46**	**81.39**
	Soft prediction	79.01	78.35	78.61	78.47
RF	Hard prediction	78.61	77.34	77.52	77.42
	Soft prediction	**80.62**	**78.32**	**81.21**	**79.73**
LR	Hard prediction	77.84	76.16	77.22	76.68
	Soft prediction	**79.02**	**78.22**	**78.61**	**78.41**
SVM	Hard prediction	**83.61**	**82.71**	**83.02**	**82.86**
	Soft prediction	79.32	78.35	79.11	78.72

Bold values represent the best results.

Table 8. Hard and soft prediction comparison of proposed MLE on UCSA-21.

Classifiers	Type	Accuracy	Precision	Recall	F1 Score
NB	Hard prediction	78.81	77.61	78.23	77.91
	Soft prediction	**80.91**	**79.21**	**79.72**	**79.46**
GB	Hard prediction	73.73	72.92	72.34	72.62
	Soft prediction	**81.62**	**80.82**	**80.51**	**80.66**
RF	Hard prediction	72.72	71.71	71.52	71.61
	Soft prediction	**78.84**	**77.61**	**77.65**	**77.62**
LR	Hard prediction	**78.01**	**77.83**	**77.55**	**77.68**
	Soft prediction	77.71	76.32	77.22	76.76
SVM	Hard prediction	80.01	79.35	80.02	79.68
	Soft prediction	**82.72**	**81.91**	**82.31**	**82.10**

Bold values represent the best results.

Table 9. Hard and soft prediction comparison of proposed MLE on UCD.

Classifiers	Type	Accuracy	Precision	Recall	F1 Score
NB	Hard prediction	**84.63**	**83.42**	**83.23**	**83.32**
	Soft prediction	82.22	81.81	81.57	81.68
GB	Hard prediction	85.03	84.22	83.19	83.70
	Soft prediction	**86.83**	**85.72**	**85.33**	**85.52**
RF	Hard prediction	85.91	84.31	84.91	84.60
	Soft prediction	**87.05**	**86.85**	**86.38**	**86.61**
LR	Hard prediction	85.92	84.84	84.47	84.65
	Soft prediction	**86.34**	**85.91**	**85.61**	**85.75**
SVM	Hard prediction	87.61	86.65	86.92	86.78
	Soft prediction	**88.22**	**87.36**	**87.65**	**87.50**

Bold values represent the best results.

Figure 3 is a graphical representation of hard and soft prediction on all datasets, we can see that the suggested ensemble outperformed the best individual baseline deep model in both hard and soft predictions. We compare the performance of the proposed ensemble approach against the results of three well-known effective ensemble techniques using the identical created baseline models. These techniques are Bagging, Voting, and Stacking. Table 10 summarizes accuracies on Urdu data.

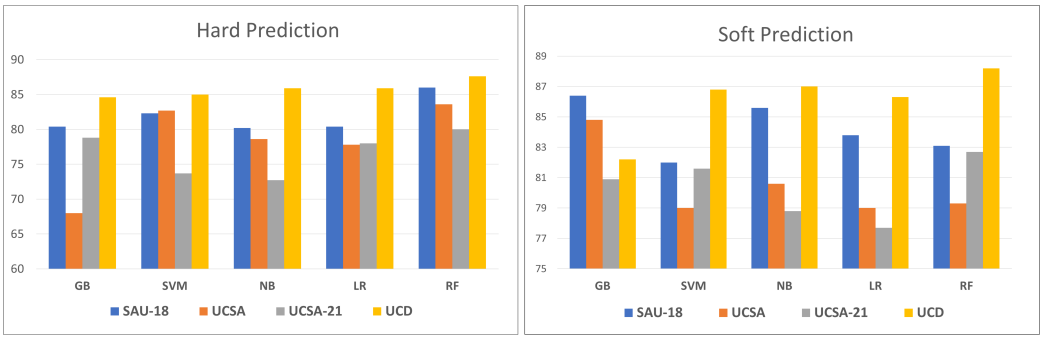

Figure 3. Hard and soft Predictions.

Table 10. Summary of Accuracy.

