




Assessing a BERT-based model for analyzing subjectivity and classifying academic articles

Atif Mehmood^{1,2}  · Farah Shahid^{1,2} · Rizwan Khan¹ · Shahzad Ahmed³ · Mostafa M. Ibrahim⁴ · Zhonglong Zheng¹

Received: 6 December 2023 / Revised: 5 March 2024 / Accepted: 5 April 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

The metaverse concept extends beyond virtual worlds and can be applied to collaborative analysis environments. Data analysts worldwide may read academic article extracts in real-time in a shared digital workplace to mine and analyze data using the metaverse. Furthermore, a semantic metaverse in natural language processing might also involve creating a digital environment with linguistic and semantic connections. Many dedicated researchers navigating the complexities of natural language processing in the metaverse era have spent considerable time searching for relevant papers. However, online reviews and evaluations of articles are helpful for their assistance and may save the researcher time. In this work, human specialists manually produced a dataset from four conferences and evaluated subjectively using rule-based techniques. Subsequently, we aim to evaluate the effectiveness of pre-trained word embeddings and pre-trained BERT models seamlessly integrated with convolutional neural networks. This endeavor focuses on the subjective analysis and classification of contributions and previous work sentences extracted from academic literature. For comparison, various deep learning architectures were systematically employed, including long short-term memory-GloVe and bi-directional long short-term memory-GloVe, alongside classical machine learning methods. Our findings show that the proposed BERT model achieved state-of-the-art performance in classification and subjective analysis tasks with an accuracy of 91.50% and F1 score of 91.00%. Finally, we plan to utilize sentence similarity to identify contributions within abstracts, thus outlining potential avenues for future research.

Keywords Natural language processing · Context classification · Subjective analysis · Rule-based techniques · BERT · Text mining · NLP libraries

1 Introduction

Subjective classification is a kind of study that evaluates points of view of people, emotions, and attitudes against entities and their characteristics asserted as a written text. With the fast development of social media on the web, such as blogs, reviews, news, comments,

Extended author information available on the last page of the article

and forum discussions, through the web, more people have shared their opinions and impressions online. Hence, this fascinating dilemma is progressively important in business society and academia. As a result, analyzing the content of the scientific articles is critical for quantifying the quality of the referenced article and calculating its impact [1]. Assigning a new text to a predetermined category based on its similarity to existing texts in that group is an essential challenge in machine learning (ML) and natural language processing (NLP). Text classification, as described, is the process of labeling a new text by comparing it to labeled texts in the training set. By automating this procedure, you may decrease the possibility of human subjectivity leading to incorrect classification while simultaneously accelerating the storage and retrieval of information. Text categorization has several uses, such as reducing email spam, topic-based news article classification, knowledge management, and Internet search engine optimization [2].

As technology develops, the use of sentiment analysis in NLP is growing rapidly. It is critical to comprehend the sentiment and subjectivity shown in academic articles. Researchers and readers can better understand the content of academic articles by using subjective-based analysis, which offers insightful information about the overall tone and personal assessment of articles. NLP techniques are useful to identify and evaluate the opinion stated in textual data. However, readers can find the core ideas and personal opinions in academic work using machine learning algorithms and linguistic patterns [3]. The BERT-MSL model, which stands for BERT-based Multi-Semantic Learning, follows the same Transformer architecture as BERT and uses an aspect-aware augmentation for aspect polarity categorization. A lightweight multi-head self-attention encoding scheme is employed in this model. The authors begin by obtaining initialization parameters for the BERT-MSL model that are enhanced with complete knowledge by utilizing the thorough pre- and post-training of the BERT model. By refining their model on a small corpus, they can quickly adapt it to the Aspect-Based Sentiment Analysis (ABSA) challenge. In addition, the authors present a multi-semantic learning model based on BERT to facilitate aspect-targeted fine-grained sentiment evaluation. Also, the authors provide a way to improve things while being conscious of different aspects, using BERT and multi-head attention techniques [4]. They use the BERT algorithm for aspect-level sentiment categorization, and they achieve groundbreaking performance on three datasets that are available to the public. As a further point of interest, substituting embedding representations with BERT does not intrinsically improve the performance of current neural network models [5].

Analysis based on opinion can be approached from various perspectives, and various relevant resources available, including sentiment lexicons, which are lists of words and phrases connected to different sentiment categories, can be used. With these lexicons, one can rate the subjectivity or objectivity of specific words or sentences in a paper to produce a final sentiment analysis. Furthermore, applying sentiment analysis and text classification relates to assigning specified class labels to raw text content. Several works have concentrated on describing different approaches for text classification [6]. Initially, the previous work on text classification in other domains is described here, and then machine learning techniques used for these tasks are discussed later [7]. Subjective and objective sentence classification distinguishes sentences that express different factual information from the whole text [8]. In various scenarios, social science researchers still lean on conventional qualitative methods like focus groups and interviews due to constraints like budget and time. Consequently, these studies often involve a limited number of participants. However, many social researchers use advanced technology to unearth social patterns, sometimes complementing or replacing traditional qualitative approaches [9].

This study aims to delve into subjective-based analysis and classification to assess the contributions made by researchers in conference articles. Moreover, we seek to explore the intricate challenges inherent in this endeavor, including domain-specific vocabulary usage, divergent writing styles, and the presence of conditional statements, all of which pose significant hurdles to effective opinion classification. The originality of our research lies in creating a specialized corpus tailored to the field of Natural Language Processing (NLP), sourced from four prominent NLP conferences—ACL, NAACL, EMNLP, and CoNLL. This corpus is meticulously annotated by domain specialists, ensuring its relevance and accuracy for subsequent analysis. To preprocess the raw text data, we employ sophisticated pre-trained word embedding techniques and embedding layers, which serve as critical components in our preprocessing pipeline. In our methodology, we leverage Convolutional Neural Networks (CNN) integrated with Long Short-Term Memory (LSTM) networks, Bidirectional LSTM (Bi-LSTM), and Bidirectional Encoder Representations from Transformers (BERT). These neural network architectures enable us to extract higher-level representations of features from the text data while also possessing the ability to effectively process sequences, thereby capturing the nuanced context in subjective language. Furthermore, the key features of our proposed methodology can be outlined as follows:

- Compiled an academic conference corpus by extracting content from abstracts, limitations, and conclusions sections of articles, ensuring comprehensive coverage of relevant textual data.
- Conducted a comparative analysis between human-assigned labels and NLP-based analyzers, including Flair, Vader, and Textblob, to evaluate their effectiveness in opinion classification tasks. Additionally, feature vectors were generated using pre-trained word embedding such as word2vec and GloVe to enhance the representation of textual features.
- It has formulated and implemented a hybrid CNN framework designed to unveil latent text properties, leveraging GloVe embedding both with and without LSTM layers. The performance of the framework was rigorously assessed to determine its efficacy in capturing nuanced textual information.
- We introduced a novel approach of stratified cross-validation to mitigate challenges associated with unbalanced datasets, ensuring robust model performance. Weighted BERT extraction was utilized for feature vector generation, and comprehensive model analyses were conducted against baseline classifiers, including Naive Bayes, KNN, and SVC, to ascertain the superiority of the proposed methodology.

The organization of the article is as follows: Section 2 presents the literature review regarding subjective analysis and classification using NLP, ML, and DL techniques. Section 3 provides the materials and methods of the proposed models I, II, and III and performance metrics. Sections 4 and 5 comprise the simulation results and discussion of the designed models. The last section includes the conclusion.

