computers

MDPI

Article

Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning

Nasrin Elhassan [1], Giuseppe Varone [2], Rami Ahmed [1], Mandar Gogate [3], Kia Dashtipour [3], Hani Almoamari [4], Mohammed A. El-Affendi [5], Bassam Naji Al-Tamimi [6], Faisal Albalwy [7,8] and Amir Hussain [3,*]

1 College of Computer Sciences and Information Technology, Sudan University of Science and Technology, Khartoum P.O. Box 407, Sudan; nisreenosmaneltom@gmail.com (N.E.); ramiscience@gmail.com (R.A.)
2 Department of Physical Therapy, Movement and Rehabilitation Science, Northeastern University, Boston, MA 02115, USA; g.varone@northeastern.edu
3 School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, UK; m.gogate@napier.ac.uk (M.G.); k.dashtipour@napier.ac.uk (K.D.)
4 Faculty of Computer and Information Systems, Islamic University of Madinah, Medina 42351, Saudi Arabia; hani.almoamari@iu.edu.sa
5 Department of Computer Science, College of Computer and Information Sciences, Prince Sultan University, Riyadh 12435, Saudi Arabia; affendi@psu.edu.sa
6 School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK; bassam.al-tamimi@bcu.ac.uk
7 Department of Computer Science, College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia; fbalwy@taibah.edu.sa
8 Division of Informatics, Imaging and Data Sciences, School of Health Sciences, Faculty of Biology, Medicine and Health, The University of Manchester, Stopford Building, Oxford Road, Manchester M13 9PL, UK
* Correspondence: a.hussain@napier.ac.uk

Abstract: Social media networks have grown exponentially over the last two decades, providing the opportunity for users of the internet to communicate and exchange ideas on a variety of topics. The outcome is that opinion mining plays a crucial role in analyzing user opinions and applying these to guide choices, making it one of the most popular areas of research in the field of natural language processing. Despite the fact that several languages, including English, have been the subjects of several studies, not much has been conducted in the area of the Arabic language. The morphological complexities and various dialects of the language make semantic analysis particularly challenging. Moreover, the lack of accurate pre-processing tools and limited resources are constraining factors. This novel study was motivated by the accomplishments of deep learning algorithms and word embeddings in the field of English sentiment analysis. Extensive experiments were conducted based on supervised machine learning in which word embeddings were exploited to determine the sentiment of Arabic reviews. Three deep learning algorithms, convolutional neural networks (CNNs), long short-term memory (LSTM), and a hybrid CNN-LSTM, were introduced. The models used features learned by word embeddings such as Word2Vec and fastText rather than hand-crafted features. The models were tested using two benchmark Arabic datasets: Hotel Arabic Reviews Dataset (HARD) for hotel reviews and Large-Scale Arabic Book Reviews (LARB) for book reviews, with different setups. Comparative experiments utilized the three models with two-word embeddings and different setups of the datasets. The main novelty of this study is to explore the effectiveness of using various word embeddings and different setups of benchmark datasets relating to balance, imbalance, and binary and multi-classification aspects. Findings showed that the best results were obtained in most cases when applying the fastText word embedding using the HARD 2-imbalance dataset for all three proposed models: CNN, LSTM, and CNN-LSTM. Further, the proposed CNN model outperformed the LSTM and CNN-LSTM models for the benchmark HARD dataset by achieving 94.69%, 94.63%, and 94.54% accuracy with fastText, respectively. Although the worst results were obtained for the LABR 3-imbalance dataset using both Word2Vec and FastText, they still outperformed other researchers' state-of-the-art outcomes applying the same dataset.

Keywords: Arabic Sentiment Analysis; Word2Vec; FastText; convolutional neural networks; long short-term memory; recurrent neural networks

Citation: Elhassan, N.; Varone, G.; Ahmed, R.; Gogate, M.; Dashtipour, K.; Almoamari, H.; El-Affendi, M.A.; Al-Tamimi, B.N.; Albalwy, F.; Hussain, A. Arabic Sentiment Analysis Based on Word Embeddings and Deep Learning. *Computers* **2023**, *12*, 126. https://doi.org/10.3390/computers12060126

Academic Editors: Lu Bai, Huiru Zheng and Zhibao Wang

Received: 16 April 2023
Revised: 24 May 2023
Accepted: 13 June 2023
Published: 19 June 2023