Dataset	Evaluation Metric	Bagging	Voting	Stacking	Proposed MLE
SAU-18	Accuracy	80.55	81.34	82.32	**86.42**
	Precision	80.33	80.91	81.89	**85.21**
	Recall	80.14	80.61	81.59	**87.62**
	F1-Score	80.22	80.75	81.73	**86.39**
UCSA	Accuracy	76.38	80.16	81.51	**84.81**
	Precision	75.22	79.22	80.33	**82.32**
	Recall	75.14	79.02	80.23	**83.73**
	F1-Score	75.09	79.12	80.14	**83.01**
UCSA-21	Accuracy	72.16	72.35	76.12	**82.72**
	Precision	71.12	71.23	75.91	**81.91**
	Recall	71.06	71.12	75.78	**82.31**
	F1-Score	71.01	71.04	75.68	**82.10**
UCD	Accuracy	80.77	83.28	84.94	**88.22**
	Precision	79.52	82.12	83.22	**87.36**
	Recall	79.43	82.08	83.23	**87.65**
	F1-Score	79.23	82.01	83.12	**87.50**

Bold values represent the best results.

In addition to making quantitative comparisons between the suggested method and other approaches, we also did qualitative comparisons between the methods. A qualitative comparison is carried out based on the amount of time the algorithms require to complete their training. The studies demonstrate that there is a tradeoff between the accuracy achieved by the proposed model and the amount of training time it requires. Proposed MLE model takes more training time than the other models, as seen in Figure 4. It can be concluded that in order to train an efficient model, that can achieve high classification performance, there needs to be a tradeoff made regarding the amount of time spent on training.

The accuracy of the suggested MLE model for Urdu sentiment analysis with the relevant corpora is compared to the performance of other state-of-the art techniques in Table 11. The proposed MLE model have achieved highest accuracy across all corpora evaluated.

Table 11. Comparison of the proposed ensemble MLE model with state-of-the-art models.

Dataset	Classifiers	Accuracy
SAU-18	NB [90]	79.98
	RF [90]	80.92
	DT [90]	80.12
	SVM [90]	81.64
	LSTM	82.35
	RCNN	84.98
	Proposed MLE	**86.42**

Table 11. *Cont.*

Dataset	Classifiers	Accuracy
UCSA	Bi-LSTM [92]	81.10
	CNN-1D + ATT [92]	79.05
	LSTM [92]	78.85
	LSTM + ATT [92]	79.05
	GRU [92]	78.35
	Proposed MLE	**84.81**
UCSA-21	Bi-LSTM [92]	76.50
	CNN-1D + ATT [92]	73.80
	LSTM [92]	73.15
	LSTM + ATT [92]	74.80
	GRU [92]	72.50
	Proposed MLE	**82.72**

Bold values represent the best results.

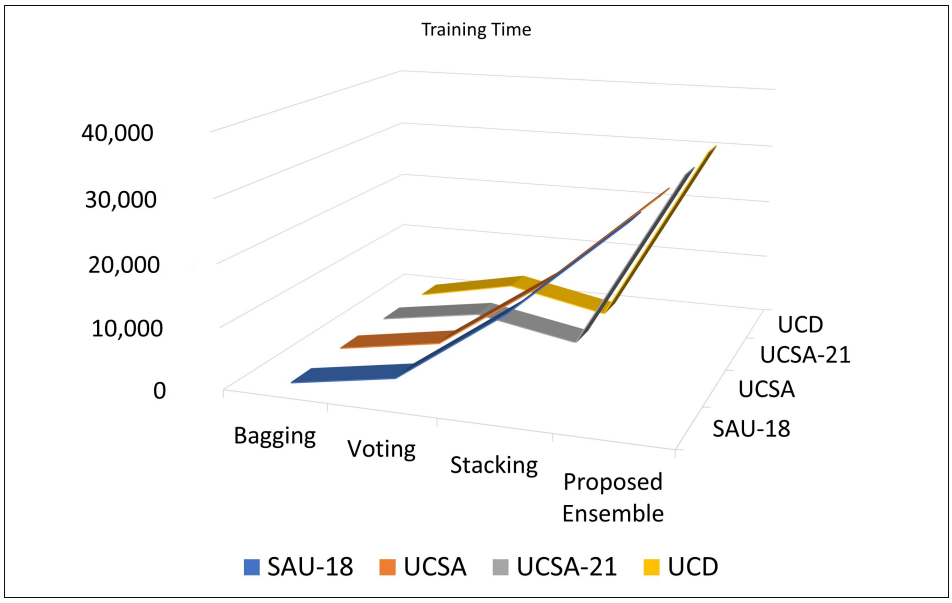

Figure 4. Training time comparison.