2 Related works

In a current literature review, the analysis of sentiment classification is based on two main approaches: lexicon-based techniques and ML/DL-based techniques. A lexicon-based approach gathers lists of words and sentences with positive and negative meanings [10]. These tactics are straightforward and practical, allowing scalable computing performance to

address general sentiment analysis problems. However, with lexicon-based techniques, the linguistic document has low coverage and insufficient information, requiring human labor to classify it [11]. In this perspective, different researchers presented lexicon-based techniques to represent the syntactic and semantic information of words by their co-occurrence patterns that can be helpful for sentiment classification [12]. Apple et al. proposed an article to present a hybrid approach to the problem of sentiment analysis, with a particular emphasis on analysis at the phrase level. This innovative approach incorporates basic natural language processing (NLP) techniques to estimate the semantic orientation polarity and its intensity for sentences. This sentiment lexicon has been supplemented using SentiWordNet and fuzzy groups. By establishing a foundation for sentiment computation, this all-encompassing methodology makes it possible to gain a more profound comprehension of the feelings conveyed during sentence construction [13]. Authors have developed a method to collect concept-level sentiments regarding political themes discussed on Twitter. In the discipline of NLP, various techniques and algorithms are applied to conduct subjective analysis. Different types of machine learning models, rule-based systems, and deep learning approaches are included in this category. To determine the subjectivity and sentiment conveyed in a text, these methods use components such as syntactic structures, lexical clues, and semantic information [14].

This work intends to examine and determine if it is possible to autonomously identify hate speech by utilizing domain-specific word embedding as attributes and a bidirectional LSTM-based deep model as a classifier. This strategy ensures that the word is made to have its negative connotation given to it, which is a highly beneficial way of identifying words that have been coded [15]. Muhammad et al. presented an approach that captures the contextual polarity over local and global context to lexical sentiment classification of social media genres [16]. Fernández-Gavilanes et al. have proposed a dependency parsing-based method to predict online text classification [17]. Araque et al. have employed word embedding algorithms such as Word2vec and Glove to transform words into meaningful vectors, which has some benefits over the bag-of-words representations [18]. The performance of word2vec and glove depends on the corpus size; different applications, such as sentiment analysis of tweets for clinical text and text classification, have employed the glove and word2vec as pre-trained word representations as to the input of deep learning models [19]. This paper proposes a methodology that can dynamically produce aspect and context word vector encodings for use in review writing. Then, in order to process the vectorized phrases in parallel, a transformer structure is used to convey the aspect-context pairings semantically. The next step is for the model to learn the most important parts of the reviews by using a synthetic attention mechanism [20]. Bansal et al. proposed the word2vec model for sentiment analysis. They proposed the methods of vector representation for Bengali words to classify the positive and negative scores against each word, define the sentiment score of a certain text, and achieve higher accuracy [21]. In all, machine learning approaches enhanced the accuracy of sentiment classification. Therefore, the CNN technique is applied as a convolutional filter to learn the local dependencies and uses a pooling layer to extract global features, and a deep recursive neural networks (DRNN) approach with binary parse trees has been employed to capture sentence representation for sentiment classification [22].

The researchers employed a hybrid CNN and LSTM to learn the representation of words and then transformed these semantic vectors into document form using a gated recurrent neural network [23]. Rao et al. have used a new method of sentence vectors with LSTM (SR-LSTM) to learn the syntactic and semantics of sentences [24]. Li et al. proposed CNN-Bi-LSTM to encode the contextual sentiment polarity and quality analysis to compare the performances of CNN-Bi-LSTM and Bi-LSTM [25]. The creation of word embedding was a significant advance that led to today's language models. By anticipating words, it is possible

to move beyond one-hot and similar encodings and turn words into vectors that encode their meaning and are more valuable for downstream applications [26]. Attention and transformer models, which were based on attention, were the subjects of yet another significant advance. Compared to LSTMs, they allow for significantly more parallelization, which enables models built on top of them to be trained on far bigger and more diverse corpora. One important early model that made use of them is BERT [27]. Within the scope of this investigation, the authors make an effort to use the BERT language model for the purpose of producing word representations that are obtained from reviews on social media. In addition to this, they make use of DCNNs in order to improve the model so that it may be utilized in certain smart city scenarios. Furthermore, SenticNet is incorporated into the proposed architecture as a knowledge base, which makes it possible to incorporate concept-level analysis into the framework [28]. The bidirectional slice-gated recurrent unit, also known as BiSlice-GRU, is a notion that the writers of this article present. The slicing network that was implemented makes it possible for the model to strike an acceptable compromise between effectiveness and efficacy while also facilitating the acceleration of the training process [29].

3 Materials and methods

This paper is a subjective analysis and content classification of academic journal articles through NLP techniques and different deep-learning models. The proposed approach consists of several processing steps, each described in detail in this section, particularly addressing the construction, cleaning, and analysis of the customized dataset (including contribution and limitation-related sentences). The evaluation involves a comparative study between human-based corpus annotation and NLP rule-based techniques. A flow chart of the proposed work is shown in Fig. 1.

This analysis aims to assess the efficacy and reliability of each approach in annotating the corpus. It also tries to highlight the strengths and potential areas for improvement in both methods, demonstrating their respective contributions to the overall annotation process. The next step is employing pre-trained word embedding integrated with a variety of DL time series models for subjectivity classification, such as weighted BERT, LSTM-GloVe, Bi-LSTM-GloVe, Model-I (LSTM with single embedding layer), Model-II (an improved version of Model-I with a 1D convolutional layer), Model-III (same architecture as Model-II with the pre-trained GloVe word embedding). Baseline algorithms such as Naïve Bayes, KNN, and SVC have been experimented with concerning performance measurements to compare the proposed models. Figure 2 demonstrates the high-level design of the implemented system.

3.1 Rule-based subjectivity analysis of annotated corpus

A series of NLP techniques for subjective analysis is employed to conduct a comprehensive comparative analysis. This choice is motivated by the need to identify differences between labels provided by human experts and those produced by the techniques. Three types of natural language toolkit techniques, Text blob, Vader, and Flair, are employed in this study. One well-known Python library is Text blob, which uses a sentiment lexicon with predefined terms that give every word a score. After that, weighted averages of the results are calculated to get an overall sentiment score. Furthermore, Vader is another technique primarily dealing with social media comments, movie reviews, product ratings, and emotion-based language. All