1. Introduction

Sentiment analysis (SA) is a highly popular and active field of study in natural language processing (NLP), also known as opinion mining. It can be defined as the process of extracting and analyzing the sentiment and polarity of a given piece of text [1]. It has numerous applications across various industries, including politics, business, and social media. Social media platforms such as Twitter and Facebook have gained significant importance in analyzing people's reviews on specific topics, products, services, or other subjects. In the business sector, sentiment analysis enables companies to collect reviews about their products and services that help them make decisions to enhance their offerings [2]. Similarly, sentiments can be used in politics to analyze people's reactions toward political events and actions and help in decision-making [3]. While there have been numerous studies utilizing both classical and deep learning approaches for the English language, limited work has been conducted to develop models for sentiment analysis in the Arabic language [4,5].

Sentiment analysis can be conducted at many different levels, such as document level, sentence level, and aspect level [6,7]. This study employs sentence-level sentiment analysis on two Arabic datasets to evaluate whether the review's polarity is positive, negative, or neutral. The limited research on the Arabic language is due to its ambiguity, morphological complexities, various dialects [8], the unavailability of accurate pre-processing tools, and the shortage of its dataset resources [9]. The two techniques that can be used for the SA are the lexicon and machine learning-based techniques. The lexicon- or corpus-based technique uses a dictionary that contains words and their corresponding polarities [10], whereas machine learning-based techniques require a massive labeled dataset with manual annotation [11]. Deep learning's (DL) success in a variety of fields, including computer vision, reveals its use in the field of natural language processing [12–14]. Because DL does not use feature engineering to learn continuous text representations from data, it outperforms traditional machine learning. With the use of various word embedding approaches, deep contextual properties of words are extracted in a lower-dimensional space [15]. For NLP, many DL approaches are used, such as convolutional neural networks (CNNs) [12,16] and long short-term memory (LSTM) [17]. Although such DL techniques are frequently applied to English corpora, Arabic Sentiment Analysis deep learning models are not getting as much attention [18–20].

Each similar word is mapped closer to the other in a continuous vector space where words are represented by word embedding. Embedding methods have recently outperformed traditional text-feature extraction methods such as bag of words. Similar terms can be used to categorize new words that are missing from training texts. There are many word embedding techniques, such as Word2Vec, FastText, Glove, and BERT [21].

This study intends to investigate three DL models, CNN, LSTM, and CNN-LSTM, for SA on a highly imbalanced and balanced Arabic dataset. In addition, the study demonstrates comparative evaluation using FastText/Skip-Gram (SG) and Word2Vec/Continuous Bag of Words (CBOW) embedding models as input to the models to be investigated. This shows the effect of the embedding models and the datasets with the proposed models. This research is the first to thoroughly evaluate both the Hotel Arabic Reviews Dataset (HARD) [22] and Large-Scale Arabic Book Reviews (LABR) [23] datasets, which constitute some of the largest publicly available Arabic review corpora with different setups using three handcrafted DLs.

The following parts of this article are structured as follows: Section 2 discusses the literature review of Arabic Sentiment Analysis using word embeddings and deep learning techniques. Section 3 outlines the key components of the system and its proposed architecture. Section 4 outlines the experimental parameters and measurement measures. Section 5 represents the results and discusses the main findings. Finally, Section 6 delivers a succinct

conclusion based on the research conducted and the findings acquired to be able to present some plans for future studies.

2. Related Works

Most of the important approaches used for conventional and deep learning techniques are for the English language. Therefore, this research focuses solely on the analysis of Arabic sentiment with DL techniques. This section reviews the literature pertaining to the most recent and competitive sentiment analysis using DL models for Arabic and other languages. Moreover, the section discusses the DL models for Arabic in which the HARD and LABR datasets are used. The key objective of this part is to highlight the top DLs and word embeddings used for different languages according to their accuracy.

2.1. Arabic Language

Ombabi et al. [4] developed a novel hybrid deep learning model for Arabic Sentiment Analysis using CNN, two LSTM layers, and SVM on a multi-domain corpus, which is supported by FastText word embedding. The results showed that the proposed model outperformed a number of the previous studies, with up to 90.75% improvements. However, this study focused on binary classification with a balanced dataset and one-word embedding. On the other hand, Heikal et al. [24] developed an ensemble model that combined CNN and LSTM and used a pre-trained word vector representation, AraVec, for Arabic tweets. The model outperformed the state-of-the-art deep learning models with 65.05% accuracy on the Arabic Sentiment Tweets Dataset (ASTD). Their studies were obtained with a small amount of data by using one-word embedding.