The proposed ensemble method improves the accuracy of baseline models with tuned hyper-parameters, according to experiments on all benchmark datasets. Incorporating the class prediction probability distributions of baseline models improves ensemble performance over class label predictions. Furthermore, experimental studies show that combining the results of different classifiers can lessen generalization errors and handle the high variance of individual classifiers. As a result, the ensemble is a sophisticated method of reducing the excessive variation between individual classifiers and maximizing accuracy.

Statistics from the Wilcoxon signed-rank test are used to verify the obtained results across all of the experiment's performance metrics. The *p*-value should be less than 0.05 to indicate that the results obtained are significant [95]. Table 12 shows that the *p*-values for the three datasets used to conduct the experiments are all smaller than 0.05. As a result, the results obtained have a high probability of being correct.

As mentioned throughout the study, there is a dearth of research on Deep learning methods for sentiment analysis in Urdu. There are extremely few studies on this topic, and those that exist used various machine learning classifiers on a small dataset. The proposed MLE model outperforms other models, based on the findings of the study.

In comparison to other resource-rich languages, the Urdu language has a morphological structure that is extremely distinctive, extremely rich, and complex. Urdu is a combination of various languages, including Hindi, Arabic, Turkish, Persian, and Sanskrit,

and contains loanwords from these languages. These are the most prevalent reasons for algorithmic misclassifications. Furthermore, contributing to inaccurate classifications is the fact that the standardization of Urdu text is not yet flawless. To tokenize Urdu text, spaces must be removed or placed between words because the separation between words is not visible. Similarly, in an Urdu statement, the sequence of the words can be altered without altering the meaning, as in "Meeithay aam hain" and "Aam meeithay hain," both of which mean "Mangoes are sweet". Annotating user reviews manually is also one of the causes of misclassification.

Deep learning algorithms not only automate the process of feature engineering, but they are also significantly more capable than machine learning classifiers of uncovering hidden patterns. Machine learning methodologies are invariably less effective than deep learning algorithms due to a lack of training data. This is precisely the case with the Urdu sentiment analysis project, where proposed MLE approach significantly outperform other baseline and state-of-the-art methods.

Table 12. Wilcoxon signed-rank test results.

Dataset	*p*-Value
SAU-18	0.0027
UCSA	0.0035
UCSA-21	0.0052
UCD	0.0054

5. Conclusions and Future Work

Social media platforms have generated vast amounts of data that can be used in a wide range of contexts. Therefore, gauging how people feel about a product or service is impossible without employing sentiment analysis. We found that most of the research on the Urdu language focused on language processing tasks, while only a few experiments were completed in the field of Urdu sentiment analysis. The morphology of the Urdu language is somewhat complicated; word boundaries are not always easily distinguished, and speakers of Urdu use a variety of writing styles while expressing their thoughts in Urdu blogs or other forms of Urdu text. Due to the paucity of previous study in this field, there is still a significant amount of work that needs to be carried out in Urdu SA. Another important idea is that, in recent years, machine learning research has shown that merging classifier outputs helps reduce generalization errors and deal with classifier variance. The ensemble is an elegant way to cope with classifier variance while minimizing general mistakes. Combining different models to produce a predictive model is an old idea. Every ensemble technique weighs models and combines their forecasts to improve performance. In this research, we introduced a new MLE technique that fuses baseline classifier committees for Urdu SA. Increasing classifier diversity improves the proposed ensemble's performance. Many experiments were run to test the ensemble techniques. In addition, we have compared the accuracy of the proposed MLE method to the accuracy of existing ensemble approaches that are extensively utilized in the research literature, and we have conducted this using the same trained baseline models. According to the findings, the suggested meta learning ensemble method not only outperformed stacking, bagging, and majority voting ensemble approaches, but it also greatly improved the performance of the baseline classifiers on all the datasets. Within the scope of this study, a high level of classification accuracy was accomplished for the Urdu sentiment analysis. More and more people comment on various information they care about on social platforms, which can identify the sentiments of public opinion more accurately and efficiently. Therefore, our future work is to pay attention to the social media comment information. Furthermore, the utilization of deep learning techniques to investigate ontology-based concept level sentiment analysis for Urdu text and domain-specific words can be added for efficient sentiment classification by applying different statistical techniques.