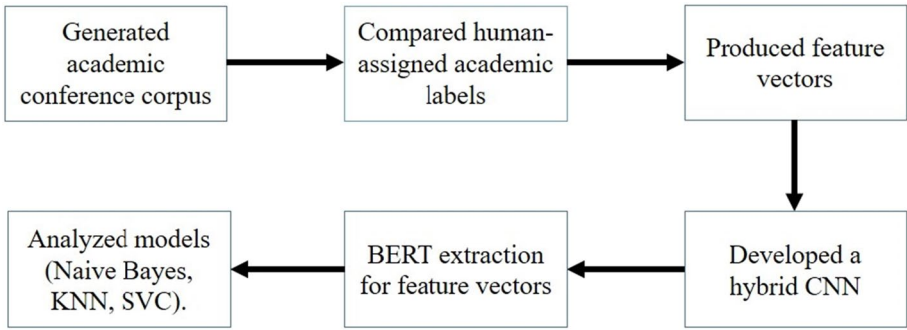


Fig. 1 A flowchart of the proposed work

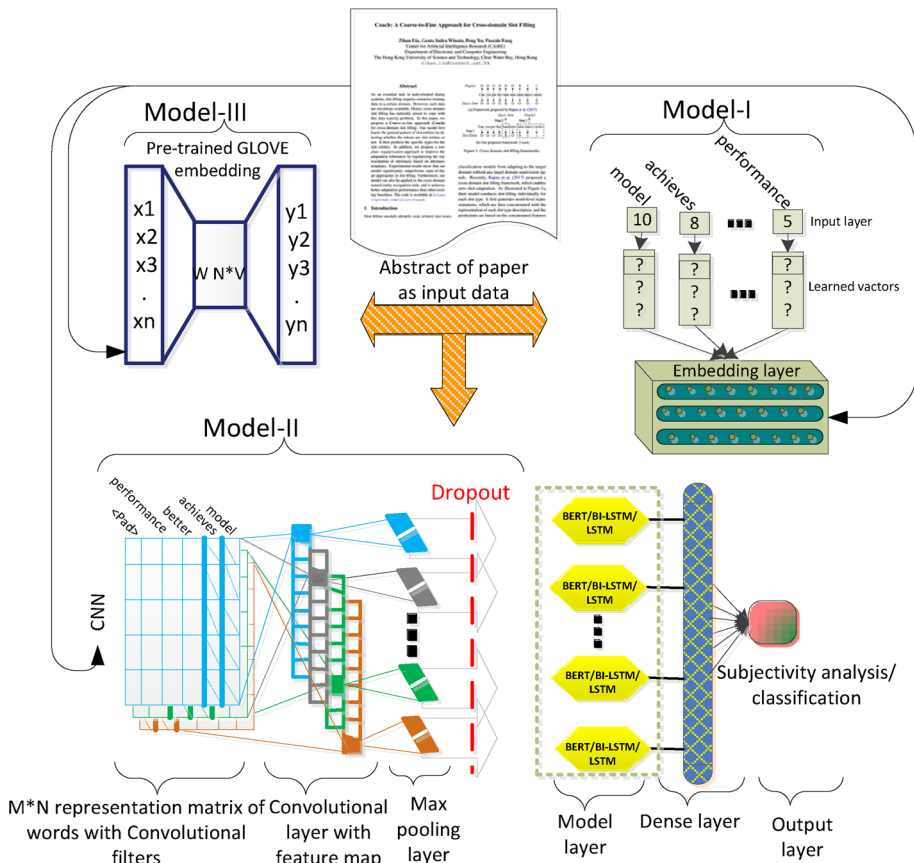


Fig. 2 A graphical overview of the generated corpus, GloVe embeddings, and proposed hybrid deep learning model for Subjectivity analysis and classification of academic articles

semantic ratings are computed using the compound score, limited to -1 (the most severe negative) and +1 (the extreme positive). Finally, modern NLP techniques can be built into text segments using Flair, which performs far better than earlier versions. Flair employs a pre-trained algorithm to identify positive or negative comments and produces prediction probability values as a label. Figure 3 of the average, minimum subjective analysis, standard deviation, and subjective analysis using the Textblob and Vader and Flair rule-based analysis of the annotated corpus. Upon close examination of the subfigures in Fig. 3, it can be observed that the annotations of Flair for subjectivity analysis closely align with those made by human annotators.

3.2 Pre-trained word embeddings

Word embedding is a fundamental technique in NLP for representing words in a dense vector form, capturing both syntactic and semantic problems. The initialization of these embedding vectors, which are subject to modifications via back-propagation during training, can be performed through random assignment or by using pre-trained word embedding supplied by models such as Word2Vec or GloVe. The size of these vectors is determined by the dimensionality of the embedding space, which is a crucial hyper-parameter. This dimensionality is frequently lowered to a more manageable amount, typically 100 to 300, to balance computing demand with the requirement to retain helpful information on word associations. We use GloVe to build word embeddings in our technique, concentrating on a vocabulary size of 1000 words drawn from a chosen selection of abstract datasets. This pre-trained word embedding layer serves as the initial foundation for our network design, providing critical contextual information for later stages of analysis.

3.3 Embedding Layer

An essential component of time series models for NLP tasks, especially RNNs, depends on the embedding layer as a critical building block. It is the initial layer of the network and is responsible for converting input tokens, such as words or symbols, into complex

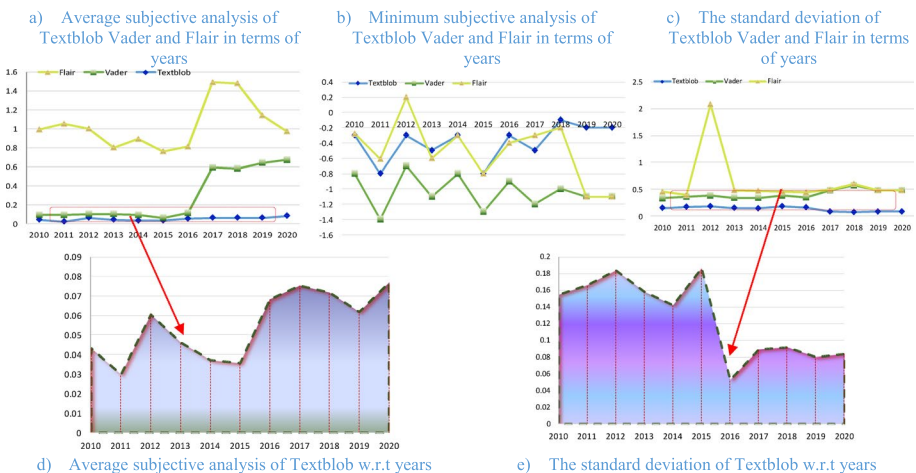


Fig. 3 Comprehensive comparisons evaluations between NLP techniques: Textblob, Vader, and Flair of labeled corpus for ACL repository w.r.t ten years

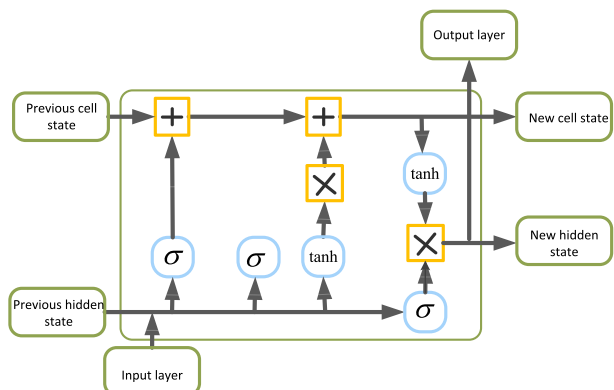
mathematical representations known as embeddings. The primary goal of the embedding layer is to preserve the relationships between words and their semantic meaning in a continuous vector space. This representation, as compared to coping with dense and high-dimensional formats like one-hot encodings, enables the neural network to analyze and learn from textual material effectively. This study uses a text of abstract limitation or a conclusion sequence of words as input to the model. The word sequence must be transformed using the embedding layer; $x = \{x_1, x_2, \dots, x_T\}$ to the R^E low dimensional vector space. Here, E and T are defined as the size of the embedding layer and the number of words in the abstract.

3.4 Time series models for subjectivity classification

Recurrent neural networks (RNN) have been demonstrated to be effective algorithms for various kinds of NLP applications, including the analysis of sentiment. The process of determining the contribution/limitations represented in an article of text, such as positive (emphasize new contribution) or negative (highlight the previous work), is known as subjective analysis. Although they can capture the sequential structure of words, RNNs are especially suitable for the analysis of sentiment. RNNs, unlike standard feed-forward neural networks (FFNN), feature a feedback loop that allows them to remember prior inputs. This memory allows the network to comprehend sequential input, such as phrases or paragraphs, by taking context and word relationships into account. Many neural network models, including RNN [30] and CNN, are utilized for text modeling.

In this paper, LSTM (advanced version of RNN) deploys a recursive neural network, which outperforms several NLP tasks [31]. LSTM network is an extended form of RNN that was introduced by Hochreiter and Schmidhuber and explicitly addresses the problems of learning long-term dependencies. The vanishing gradient problem, which is dominant in the training of the classical RNN model, is addressed with LSTM. The vanishing gradient issue happens when the gradient gets exceedingly small, making long-range relationships harder for the network to understand. LSTM addresses this issue by integrating a system of gates that regulates the flow of data across the network. The gates control how much data is retained or destroyed at each phase, allowing LSTMs to capture long-term dependencies. Although there are various LSTM versions, the architecture of LSTM is employed in this study in a manner similar to [32], shown in Fig. 4.

Fig. 4 Structure of LSTM cell



The LSTM units are defined as an input i_t , a forget f_t and output gate, a hidden state h_t , and a memory cell c_t of a single time step t . The elements of gating vectors are represented as $[0, 1]$ for i_t, f_t and o_t . The equation of LSTM is defined as:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ \tilde{c}_t \end{bmatrix} \approx \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(w_a \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_a \right) \tag{1}$$

$$c_t = c_t \otimes i_t + c_{t-1} \otimes f_t \tag{2}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{3}$$

Here, represented as input sequence, are the weight and bias vectors. σ , and \otimes denotes as sigmoid function and element-wise multiplication. The hidden state is worked as follows:

$$h_t = lstm(h_{t-1}, x_t, \theta_a) \tag{4}$$

where θ_a represents all LSTM parameters in a hidden state.