Another study by Alahmary et al. [25] applied two DLs to classify sentiments in Saudi Arabia's dialect texts by using LSTM and bidirectional LSTM (Bi-LSTM). In this study, the two deep learning techniques were used together with the popular support vector machine (SVM) on a collected dataset of 32,063 tweets. Their findings stated that Bi-LSTM surpassed other models by 94% compared to LSTM (92%), and SVM (86.4%). Additionally, Baly et al. [26] evaluated the DL advances, namely the recursive neural tensor networks (RNTN), as a case study of morphologically rich languages (MRL). Despite the fact that RNTN required the existence of a sentiment treebank, which only exists for the English language, this study contributed to the development of the first Arabic sentiment treebank (ARSENTB). According to the experimental findings, the proposed RNTN achieved roughly 8% better at the phrase level and 10% better at the sentence level than the one for English, as shown by the average F1 score. Additionally, it performed 7.6%, 3.2%, and 1.6% better than popular classifiers such as support vector machines, recursive auto encoders, and LSTM. The best accuracy obtained by RNTN when applying stems was 81%.

Al-Sallab et al. [27] developed a DL framework in which four different architectures were proposed. Three were based on deep belief networks and deep-auto encoders, which are based on bags of words (BOW). This approach was recently utilized to create the Arabic Sentiment Lexicon along with the other standard lexical properties. In order to solve the previous three models' lack of context, the fourth model was offered. The experiments were conducted by employing the Linguistic Data Consortium Arabic Tree Bank dataset. The results showed that the fourth model was superior to the previous studies by around 9% on average F1, and the accuracy was about 74.3%. However, this study did not use the pre-trained model for vector representation or enough corpora. In a subsequent paper [28], the same scholars developed a recursive deep learning model for opinion mining in Arabic (AROMA) to address several challenges and limitations that arise when applying the RAE model [27]. AROMA was tested on three Arabic corpora. The first corpus consisted of newswire articles written in MSA; the second corpus was a set of tweets; and by utilizing topic modeling, the third corpus was created by QALB by extracting internet comments. Results revealed that AROMA surpasses the previous studies as well as the standard recursive auto encoder (RAE). In order to improve the performance of deep learning models for opinion mining, this work did not explore the

whole orthographic and morphological space in Arabic, including different levels of surface representation and morphological abstraction.

Another study conducted by Al-Surayyi et al. [29] developed different structures of DL techniques such as recurrent neural networks (RNNs) with LSTM, RNNs with Bi-LSTM, and CNNs. The experiments were conducted by using restaurant reviews from Yelp for binary and multi-class sentiment classification. The results revealed that Bi-LSTM achieved a better accuracy score compared to others, achieving 95.76% and 64.03% for binary and multi-classification, respectively. Nevertheless, no empirical analysis (such as word embedding dimensions, number of hidden units, activation functions, and cross-validation) was conducted on the hyperparameters' effects. In this study, various feature extraction techniques, such as BOW and TF-IDF, were not used.

Interestingly, Alayba et al. [9] investigated the advantages of combining CNNs and LSTMs for improving Arabic Sentiment Analysis accuracy on various datasets. Results showed improvement in their Arabic Health Services (AHS) dataset to reach 94.2% and 95.7% for the Main-AHS dataset and Sub-AHS dataset, respectively. This study used character level to help extend the number of features for each tweet. However, no pre-trained word embeddings were used. The outcomes demonstrated that the character level was not the model's optimal choice.

Authors Alayba et al. [30] exhibited their Twitter collection of user reviews of health-care services. They carried out a number of assessments utilizing DNN and CNN, among other machine learning methods. The DL techniques experienced encouraging outcomes, roughly between 85% and 91%. However, the linear support vector machine outperformed other classifiers, achieving 91.87% accuracy. This study did not use different feature extractions and did not cover the imbalanced dataset. Instead, it focused on increasing accuracy by enhancing the generalization of the model. On the other hand, Al-Azani et al. [31] proposed the CNN and LSTM techniques that use two different Word2Vec-based techniques: CBOW and skip-gram features on Arabic microblogs. The LSTM model showed the best results compared to CNN, with 87.27% accuracy on the ArTwitter dataset. However, to have better results, a large corpus was required in this study. In [32], Al-Azani et al. emphasized the importance of sentiment analysis of dialectical tweets from imbalanced datasets. They used Word2Vec for word embedding and over-sampled the minority class for the imbalance dataset by adding synthetic samples using the SMOTE technique.