Author Contributions: Conceptualization, K.A. and M.I.N.; methodology, K.A. and M.I.N.; software, K.A. and D.L.; validation, M.I.N., S.M.M. and N.A.-K.; formal analysis, K.A., M.I.N. and N.A.-K.; investigation, D.L. and Z.Z.; resources, H.K.A. and O.M.; data curation, O.M.; writing—original draft preparation, K.A.; writing—review and editing, M.I.N. and Z.Z.; visualization, D.L., S.M.M. and H.K.A.; supervision, Z.Z. and D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been funded by the Science Committee of the Ministry of Education and Science of the Republic Kazakhstan (Grant No. AP09259309).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SA	Sentiment Analysis
NB	Naive Bayes
LDA	Latent Dirichlet allocation
SVM	Support vector machine
ANN	Artificial neural network
CNN	Convolutional neural network
DNN	Deep neural network
LSTM	Long short-term memory
SVM	Support vector machine

References

1. Bos, T.; Frasincar, F. Automatically building financial sentiment lexicons while accounting for negation. *Cognit. Comput.* **2022**, *14*, 442–460. [CrossRef]
2. Ahmed, K.; Nadeem, M.I.; Li, D.; Zheng, Z.; Ghadi, Y.Y.; Assam, M.; Mohamed, H.G. Exploiting Stacked Autoencoders for Improved Sentiment Analysis. *Appl. Sci.* **2022**, *12*, 12380. [CrossRef]
3. Li, D.; Ahmed, K.; Zheng, Z.; Mohsan, S.A.H.; Alsharif, M.H.; Hadjouni, M.; Jamjoom, M.M.; Mostafa, S.M. Roman Urdu Sentiment Analysis Using Transfer Learning. *Appl. Sci.* **2022**, *12*, 10344. [CrossRef]
4. Britannica. The Editors of Encyclopaedia. "Urdu Language". Encyclopedia Britannica, 20 October 2022. Available online: https://www.britannica.com/topic/Urdu-language (accessed on 20 February 2023).
5. Asghar, M.Z.; Sattar, A.; Khan, A.; Ali, A.; Masud Kundi, F.; Ahmad, S. Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language. *Expert Syst.* **2019**, *36*, e12397. [CrossRef]
6. Sabah, F.; Hassan, S.U.; Muazzam, A.; Iqbal, S.; Soroya, S.H.; Sarwar, R. Scientific collaboration networks in Pakistan and their impact on institutional research performance: A case study based on Scopus publications. *Libr. Hi Tech* **2019**, *37*, 19–29. [CrossRef]
7. Sarwar, R.; Rutherford, A.T.; Hassan, S.U.; Rakthanmanon, T.; Nutanong, S. Native language identification of fluent and advanced non-native writers. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process. (TALLIP)* **2020**, *19*, 1–19. [CrossRef]
8. Sarwar, R.; Yu, C.; Tungare, N.; Chitavisutthivong, K.; Sriratanawilai, S.; Xu, Y.; Nutanong, S. An effective and scalable framework for authorship attribution query processing. *IEEE Access* **2018**, *6*, 50030–50048. [CrossRef]
9. Bibi, R.; Qamar, U.; Ansar, M.; Shaheen, A. Sentiment analysis for Urdu news tweets using decision tree. In Proceedings of the 2019 IEEE 17th International Conference on Software Engineering Research, Management and Applications (SERA), Honolulu, HI, USA, 29–31 May 2019; pp. 66–70.
10. NLPL Word Embeddings Repository. Available online: http://vectors.nlpl.eu/repository/ (accessed on 19 July 2021).
11. Humayoun, M.; Hammarström, H.; Ranta, A. Implementing Urdu Grammar as Open Source Software. In Proceedings of the Conference on Language and Technology, Khyber Pakhtunkhwa, 7–11 August 2007.
12. Humayoun, M.; Akhtar, N. CORPURES: Benchmark Corpus for Urdu Extractive Summaries and Experiments using Supervised Learning. In *Intelligent Systems with Applications*; Elsevier: Amsterdam, The Netherlands, 2021.
13. Kiritchenko, S.; Mohammad, S.; Salameh, M. SemEval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In Proceedings of the 10th international workshop on semantic evaluation (SEMEVAL-2016), San Diego, CA, USA, 16–17 June 2016; pp. 42–51.
14. Villena-Román, J.; García-Morera, J.; González-Cristóbal, J.C. DAEDALUS at SemEval-2014 task 9: Comparing approaches for sentiment analysis in Twitter. In Proceedings of the 8th International Workshop Semantic Eval. (SemEval), Dublin, Ireland, 23–24 August 2014; pp. 218–222.