The representation of LSTM for sentiment classification is defined with an input text sequence $x = \{x_1, x_2, \dots, x_T\}$. There is an embedding layer to denote each word in input x_t in vector form. This is the output of the whole sequence, followed by the fully connected softmax layer.

$$y' = softmax(wh_T + b) \tag{5}$$

y' the prediction probabilities and weight w to be learned, and b is a bias. A corpus of N samples (x_i, y_j) .

and the model parameters are trained to reduce the cross-entropy of the estimated and the actual distributions.

$$loss(y', y) = - \sum_{k=1}^N \sum_{l=1}^C y_k^l \log(y_l^k) \tag{6}$$

Here, are the actual labels the estimated values, and C the number of classes.

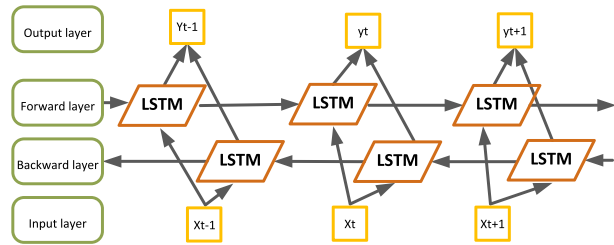
In addition, it works in a unidirectional way. To further optimize the model, bidirectional LSTM (Bi-LSTM) is used, which is comprised of forwarding and a backward pass that is employed to read the sequence of the corpus in sequential and reverse order, respectively, illustrated in Fig. 5.

In this way, the model captures the semantics of words in both directions to improve the expression of the text. The representation of word vector on both ends; the left and right context of the word is defined through expressions:

$$c_r(x_i) = g(w_r c_r(x_{i+1}) + w_{sr} E(x_{i+1})) \tag{7}$$

$c_l(x_i)$ Represents the left context vector and w_l is the weight matrix that converts the one hidden layer to the other hidden layer w_{sl} comprised of the current word semantic with the left context of the next word, $c_l(x_{i-1})$ and $E(x_{i-1})$ shows the left context of the previous

Fig. 5 Architecture of Bi-LSTM



word and previous word vector respectively. It is calculated for the right context in a similar way to the left context.

However, Bi-LSTM uses convolution to combine both contexts of the current hidden state and form a new expression. Equation (4) denotes the concatenation of the left and right context vectors to eliminate word ambiguity and accurate representation.

$$x_i = (c_l(x_i), E(x_i), c_r(x_i)) \quad (8)$$

After getting the x_i word expression from this equation, the tanh activation function is applied for linear transformation, and the pooling layer gets these results in different lengths, converts them into fixed-length vectors, and is capable of learning the information throughout the text.

$$y_i = \tanh(w_i(x_i) + b_i) \quad (9)$$

3.5 BERT-based model for subjectivity classification

The introduction of Language Models (LM) and transformers has resulted in significant advancements in the field of NLP. BERT, for instance, is a pre-trained language model trained on a large corpus of unannotated data, including 800 million words from Books Corpus and 2,500 million words from English Wikipedia. Notably, it was introduced without a specific task, making it adaptable for a wide range of NLP tasks through fine-tuning. BERT excels at comprehending the contextual meaning of words by considering both their left and right context. Moreover, it can represent words and sentences as numeric vectors. BERT comes in two standard architectures: BERT-base, which comprises 12 encoder layers with 110 million parameters and 768 hidden layers, and BERT-large, which features 16 encoder layers with 340 million parameters and 1,024 hidden layers. Several researchers who used BERT-based LMs to classify misinformation reported to have excellent performance by employing LMs as opposed to standard machine learning methods. Other researchers compared the performance of LMs to assess the cross-source failure problem in current misinformation detection methods. They concentrated on developing generalizable representations to apply the classification model to real-world data. They investigated cross-source generalizability using one dataset as a training set and the rest as testing sets. Therefore, this study is included as a BERT model. This big bidirectional transformer has already undergone pre-training to classify the subjectivity classification of the academic articles and compare it with state-of-the-art techniques.

3.6 Performance metrics

Different performance measurements are used to evaluate the actual and predicted content labels for classification. Accuracy and F1 scores are calculated in percentages and their expressions in Eqs. (10) and (13). Recall and Precision are computed in Eqs. (11) and (12), respectively. Here, Recall is used to calculate "how many of this class you find over the whole number of elements of this class," and Precision is for "how many are correctly classified among that class." TP, denoting True Positives, represents the count of accurately predicted positive instances. Conversely, FP, signifying False Positives, pertains to the tally of erroneously predicted positive instances. True Negatives (TN) encapsulates the number of accurately predicted negative instances. Lastly, FN, referring to False Negatives, accounts for the instances incorrectly predicted as negative. The F1 score evaluates the mean between precision and recall. The best measures for accuracy and F1 score are close to 1; near to zero means not good classification.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \quad (10)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (11)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (12)$$

$$\text{F1score} = 2 \times \left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \right) \quad (13)$$

4 Experimental results

4.1 Description and pre-processing of corpus

The main purpose of this paper is to make use of a distinct dataset that is constructed from top NLP conferences: ACL, EMNLP NAACL, and CoNLL, which comprised articles from the years 2011–2023. For this aim, these four conference proceedings chapters with different numbers of papers were downloaded from the ACL database (<https://aclanthology.org/>) to create a corpus of titles, abstracts, limitations, and conclusions containing 13,221 papers. Table 1 shows the year-wise articles from selected conferences. After that, the downloaded repositories related to abstracts, conclusions, and limitations were manually eliminated in order to restrict the scope of the preceding annotations of CS and PW. Generally, authors express their contribution in terms of model, dataset, and results, highlight or emphasize failures in previous work, and address the limitations in the "abstract" or "conclusion" and the "limitation" sections. Therefore, if a paper contains an "abstract," "limitations," and "conclusion," related sections are extracted directly. Otherwise, papers that do not include these sections are deleted from the generated corpus.

The opinion mining and classification in this paper represent a specific domain of subjectivity analysis of abstracts for academic publications; this represents a noteworthy advancement in the realm of evaluating author contributions in the fields of NLP and AI. The assessment of rule-based NLP techniques may be summarized as follows: personal opinions from the selected sentences that reflect subjectivity framing in terms of the author's expressing

Table 1 Detail presentation of a number of articles per year of top international conferences in the NLP domain

| Years | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| # of papers | 157 | 155 | 111 | 136 | 121 | 171 | 156 | 192 | 253 | 657 | 775 | 572 | 604 | 912 |
| ACL | 126 | 150 | 140 | 206 | 227 | 313 | 265 | 324 | 550 | 683 | 752 | 848 | 829 | — |
| EMNLP | 147 | — | 98 | 141 | — | 187 | 182 | — | 206 | 424 | — | 478 | 443 | — |
| NAACL | 28 | 30 | 17 | 26 | 21 | 39 | 32 | 46 | 57 | 98 | 54 | 53 | 29 | — |
| CoNLL | | | | | | | | | | | | | | |

SOTA new contribution and only highlighting shortcomings of earlier work. Furthermore, data was collected and extracted to analyze every article for various attributes, such as title, abstract, limitations, future work, and conclusion, which were subsequently listed using an Excel spreadsheet. These attributes were considered the most important aspects of obtaining the main discussion points of this subjectivity analysis and classification. Table 2 shows a small portion here as an explanation of both CS and PW framing subjective analysis, thereby helping in the further classification of the context of academic papers.

Figure 6 illustrates the number of words and number of sentences per abstract (the original content of this abstract text). Furthermore, every abstract/conclusion consists of approximately a range of words from 44 to 280 and sentences from 2 to 14. Still, owing to space constraints, a selected number of examples are shown in Fig. 6.

Data preprocessing and cleaning are essential before using NLP corpus for any machine learning applications. The customized corpus consists of a wide range of letters and alphabets that create inconsistency and redundancy and affect the results. Therefore, to increase the efficacy of data, it is tokenized to use most of the related information from text, removing punctuation from the original data and then using the lower function to convert the sentences into lowercase. Subsequently, removing stop words is essential like 'a', 'eg', 'the', and 'email', which hold minimal semantic values [30]. This is the reason for their exclusion; these frequently occurring words provide little discriminating information for classification. The NLTK python package makes it easier to carry out these data-cleansing procedures. Overall, the detailed process of proposed methods and models is described in Table 3.