In related research [32], based on emoji-extracted features, the researchers applied two types of RNN: LSTM and GRU deep learning, both unidirectional and bidirectional, for Arabic sentiment in Arabic microblogs. Experimental outcomes showed that the Bi-GRU model was superior to other models by achieving 78.71% accuracy.

In a different study, LSTM was used on the AraSen Corpus by Al-Laith et al. [33]. In order to guarantee the advancement of the Arabic sentiment classification, the suggested system was assessed against two external benchmark datasets. The outcome of the experiment showed that the developed framework outperformed the existing systems by achieving accuracies of 87.4% and 85.2% for SemEval 2017 and ASTD, respectively, for binary classification. For three-way classification, the model achieved 69.4% and 68.1% for the same datasets. However, the study did not use dialects or different pre-trained vector representations.

Oussous et al. [34] proposed a framework for Arabic Sentiment Analysis (ASA) by exploiting two DL models: CNN and LSTM models. The outcomes of the experiments demonstrated that DL models outperformed more traditional techniques such as support vector machines, naive Bayes classifiers, and maximum entropy. CNN and LSTM achieved an accuracy of 92.1% and 90%, respectively. Although the study investigated several aspects of data representation and their effects on the classifier's accuracy (such as stop word removal, normalization, and stemming), it did not investigate the effects of classifiers' accuracy to find the best parameters for LSTM and CNN.

Interestingly, Omara et al. [5] applied a novel application that uses two deep CNNs for Arabic Sentiment Analysis by using only character-level features. They used large-scale

datasets from different domains in different Arabic forms (modern standard and dialectal). Experiments were conducted for different machine learning algorithms such as logistic regression (LR), SVM, and naïve Bayes (NB). Findings revealed that CNN had the best accuracy, coming in at 94.3%.

In their study, Dahou et al. [35] presented a framework for CNN based on the differential evolution (DE) method. The experiments generated two DE-CNN models that were evaluated. The outcomes demonstrated that the proposed approaches outperformed the state-of-the-art models and had higher accuracy and a lower computational cost. For instance, for the ArTwitter dataset, DE-CNN with five cross-validations obtained an accuracy of 93.28% in comparison to a previous study that achieved 87.27% on the same dataset that combined LSTM. However, the study found that training a deep neural network usually relies on randomness to perform better, and that may affect the stability and repeatability of the obtained results on different evaluation techniques.

A different study by Altaher [36] applied a hybrid approach based on deep learning with feature weighting algorithms, information gain, and chi-square to analyze the sentiment of Arabic tweets. The research's findings revealed that the proposed approach surpassed other SVMs, DTs, and NNs in terms of accuracy and precision, achieving 90% and 93.7%, respectively. However, the semantics of the text in the Arabic tweets were not considered for this study.

This part sheds light on previous studies in which HARD and LABR datasets with deep learning techniques were used. As far as the authors are aware, few attempts have been made using various Arabic datasets with different setups, including binary and multi-classification for balance and imbalance datasets. Most of the previous studies that used HARD and LABR used classical machine learning techniques instead of deep learning.

For the HARD dataset, many studies were conducted based on classical machine learning [37,38]. For instance, a triple combination of N-gram features with PCA and LDA was used for feature selection [37]. Furthermore, several supervised classifiers were tested, such as DT, RF, bagging, NN, NB, boosting, and logistic regression (LR). The experiments were conducted for binary and multi-class classifications. As for deep learning, a study by Muaad et al. [39] developed a novel deep learning Arabic text computer-aided recognition (ArCAR) for character level instead of sentence level. The proposed model was based on CNN with six layers and two fully connected layers. The experiments were conducted on various datasets, including the HARD binary classification with balanced data and three classifications with imbalanced data. The accuracy for the two datasets was 93.58% and 93.23%, respectively.

A study by Mhamed et al. [40] developed a five-layer CNN architecture with an innovative mean max average (MMA) pooling layer to extract the best features. Three datasets, the 2-Class Sudanese Sentiment Dataset (SudSenti2), the 3-Class Sudanese Sentiment Dataset (SudSenti3), and HARD, were tested for binary classification with imbalanced data. The experiments were conducted for different deep learning models and included the baseline CNN, RNN, CNN/LSTM, and the proposed model. The developed model surpassed other deep learning models by obtaining 90.01% accuracy for the HARD dataset.