15. Nadeem, M.I.; Ahmed, K.; Li, D.; Zheng, Z.; Alkahtani, H.K.; Mostafa, S.M.; Mamyrbayev, O.; Abdel Hameed, H. EFND: A Semantic, Visual, and Socially Augmented Deep Framework for Extreme Fake News Detection. *Sustainability* **2023**, *15*, 133. [CrossRef]
16. Nadeem, M.I.; Ahmed, K.; Li, D.; Zheng, Z.; Naheed, H.; Muaad, A.Y.; Alqarafi, A.; Abdel Hameed, H. SHO-CNN: A Metaheuristic Optimization of a Convolutional Neural Network for Multi-Label News Classification. *Electronics* **2023**, *12*, 113. [CrossRef]
17. Nadeem, M.I.; Mohsan, S.A.H.; Ahmed, K.; Li, D.; Zheng, Z.; Shafiq, M.; Karim, F.K.; Mostafa, S.M. HyproBert: A Fake News Detection Model Based on Deep Hypercontext. *Symmetry* **2023**, *15*, 296. [CrossRef]
18. Awais, D.M.; Shoaib, D.M. Role of discourse information in Urdu sentiment classification: A rule-based method and machine-learning technique. *ACM Trans. Asian-Low-Resour. Lang. Inf. Process. (TALLIP)* **2019**, *18*, 1–37. [CrossRef]
19. Khattak, A.; Asghar, M.Z.; Saeed, A.; Hameed, I.A.; Hassan, S.A.; Ahmad, S. A survey on sentiment analysis in Urdu: A resource-poor language. *Egypt. Inform. J.* **2021**, *22*, 53–74. [CrossRef]
20. Chauhan, U.A.; Afzal, M.T.; Shahid, A.; Abdar, M.; Basiri, M.E.; Zhou, X. *A Comprehensive Analysis of Adverb Types for Mining User Sentiments on Amazon Product Reviews*; World Wide Web: Austin, TX, USA, 2020; pp. 1–19.
21. Poria, S.; Chaturvedi, I.; Cambria, E.; Bisio, F. Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016; pp. 4465–4473.
22. Basiri, M.E.; Kabiri, A. Words are important: Improving sentiment analysis in the persian language by lexicon refining. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2018**, *17*, 26. [CrossRef]
23. Liu, B. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*; Cambridge University Press: Cambridge, UK, 2015.
24. Basiri, M.E.; Ghasem-Aghaee, N.; Reza, A. Lexicon-based sentiment analysis in Persian. *Curr. Future Dev. Artif. Intell.* **2017**, *1*, 154.
25. Basiri, M.E.; Kabiri, A. HOMPer: A new hybrid system for opinion mining in the Persian language. *J. Inf. Sci.* **2020**, *46*, 101–117. [CrossRef]
26. Cambria, E.; Li, Y.; Xing, F.; Poria, S.; Kwok, K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, Ireland, 19–23 October 2020.
27. Abdar, M.; Basiri, M.E.; Yin, J.; Habibnezhad, M.; Chi, G.; Nemati, S.; Asadi, S. Energy choices in Alaska: Mining people's perception and attitudes from geotagged tweets. *Renew. Sustain. Energy Rev.* **2020**, *124*, 109781. [CrossRef]
28. Zhang, L.; Ghosh, R.; Dekhil, M.; Hsu, M.; Liu, B. Combining Lexicon-Based and Learning-Based Methods for Twitter Sentiment Analysis. *Lab. Tech. Rep.-Hpl-2011* **2011**, *89*, 2011.