Subsequently, the dataset is manually labeled by reading each abstract or conclusion and limitation that is described in detail in Table 2 and Fig. 6. During the experimental process, the corpus is divided into seventy percent (70%) as a training set and thirty percent (30%) for the testing set, which is passed to the proposed algorithm for classification with parameter values, are given in Table 4. Moreover, the imbalanced class data is handled through a stratified cross-validation approach. It is beneficial to ensure that the subset of training data

Table 2 Random sample of abstracts annotated as contribution sentences (CS) and previous work (PW) framing

| | |
|--|--|
| <ul style="list-style-type: none"> • Neural-based end-to-end methods to NLG,..., <i>generate extremely good results</i> over baseline by an average of more than 8.0 BLEU points. • Existing top techniques using the structure-to-sequence architecture,..., <i>assessments highlight our framework's cutting-edge performance</i> auto-evaluated metrics and case studies. | <ul style="list-style-type: none"> • Neural abstractive summarizers generate summary texts,..., <i>find systems fail to understand the source text</i> in a majority of the cases. • Can AI learn to express inflectional morphology and generalize to new words in the same way as human speakers do? ... models may still <i>struggle with generalization to minority classes</i>. |
|--|--|

Fig. 6 Representations of academic article title w.r.t four repositories

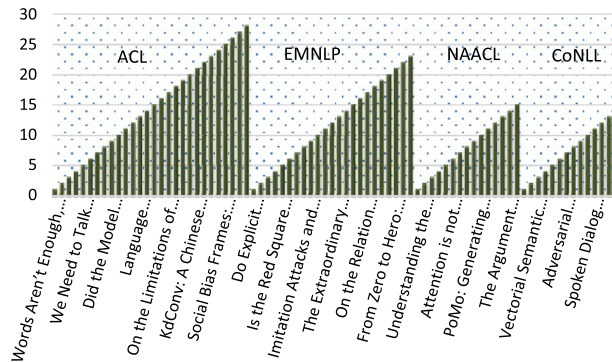


Table 3 Pseudo code of proposed method

```

# Step 1: Construct and Clean Dataset
function construct_and_clean_dataset():
    # Step 1.1: Load raw data
    raw_data = load_raw_data()
    # Step 1.2: Preprocess data (e.g., tokenize, remove stop words)
    preprocessed_data = preprocess_data(raw_data)
    # Step 1.3: Extract contribution and limitation sentences
    contribution_sentences = extract_contribution_sentences(preprocessed_data)
    limitation_sentences = extract_limitation_sentences(preprocessed_data)
    # Step 1.4: Create a customized dataset
    customized_dataset = create_customized_dataset(contribution_sentences,
    limitation_sentences)
    # Step 1.5: Analyze the dataset (e.g., statistical analysis)
    analyze_dataset(customized_dataset)

```

```

# Step 2: Evaluate Corpus Annotation
function evaluate_corpus_annotation():
    # Step 2.1: Human-based annotation
    human_annotations = perform_human_annotation()
    # Step 2.2: NLP rule-based annotation
    nlp_annotations = perform_nlp_rule_based_annotation()
    # Step 2.3: Comparative analysis
    comparative_analysis(human_annotations, nlp_annotations)

```

```

# Step 3: Semantic Classification with DL Models
function semantic_classification_with_dl_models():
    # Step 3.1: Load pre-trained word embeddings
    word_embeddings = load_pretrained_word_embeddings()
    # Step 3.2: Train deep learning models
    trained_models = train_dl_models(word_embeddings)
    # Step 3.3: Perform subjective classification
    semantic_classification_results = perform_subjective_classification(trained_models)

```

```

# Step 4: Compare Models with Baseline Algorithms
function compare_models_with_baseline_algorithms():
    # Step 4.1: Experiment with baseline algorithms
    baseline_results = experiment_with_baseline_algorithms()
    # Step 4.2: Compare results
    compare_results(semantic_classification_results, baseline_results)

```

```

# Main Execution
function main():
    construct_and_clean_dataset() # Step 1
    evaluate_corpus_annotation() # Step 2
    semantic_classification_with_dl_models() # Step 3
    compare_models_with_baseline_algorithms() # Step 4
    display_system_design().

```

has an equal number of examples from each class. The system specifications to conduct the simulations for this work are described as GPU: precision DEL 7670 workstations; RAM: 64 GB DDR4; graphics card: 2000; SSD drive: 1 TB.

4.2 Evaluating flair and human-assigned annotations

In this study, the raw data was initially collected from NLP international conferences and pre-processed using stop words and punctuation, as discussed in the above section. After

Table 4 Adjustable parameter values for proposed time series models

| Parameter | Value |
|-----------------------|----------------------|
| Filter size | 5 |
| Word vector dimension | 100 |
| Pooling layer | 1-max pooling |
| Dropout rate | 0.2 |
| Loss | Binary cross-entropy |
| Optimizer | Adam |
| Vocabulary size | 1000 |
| Learning rate | 0.0001 |
| Neural units | 64 |

that, NLP techniques are implemented in the overall repository, and statistical results are shown as a line graph in Fig. 3, Section 3.1. Subfigures (a-c) of Fig. 3 show that each year has a distinct average, the minimal subjectivity analysis over all ten years. Meanwhile, Subfigures (d and e) show the average and standard deviation of opinion analysis using Textblob, Vader, and Flair. Among all NLP strategies, Flair works well. In addition, Fig. 7 displays the comparison results of the NLP methodology Flair and human-assigned labels to the contribution and limitation sentences when applied to the overall generated corpus using randomly selected abstracts. It can be inferred from this analysis that Flair-based annotation mainly considers the sentences in the author's contribution. However, the fundamental disadvantage of the rule-based approach to subjectivity analysis is that it is only concerned with individual words and entirely disregards the context in which it is used.

Therefore, when compared to the human annotation of (contribution and previous work) CS/PW sentences, the concept of using these NLP subjectivity analysis methodologies for annotation is less appropriate. These NLP-based analyzers tend to enormously weight keywords with high symmetry, even when additional words with a weaker symmetry (or negative words) change the overall attitude of the phrase. The comparison between manually labeled and Flair-based subjectivity scores has been shown in Fig. 7. Although the native language from ordinary life contains a range of rules for subjective analysis, hard-coded criteria function effectively in many scenarios. While encouraging results have not been obtained, this is the motivation for manually labeling the CS/PW text of articles in order to

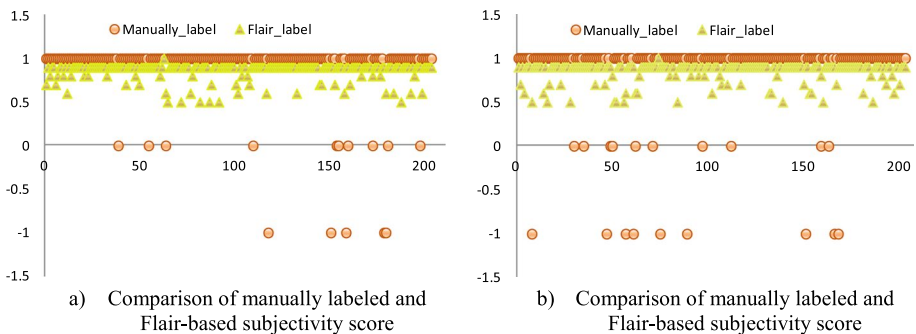


Fig. 7 Scatter plot representation of the relationship between manually and Flair-based labeled of corpus shown in the a & b subfigure for the years 2020 and 2019, respectively

assess the author's contribution to the research domain and to assist the reader in expanding on prior work.

Following that, the next step is to preprocess the customized corpus, convert it into a token form using the embedding layer as discussed in Section 3.2, and be ready to pass to the proposed models. Deep learning time series models such as LSTM-GloVe, Bi-LSTM-GloVe, pre-trained word embeddings as Model (I, II, III), and pre-trained model BERT have been employed in the field of classifying the context of scientific papers. Here, Model-I generates over the LSTM and a single embedding layer. In contrast, Model-II is an enhanced version of Model-I made up of an additional 1D convolutional layer built on top of the LSTM layer to minimize training time and complexity. Model-III employed the same architecture as Model-II but with the addition of pre-trained GloVe word embedding. Therefore, the results are obtained through these models in terms of relative performances, specifically, positive recall, negative recall, positive precision, and negative precision, which are computed as in Fig. 8.