29. Mudinas, A.; Zhang, D.; Levene, M. Combining lexicon and learning based approaches for concept-level sentiment analysis. In Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining, ACM, Beijing, China, 12 August 2012; p. 5.
30. Ghiassi, M.; Lee, S. A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach. *Expert Syst. Appl.* **2018**, *106*, 197–216. [CrossRef]
31. Chikersal, P.; Poria, S.; Cambria, E.; Gelbukh, A.; Siong, C.E. Modelling public sentiment in twitter: Using linguistic patterns to enhance supervised learning. In *Computational Linguistics and Intelligent Text Processing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 49–65.
32. Fersini, E.; Messina, E.; Pozzi, F.A. Sentiment analysis: Bayesian ensemble learning. *Decis. Support Syst.* **2014**, *68*, 26–38. [CrossRef]
33. Perikos, I.; Hatzilygeroudis, I. Recognizing emotions in text using ensemble of classifiers. *Eng. Appl. Artif. Intell.* **2016**, *51*, 191–201. [CrossRef]
34. Chalothom, T.; Ellman, J. Simple approaches of sentiment analysis via ensemble learning. In *Information Science and Applications*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 631–639.
35. Prusa, J.; Khoshgoftaar, T.M.; Dittman, D.J. Using ensemble learners to improve classifier performance on tweet sentiment data. In Proceedings of the 2015 IEEE International Conference on Information Reuse and Integration, San Francisco, CA, USA, 13–15 August 2015; pp. 252–257.
36. Go, A.; Bhayani, R.; Huang, L. Twitter sentiment classification using distant supervision. *CS224N Proj. Rep. Stanf.* **2009**, *1*, 2009.
37. Jameel, S.; Bouraoui, Z.; Schockaert, S. Unsupervised learning of distributional relation vectors. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 15–20 July 2018; Volume 1, pp. 23–33.
38. Song, M.; Park, H.; Shin, K.-s. Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. *Inf. Process. Manage.* **2019**, *56*, 637–653. [CrossRef]
39. Sharma, R.; Somani, A.; Kumar, L.; Bhattacharyya, P. Sentiment intensity ranking among adjectives using sentiment bearing word embeddings. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 547–552.
40. Xiong, S.; Lv, H.; Zhao, W.; Ji, D. Towards twitter sentiment classification by multi-level sentiment-enriched word embeddings. *Neurocomputing* **2018**, *275*, 2459–2466. [CrossRef]
41. Smetanin, S.; Komarov, M. Deep transfer learning baselines for sentiment analysis in russian. *Inf. Process. Manag.* **2021**, *58*, 102484. [CrossRef]
42. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.

43. Mahmood, Z.; Safder, I.; Nawab, R.M.A.; Bukhari, F.; Nawaz, R.; Alfakeeh, A.S.; Aljohani, N.R.; Hassan, S.-U. Deep sentiments in roman urdu text using recurrent convolutional neural network model. *Inf Process. Manag.* **2020**, *57*, 102233. [CrossRef]
44. Huang, M.; Cao, Y.; Dong, C. Modeling rich contexts for sentiment classification with lstm. *arXiv* **2016**, arXiv:1605.01478.
45. Baly, R.; El-Khoury, G.; Moukalled, R.; Aoun, R.; Hajj, H.; Shaban, K.B.; El-Hajj, W. Comparative evaluation of sentiment analysis methods across arabic dialects. *Procedia Comput. Sci.* **2017**, *117*, 266–273. [CrossRef]
46. Akhtar, M.S.; Ghosal, D.; Ekbal, A.; Bhattacharyya, P.; Kurohashi, S. A multitask ensemble framework for emotion, sentiment and intensity prediction. *arXiv* **2018**, arXiv:1808.01216.
47. Heikal, M.; Torki, M.; El-Makky, N. Sentiment analysis of arabic tweets using deep learning. *Procedia Comput. Sci.* **2018**, *142*, 114–122. [CrossRef]
48. Nabil, M.; Aly, M.; Atiya, A. Astd: Arabic sentiment tweets dataset. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 2515–2519
49. Minaee, S.; Azimi, E.; Abdolrashidi, A. Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. *arXiv* **2019**, arXiv:1904.04206.
50. Müller, M.; Salathé, M.; Kummervold, P.E. Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter. *arXiv* **2020**, arXiv:2005.07503.
51. Syed, A.Z.; Aslam, M.; Martinez-Enriquez, A.M. Lexicon based sentiment analysis of Urdu text using SentiUnits, Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). In Proceedings of the 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, 8–13 November 2010; Volume 6437, pp. 32–43. [CrossRef]
52. Syed, A.Z.; Aslam, M.; Martinez-Enriquez, A.M. Associating targets with SentiUnits: A step forward in sentiment analysis of Urdu text. *Artif. Intell. Rev.* **2014**, *41*, 535–561. [CrossRef]
53. Syed, A.Z.; Martinez-Enriquez, A.M.; Nazir, A.; Aslam, M.; Basit, R.H. Mining the Urdu language-based web content for opinion extraction, in Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). In Proceedings of the Pattern Recognition: 9th Mexican Conference, MCPR 2017, Huatulco, Mexico, 21–24 June 2017; Volume 10267, pp. 244–253. [CrossRef]
54. Mukhtar, N.; Khan, M.A.; Chiragh, N. Lexicon-based approach outperforms supervised machine learning approach for Urdu sentiment analysis in multiple domains. *Telemat. Informat.* **2018**, *35*, 2173–2183. [CrossRef]
55. Hassan, M.; Shoaib, M. Opinion within opinion: Segmentation approach for Urdu sentiment analysis. *Int. Arab J. Inf. Technol.* **2018**, *15*, 21–28.
56. Mukhtar, N.; Khan, M.A.; Chiragh, N. Effective use of evaluation measures for the validation of best classifier in Urdu sentiment analysis. *Cognit. Comput.* **2017**, *9*, 446–456. [CrossRef]
57. Mukhtar, N.; Khan, M.A. Urdu sentiment analysis using supervised machine learning approach. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*, 1851001. [CrossRef]
58. Nasim, Z.; Ghani, S. Sentiment analysis on Urdu tweets using Markov chains. *Social Netw. Comput. Sci.* **2020**, *1*, 269. [CrossRef]
59. Akhter, M.P.; Jiangbin, Z.; Naqvi, I.R.; Abdelmajeed, M.; Fayyaz, M. Exploring deep learning approaches for Urdu text classification in product manufacturing. *Enterp. Inf. Syst.* **2022**, *16*, 223–248. [CrossRef]
60. Ghulam, H.; Zeng, F.; Li, W.; Xiao, Y. Deep learning-based sentiment analysis for Roman Urdu text. *Procedia Comput. Sci.* **2019**, *147*, 131–135. [CrossRef]
61. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
62. Riaz, K. Challenges in urdu stemming (a progress report). In *BCS IRSG Symposium: Future Directions in Information Access*; Association for Computing Machinery: Glasgow, UK, 2007; pp. 1–6.
63. Khan, I.U.; Khan, A.; Khan, W.; Su'ud, M.M.; Alam, M.M.; Subhan, F.; Asghar, M.Z. A review of Urdu sentiment analysis with multilingual perspective: A case of Urdu and roman Urdu language. *Computers* **2022**, *11*, 3. [CrossRef]
64. Liaqat, M.I.; Hassan, M.A.; Shoaib, M.; Khurshid, S.K.; Shamseldin, M.A. Sentiment analysis techniques, challenges, and opportunities: Urdu language-based analytical study. *PeerJ Comput. Sci.* **2022**, *8*, e1032. [CrossRef]
65. Vilalta, R.; Drissi, Y. A perspective view and survey of meta-learning. *Artif. Intell. Rev.* **2002**, *18*, 77–95. [CrossRef]
66. Prodromidis, A.; Chan, P.; Stolfo, S. Meta-learning in distributed data mining systems: Issues and approaches. *Adv. Distrib. Parallel Knowl. Discov.* **2000**, *3*, 81–114.
67. Wang, S.; Christopher, D. Manning, Baselines and bigrams: Simple, good sentiment and topic classification. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Jeju, Republic of Korea, 8–14 July 2012; Association for Computational Linguistics: Mexico City, Mexico, 2012; Volume 2, pp. 90–94.
68. Jabreel, M.; Hassan, F.; Moreno, A. Target-dependent sentiment analysis of tweets using bidirectional gated recurrent neural networks. In *Advances in Hybridization of Intelligent Methods*; Springer: Cham, Switzerland, 2018; pp. 39–55.
69. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. Distilbert a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
70. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. Bert: Pretraining of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MI, USA, 2–7 June 2019; pp. 4171–4186.