This research focuses on binary classification with several classifiers. Therefore, the based model reports high per-class-sensitivity (recall) and per-class-specificity (precision) values of 0.94 and 0.89 see green bar chart in Fig. 8. For the first two of these, sufficient numbers are reported to reproduce the reported metrics, as well as to calculate the corresponding positive class precisions, which are written of Model-I, and Model-III as 0.83 and 0.77, respectively. The two models, Bi-LSTM-GloVe and LSTM-GloVe, show higher recall and precision for the positive class at 0.71 and 0.80, respectively. Your precision is so poor for label -1/0 because there are many more CS sentences than PW sentences, increasing the chance for false positives to occur and affecting your precision. It can be observed that Model II works better for positive recall and positive precision among all ML and DL classifiers. Therefore, the dataset is imbalanced and focuses on maximizing the positive and negative recall. Another experiment uses different training sets and testing data through a stratified fold cross-validation approach. Table 5 better represents Model-I and Model-II performance regarding other evaluation metrics.

The different stratified fold schemes (5, 10, 15, and 20) are used to repeat the random sub-sampling by assigning the samples to train (60%), validate (20%), and test (20%) sets. The random assigning depends on the fold size and is performed at different times to generate different train, validate, and test sets. Consequently, two proposed Model-I and Model-II (hybrid CNN and LSTM with word embeddings) classifiers are built for each set of data, and their results are averaged to measure overall performance. A comparative measurement of these models with respect to precision, recall, accuracy, and F1 score is

Fig. 8 Bar graphs of proposed techniques integrated with GloVe embeddings and baseline models in terms of precision and recall evaluating the performance analysis for positive and negative classes

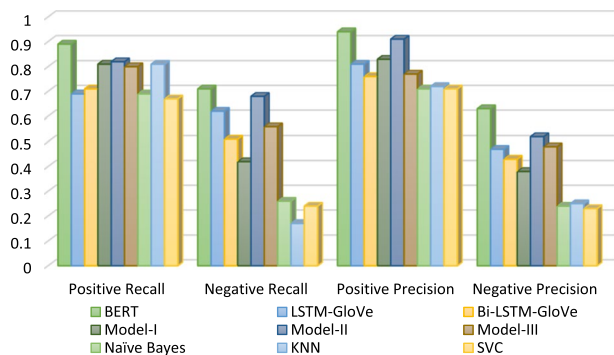


Table 5 Illustrations of stratified cross-validation folds w.r.t precision, recall, accuracy, and F1 score for Model-I and Model-II

| # of Folds | Model-I | | | | Model-II | | | |
|------------|-----------|--------|----------|----------|-----------|--------|----------|----------|
| | Precision | Recall | Accuracy | F1 score | Precision | Recall | Accuracy | F1 score |
| 20-fold | 0.8510 | 0.8310 | 0.8670 | 0.8408 | 0.8692 | 0.8563 | 0.8719 | 0.8627 |
| 15-fold | 0.8280 | 0.8080 | 0.8440 | 0.8178 | 0.8761 | 0.8660 | 0.8728 | 0.8710 |
| 10-fold | 0.8340 | 0.8140 | 0.8500 | 0.8238 | 0.8739 | 0.8641 | 0.8701 | 0.8689 |
| 5-fold | 0.8040 | 0.8010 | 0.8370 | 0.8024 | 0.8705 | 0.8609 | 0.8682 | 0.8656 |

used for four stratified fold cross-validation, demonstrated in Table 5. The degree of class imbalance from the generated corpus is treated as a challenge, specifically for the small number of negatives. However, the small class does appear less among the raw text, and our proposed model here can classify the real examples. It can be observed from the table that the classifier Model-II made a comparatively high accuracy with a 0.8719 value on the 20-folds, leading to high precision and recall for 20-fold. Whenever the precision and the recall are high, then the F1 score of 0.8682 will also be high as compared to Model-I for 20-fold. Among all the folds, the performance of the greater number of folds is higher than the smaller number of folds.

Figure 9 (subfigures a and b) illustrates the performance scores of Model-I and Model-II for various stratified fold variations. It becomes clear that tenfold and 20-fold give good performance among the other fold sizes for Model-I and Model-II, respectively. Accuracy and F1 score, in particular, are greater than the rest.

The experimental results achieved through both the DL and ML approaches in terms of four performance metrics are shown in Table 6. These approaches have been further categorized into three distinct subgroups for a comprehensive analysis. The first subgroup is centered on BERT, LSTM-GloVe, and Bi-LSTM-GloVe models, which yielded noteworthy results. Notably, this subgroup achieved a maximum accuracy rate of 91.50% and an F1 score of 91.00%. Moving on to the second subgroup, comprising Model-I, Model-II, and Model-III, it can be observed that Model-II emerged as the better performer among others. Remarkably, it attained an accuracy rate of 87.30% and an F1-Score of 84.69%, showcasing its effectiveness in the given context. Lastly, the third subgroup was characterized by

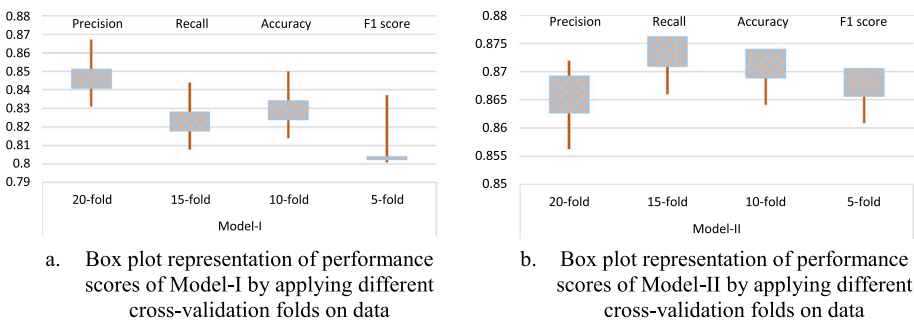
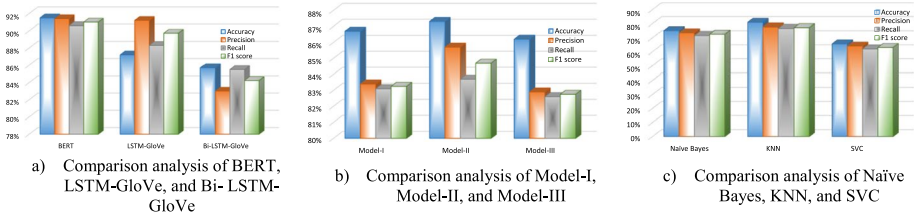


Fig. 9 Implementing a variety of cross-validation folds to the data and displaying the results as a box plot provided performance scores for hybrid models

Table 6 Performance evaluation of proposed DL and ML classifiers in terms of four metrics

| Metrics | BERT | LSTM-GloVe | Bi-LSTM-GloVe | Model-I | Model-II | Model-III | Naïve Bayes | KNN | SVC |
|-----------|--------------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Accuracy | 91.50 | 87.20 | 85.70 | 86.70 | 87.30 | 86.20 | 75.10 | 80.90 | 65.60 |
| Precision | 91.40 | 91.20 | 83.00 | 83.40 | 85.70 | 82.90 | 73.50 | 77.60 | 64.00 |
| Recall | 90.60 | 88.30 | 85.50 | 83.10 | 83.70 | 82.60 | 71.50 | 76.60 | 62.00 |
| F1 score | 91.00 | 89.73 | 84.23 | 83.25 | 84.69 | 82.75 | 72.49 | 77.09 | 62.98 |

**Fig. 10** Bar charts (a, b, c) representation of proposed BERT, DL, and baseline models w.r.t accuracy and F1 score

utilizing traditional ML techniques. However, this subgroup did not achieve as high accuracy rates as the DL-based subgroups. In this category, the KNN algorithm stood out with a maximum accuracy rate of 80.90% and an F1-Score of 77.09%. Figure 10 (subfigures a, b, and c) represents the accuracy and F1 score of all the proposed models, highlighting the model performances in terms of best fitting. These findings illustrate the varying outcomes among the approaches employed, with the pre-trained BERT-based model showing promise in achieving higher accuracy and F1 scores. At the same time, the ML-based techniques, which are less accurate, still offer valuable insights into the data.