71. Zia, H.B.; Raza, A.A.; Athar, A. Urdu word segmentation using conditional random fields (CRFs). In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–25 August 2018; pp. 2562–2569. Available online: http://aclweb.org/anthology/C18-1217 (accessed on 20 February 2023).
72. Akram, Q.-u.-A.; Naseer, A.; Hussain, S. Assas-band, an affix-exception-list based Urdu stemmer. In Proceedings of the 7th Workshop on Asian Language Resources, Singapore, 6–7 August 2009; pp. 40–46.
73. Alam, M.; Hussain, S. Sequence to sequence networks for Roman-Urdu to Urdu transliteration. In Proceedings of the 2017 International Multi-Topic Conference (INMIC), Lahore, Pakistan, 24–26 November 2017; pp. 1–7.
74. Khan, M.; Malik, K. Sentiment classification of customer's reviews about automobiles in roman urdu. In Proceedings of the Future of Information and Communication Conference, Cham, Switzerland, 5–6 April 2018; Springer: Berlin/Heidelberg, Germany; pp. 630–640.
75. Silic, A.; Chauchat, J.-H.; Basic, B.D.; Morin, A. N-grams and morphological normalization in text classification: A comparison on a croatian-english parallel corpus. In Proceedings of the Portuguese Conference on Artificial Intelligence, Guimarães, Portugal 3–7 December 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 671–682.
76. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*; Springer: New York, NY, USA, 2007.
77. Hassan, S.-U.; Safder, I.; Akram, A.; Kamiran, F. A novel machine-learning approach to measuring scientific knowledge flows using citation context analysis. *Scientometrics* **2018**, *116*, 973–996. [CrossRef]
78. Adeeba, F.; Akram, Q.; Khalid, H.; Hussain, S. Cle urdu books n-grams. In Proceedings of the Conference on language and technology, Center for Language Engineering, Karachi, Pakistan, 13–15 November 2014.
79. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
80. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. *arXiv* **2013**, arXiv:1310.4546.
81. Xia, Y.; Cambria, E.; Hussain, A.; Zhao, H. Word polarity disambiguation using bayesian model and opinion-level features. *Cogn. Comput.* **2015**, *7*, 369–380. [CrossRef]
82. Armand, J.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; Volume 2, pp. 427–431.
83. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
84. Domingos, P.; Pazzani, M. On the optimality of the simple Bayesian classifier under zero-one loss. *Mach. Learn.* **1997**, *29*, 103–130. [CrossRef]
85. Breiman, L. Random forests. Mach. Learn. **2001**, *45*, 5–32. [CrossRef]
86. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
87. Kleinbaum, D.G.; Dietz, K.; Gail, M.; Klein, M.; Klein, M. Logistic Regression; Springer: New York, NY, USA, 2002; p. 536.
88. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
89. Opitz, D.W.; Shavlik, J.W. Generating accurate and diverse members of a neural-network ensemble. *Adv. Neural Inf. Process. Syst.* **1996**, *8*, 535–541.
90. Safder, I.; Mahmood, Z.; Sarwar, R.; Hassan, S.U.; Zaman, F.; Nawab, R.M.A.; Bukhari, F.; Abbasi, R.A.; Alelyani, S.; Aljohani, N.R.; et al. Sentiment analysis for Urdu online reviews using deep learning models. *Expert Syst.* **2021**, *38*, e12751. [CrossRef]
91. Khan, L.; Amjad, A.; Ashraf, N.; Chang, H.T.; Gelbukh, A. Urdu sentiment analysis with deep learning methods. *IEEE Access* **2021**, *9*, 97803–97812. [CrossRef]
92. Khan, L.; Amjad, A.; Ashraf, N.; Chang, H.T. Multi-class sentiment analysis of urdu text using multilingual BERT. *Sci. Rep.* **2022**, *12*, 5436. [CrossRef]
93. Harvey, H.B.; Sotardi, S.T. The pareto principle. *J. Am. Coll. Radiol.* **2018**, *15*, 931. [CrossRef]
94. Dong, Y.-S.; Han, K.-S. Text classification based on data partitioning and parameter varying ensembles. In Proceedings of the 2005 ACM Symposium on Applied Computing, Santa Fe, NM, USA, 13–17 March 2005; pp. 1044–1048
95. Taheri, S.; Hesamian, G. A generalization of the wilcoxon signed-rank test and its applications. *Statist. Pap.* **2013**, *54*, 457–470. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.