Precision-Recall curve (PRC) is an automatic metric for evaluating the binary classifiers for imbalanced data in which precision is comprised of positive predicted values and true values, whether a recall measures the positive label example getting positive predicted results. To compare the different classifiers based on PRC, an AUPRC (area under the precision-recall curve) is the most reasonable measure to use. Therefore, the perfect PRC curves pass through the upper right corner and higher the area under the curve. For the generated corpus, the pre-trained BERT model demonstrates the perfect curve with a higher value of AUC of approximately 0.93 with an accuracy of 91.50% and an F1 score of 91.00%. PRCs are recommended for situations where there exists a significant class imbalance, ranging from moderate to substantial. This score (Table 6) representation proves especially effective in assessing classifier performance in scenarios where one class vastly outweighs the other.

5 Discussion

With the explosive growth of research in science, a greater number of academic publications have arisen. At the same time, subjective analysis is one of the finest ways to examine publications. It is employed in most computer science conferences and publications to

assist researchers in determining whether an article has a positive contribution or provides recommendations for future research. The main challenge in the NLP domain is developing an algorithm that can understand the hierarchical structure of sentences within a text and efficiently classify the contextual meaning of sentences. Our challenge was dealing with inadequate training information in the form of international papers. As a result, we developed a pre-trained BERT-based model that was fine-tuned on the human-annotated corpus. The BERT model may be employed in a variety of text-mining applications with relatively minor architectural changes [33]. The suggested BERT model outperformed the other models in both subjectivity analysis and classification. Data imbalance became an issue since the screening process only filtered out sparse data from a huge amount of excluded data. As a result, we employed stratified cross-validation of a generated corpus in which Model-II showed the performance in the form of precision, recall, and F1 score metrics.

In our research, we employed a comprehensive approach, encompassing both conventional and deep learning techniques, to conduct an accurate evaluation of performance on datasets generated from academic journal articles. Our methodology demonstrated notable efficiency gains compared to the baseline models. A new approach incorporating pre-trained word embeddings into a CNN-LSTM model yielded better results in terms of accuracy and F1 score compared to the established methods. Furthermore, our proposed techniques utilized fewer parameters, resulting in reduced memory consumption and enhanced efficiency, particularly in the convolution layer. Furthermore, for the experimental results, the raw data has been taken from NLP international conferences: ACL, NAACL, EMNLP, and CoNLL. The overall dataset is tested by employing NLP subjective analyzers (Textblob, Vader, and Flair). The primary drawback of a rule-based method of sentiment [8] evaluation is that it is only concerned with specific phrases and entirely neglects the context in which it is expressed. Several seed keywords are used to find the positive and negative labels, as seen in the italic font in Table 2, by manually assigning (positive and negative) labels. Also, cross-check the Flair-based labeling of abstracts with human-assigned as shown in the form of scatter plots for corpus in Fig. 6. In addition, we investigate whether DL and ML classification algorithms do better on human-annotated corpus classifying scientific research abstracts into specified discipline categories. To compare their performance against the ML algorithm such as NB, KNN, and SVC.

The proposed DL models integrated with pre-trained word embeddings trained on diverse training sets are also more trustworthy than traditional classifiers, which means that different DL classifiers are more consistent than different ML classifiers in giving the same categories to any given abstract. The initial subgroup is focused on BERT, LSTM-GloVe, and Bi-LSTM-GloVe models, which demonstrated notable performance. Specifically, this subgroup achieved a peak accuracy of 91.50% and an F1 score of 91.00%. Shifting to the second subgroup encompassing Model-I, Model-II, and Model-III, Model-II emerged as the better performer. Impressively, it achieved an accuracy of 87.30% and an F1-Score of 84.69%, highlighting its effectiveness in the specified context. Finally, the third subgroup employed conventional techniques. In this category, the KNN algorithm stood out with a maximum accuracy rate of 80.90% and an F1-Score of 77.09%. Overall, BERT outperforms all the classifiers on the human-annotated academic articles that have the perfect curve with a higher AUPRC of 0.93. A group of time series models (RNNs) remains a powerful tool for analyzing subjectivity and is frequently utilized, especially for NLP tasks. This dataset can be enhanced and evaluated through more advanced algorithms, such as ViT transformers, which utilize transformer architectures and have grown in popularity in recent years for subjective analysis of applications.

6 Conclusion

This study shows that the BERT-based language model as a subjectivity classifier can increase contribution and previous work sentence accuracy. The suggested models outperformed the present state-of-the-art models on human-annotated text in the corpus, consistently achieving excellent performance. The first phase is to analyze the human-annotated text with the Flair-based for subjective-based analysis of article text, which can also assist future research on the metaverse. The second phase is to employ the pre-trained GloVe and Word2Vec word embeddings to represent the text because of its complexities efficiently, that is, integrating with hybrid model CNN and LSTM for subjectivity classification of CS and PW sentences in academic articles. The performance of the five classification models is evaluated and contrasted with baseline methods in light of their experimental results. Model II, in general, enhances performance by integrating CNN and LSTM networks with an embedding layer. However, BERT has shown better content classification results with accuracy and F1 scores of 91.50% and 91.00%, respectively, as compared to the conventional classifiers. It can be concluded that a sufficient number of papers with subjectivity framing (emphasizing new contributions) abstracts can be published in proceedings conferences, which may encourage young researchers concerning future research to explore the latest methodologies in various applications. In the future, such abstracts will be studied on a yearly basis to evaluate the publication trend of such positively framed abstracts. The dataset will also be extended to integrate the understanding of the issues and implications of the semantic metaverse and natural language processing.

Acknowledgements We want to extend our sincere gratitude to the following individuals for their valuable contributions to this article: Conceptualization; Methodology; Data Curation; Writing—original draft; Resources: Atif Mehmood Formal analysis and investigation; Resources; Writing—original draft preparation: Farah Shahid Formal analysis, language editing, and investigation; Writing—review and editing; Validation: Rizwan Khan; Mostafa M. Ibrahim Resources; Visualization: Shahzad Ahmed Funding acquisition; Visualization; Supervision: Zhonglong Zheng.

Author contributions Conceptualization; Methodology; Data Curation; Writing—original draft; Resources: Atif Mehmood; Formal analysis and investigation; Resources; Writing—original draft preparation: Farah Shahid; Formal analysis and investigation; Writing—review and editing; Validation: Rizwan Khan; Mostafa M. Ibrahim; Resources; Visualization: Shahzad Ahmed; Funding acquisition; Visualization; Supervision: Zhonglong Zheng.

Funding This work was funded by the National Natural Science Foundation of China (NSFC62272419), U22A20102, the Natural Science Foundation of Zhejiang Province (ZJNSFLZ22F020010), and the Zhejiang Normal University Research Fund (ZC304022915), and research work was partially funded by the Zhejiang Normal University research funds (YS304023947 and YS304023948).

Data availability The data supporting the findings of this study are available upon reasonable request. Please get in touch with the corresponding author for access to the data.

Declarations

Competing interest We declare that this manuscript is original, has not been published before, and is not currently being considered for publication elsewhere.

Conflicts of interest The authors declare no conflict of interest.

References

1. Dehkharghani R, Saygin Y, Yanikoglu B, Oflazer K (2016) SentiTurkNet: a Turkish polarity lexicon for sentiment analysis. *Lang Resour Eval* 50(3):667–685. <https://doi.org/10.1007/s10579-015-9307-6>


2. Thangavel P, Lourdasamy R (2023) A lexicon-based approach for sentiment analysis of multi-modal content in tweets. *Multimed Tools Appl* 82(16):24203–24226. <https://doi.org/10.1007/s11042-023-14411-3>
3. Agarwal B, Mittal N, Bansal P, Garg S (2015) Sentiment analysis using common-sense and context information. *Comput Intell Neurosci* 2015:1–9. <https://doi.org/10.1155/2015/715730>
4. Zhu X, Zhu Y, Zhang L, Chen Y (2023) A BERT-based multi-semantic learning model with aspect-aware enhancement for aspect polarity classification. *Appl Intell* 53(4):4609–4623. <https://doi.org/10.1007/s10489-022-03702-1>
5. Gao Z, Feng A, Song X, Wu X (2019) Target-dependent sentiment classification with BERT. *IEEE Access* 7:154290–154299. <https://doi.org/10.1109/ACCESS.2019.2946594>
6. Ullah A, Khan SN, Nawi NM (2023) Review on sentiment analysis for text classification techniques from 2010 to 2021. *Multimed Tools Appl* 82(6):8137–8193. <https://doi.org/10.1007/s11042-022-14112-3>
7. Ittoo A, Nguyen LM, Van Den Bosch A (2016) Text analytics in industry: challenges, desiderata and trends. *Comput Ind* 78:96–107. <https://doi.org/10.1016/j.compind.2015.12.001>
8. Ravi K, Ravi V (2015) Knowledge-based systems a survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Syst* 89:14–46. <https://doi.org/10.1016/j.knosys.2015.06.015>
9. Nguyen D et al (2020) How we do things with words: analyzing text as social and cultural data. *Front Artif Intell* 3(August):1–14. <https://doi.org/10.3389/frai.2020.00062>
10. Bhavitha BK, Rodrigues AP, Chiplunkar NN (2017) Comparative study of machine learning techniques in sentimental analysis. *Proc Int Conf Inven Commun Comput Technol ICICCT 2017*, no. Iccict, pp 216–221. <https://doi.org/10.1109/ICICCT.2017.7975191>
11. Lee SW, Jiang G, Kong HY, Liu C (2021) A difference of multimedia consumer's rating and review through sentiment analysis. *Multimed Tools Appl* 80(26–27):34625–34642. <https://doi.org/10.1007/s11042-020-08820-x>
12. Cai Y et al (2019) A hybrid model for opinion mining based on domain sentiment dictionary. *Int J Mach Learn Cybern* 10(8):2131–2142. <https://doi.org/10.1007/s13042-017-0757-6>
13. Appel O, Chiclana F, Carter J, Fujita H (2016) A hybrid approach to the sentiment analysis problem at the sentence level. *Knowledge-Based Syst* 108:110–124. <https://doi.org/10.1016/j.knosys.2016.05.040>
14. Zhu C, Yi B, Luo L (2024) Base on contextual phrases with cross-correlation attention for aspect-level sentiment analysis. *Expert Syst Appl* 241(September 2023):122683. <https://doi.org/10.1016/j.eswa.2023.122683>
15. Saleh H, Alhothali A, Moria K (2023) Detection of hate speech using BERT and hate speech word embedding with deep model. *Appl Artif Intell* 37(1). <https://doi.org/10.1080/08839514.2023.2166719>
16. Muhammad A, Wiratunga N, Lothian R (2016) Contextual sentiment analysis for social media genres. *Knowledge-Based Syst* 108:92–101. <https://doi.org/10.1016/j.knosys.2016.05.032>
17. Fernández-Gavilanes M, Álvarez-López T, Juncal-Martínez J, Costa-Montenegro E, Javier González-Castaño F (2016) Unsupervised method for sentiment analysis in online texts. *Expert Syst Appl* 58:57–75. <https://doi.org/10.1016/j.eswa.2016.03.031>
18. Araque O, Corcuera-Platas I, Sánchez-Rada JF, Iglesias CA (2017) Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst Appl* 77:236–246. <https://doi.org/10.1016/j.eswa.2017.02.002>
19. Severyn A, Moschitti A (2015) UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In: *SemEval 2015 - 9th International Workshop on Semantic Evaluation*, co-located with the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2015 - Proceedings, pp 464–469. <https://doi.org/10.18653/v1/s15-2079>
20. Mewada A, Dewang RK (2023) SA-ASBA: a hybrid model for aspect-based sentiment analysis using synthetic attention in pre-trained language BERT model with extreme gradient boosting. *J Supercomput* 79(5):5516–5551. <https://doi.org/10.1007/s11227-022-04881-x>
21. Bansal B, Srivastava S (2018) Sentiment classification of online consumer reviews using word vector representations. *Procedia Comput Sci* 132:1147–1153. <https://doi.org/10.1016/j.procs.2018.05.029>
22. Tang D, Qin B, Liu T (2015) Document modeling with gated recurrent neural network for sentiment classification. *Conf Proc - EMNLP 2015 Conf Empir Methods Nat Lang Process*, no. September, pp 1422–1432. <https://doi.org/10.18653/v1/d15-1167>
23. Peters ME, Neumann M, Zettlemoyer L, Yih WT (2018) Dissecting contextual word embeddings: Architecture and representation. *Proc 2018 Conf Empir Methods Nat Lang Process EMNLP 2018*, pp 1499–1509. <https://doi.org/10.18653/v1/d18-1179>
24. Rao G, Huang W, Feng Z, Cong Q (2018) LSTM with sentence representations for document-level sentiment classification. *Neurocomputing* 308:49–57. <https://doi.org/10.1016/j.neucom.2018.04.045>

25. Li W, Zhu L, Shi Y, Guo K, Cambria E (2020) User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models. *Appl Soft Comput J* 94:106435. <https://doi.org/10.1016/j.asoc.2020.106435>
26. Giménez M, Palanca J, Botti V (2020) Semantic-based padding in convolutional neural networks for improving the performance in natural language processing. A case of study in sentiment analysis. *Neurocomputing* 378:315–323. <https://doi.org/10.1016/j.neucom.2019.08.096>
27. Gupta K, Ahmad A, Ghosal T, Ekbal A (2024) A BERT-based sequential deep neural architecture to identify contribution statements and extract phrases for triplets from scientific publications. *Int J Digit Libr*. <https://doi.org/10.1007/s00799-023-00393-y>
28. Jain PK, Quamer W, Saravanan V, Pamula R (2023) Employing BERT-DCNN with sentic knowledge base for social media sentiment analysis. *J Ambient Intell Humaniz Comput* 14(8):10417–10429. <https://doi.org/10.1007/s12652-022-03698-z>
29. Zhang X, Wu Z, Liu K, Zhao Z, Wang J, Wu C (2023) Text sentiment classification based on BERT embedding and sliced multi-head self-attention Bi-GRU. *Sensors* 23(3):1481. <https://doi.org/10.3390/s23031481>
30. Liu S, Lee I (2021) Sequence encoding incorporated CNN model for Email document sentiment classification. *Appl Soft Comput* 102:107104. <https://doi.org/10.1016/j.asoc.2021.107104>
31. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. *Adv Neural Inf Process Syst* 4(January):3104–3112
32. Jozefowicz R, Zaremba W, Sutskever I (2015) An empirical exploration of Recurrent Network architectures. *32nd Int Conf Mach Learn ICML 2015* 3:2332–2340
33. Goularte FB, Martins BE da G, Carvalho PC da F, Won M (2024) SentPT: a customized solution for multi-genre sentiment analysis of Portuguese-language texts. *Expert Syst Appl* 245(22):123075. <https://doi.org/10.1016/j.eswa.2023.123075>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Atif Mehmood^{1,2}  · Farah Shahid^{1,2} · Rizwan Khan¹ · Shahzad Ahmed³ · Mostafa M. Ibrahim⁴ · Zhonglong Zheng¹

✉ Zhonglong Zheng
zhonglong@zjnu.edu.cn

¹ School of Computer Science and Technology, Zhejiang Normal University, Jinhua 321002, China

² Zhejiang Institute of Photoelectronics & Zhejiang Institute for Advanced Light Source, Zhejiang Normal University, Jinhua, Zhejiang 321004, China

³ Faculty of Information Technology, Beijing University of Technology, Beijing 100024, China

⁴ Department of Electrical Engineering, Faculty of Engineering, Minia University, Minia 61519, Egypt