A comparison was obtained between the shallow learning and DLs models for Arabic Sentiment Analysis tested for two datasets, including HARD 2-imbalance conducted by Nassif et al. [41]. The DL models used were CNN, GRU, Bi-LSTM, and different hybrid techniques. The study used an araBERT transformer for the embedding layer. The best results were obtained by the hybrid Bi-LSTM+CNN which achieved 94.2% accuracy for the HARD dataset.

For the LABR dataset, the literature showed that many studies focused on classical machine learning, but only a few worked [42–47] on deep learning. A study was conducted using deep learning techniques [48], in which three models were implemented: CNN, LSTM, and a hybrid CNN-LSTM model. The experiments were conducted using the original LABR dataset with five classes of imbalance, in which the hybrid model achieved the best results with 86.88% accuracy.

Another study by Abu Kwaik et al. [49] applied a hybrid CNN with LSTM and Aravec for the first layer, which are pre-trained word embeddings in Arabic. The binary balanced and unbalanced subsets of LABR, together with the three-way classification subsets, had been employed in the evaluations. The accuracies obtained for the three subsets were 81.14%, 80.2%, and 66.42%, respectively.

In another study, the authors developed a model applying the typical 1-D CNN model and tested it with the original LABR subset with five classes imbalanced [50]. It examined the Word2Vec-CBOW, Word2Vec-SG, and fastText Arabic word embedding models. The best results were obtained when fastText word embedding was used, accomplishing 87.73% accuracy. In an interesting study [51], integration between a simple recurrent unit and an attention mechanism that mainly concentrates on an input text was done. DL models have been surpassed by the demonstrated model, which achieved 94.53% accuracy with two classes of the imbalanced dataset. LSTM was used in a remarkable work by Al-Bayati et al. [3]. The model was trained using a deep neural network, and word embedding was used as the initial stage of hidden layer feature extraction. The experiments were carried out on two classes of imbalanced data. An accuracy of about 82% was achieved by using this model.

2.2. Other Languages

Naqvi et al. [52] presented an approach for determining polarity for Urdu datasets collected from different resources and domains by observing various deep learning methods combined with different word embeddings. Attention-based bi-directional LSTM (BiLSTM-ATT) surpassed other DL models by achieving 77.9% accuracy. Dashtipour et al. [10] presented a novel deep learning framework for Persian movie reviews using two deep learning algorithms, CNN, and LSTM. Simulation findings revealed that LSTM was superior to other DL models by accomplishing an accuracy of up to 95.61%.

A comparison study [53] for the Lithuanian language was carried out to compare traditional machine learning (support vector machines and nave Bayes) and deep learning methods (LSTM and CNN). The study used two-word embedding (Word2Vec/CBOW) with negative sampling and FastText to solve the sentiment analysis task. The experiments were performed on the Lithuanian Internet comment dataset. The results obtained showed that classical machine learning methods were superior compared with DL techniques. Traditional machine learning approaches obtained 73.5% accuracy, compared to 70.6% accuracy obtained by deep learning methods. The study used the morphological analyzer-lemmatizer tool, Lemuoklis, which could not process non-normative words.

A study [54] analyzed the tweets using hybrid deep learning (HDL) techniques on the SemEVAL-2017 dataset. The outcomes revealed the efficiency of ANN (achieving 92% accuracy) as compared to other classifiers such as random forest, naive Bayes, and decision tree classifiers. Another study [55] conducted different approaches for sentiment analysis in Albanian, including traditional machine learning techniques such as logistic regression, naïve Bayes, decision tree, multi-layer perceptron, SVM networks, and random forest. Sentiment lexicon-based techniques and hybrid approaches from CNN and LSTM deep learning approaches were also used. The research was conducted on a large tweet dataset in Albanian. For data pre-processing, the study used several feature extraction methods, including bag-of-words, TF-IDF, Word2Vec, and glove. Experiments showed that LSTM with glove as feature extraction outperformed other models with 79.2% accuracy.

A list of the deep learning-related research presented in this study is presented in Table 1.

Although Table 1 shows the most recent research work that used various deep learning techniques for sentiment analysis, only a few studies investigated the impact of using different setups of the same dataset, such as binary or multiclassification and balance or imbalance. In this study, different datasets with different setups were applied to different models. The study also investigated the impact of using various word embeddings on the model's accuracy.

Table 1. Summary of deep learning-related works.

Ref.	Extractor	Deep Learning	Dataset	Accuracy (%)
[4]	FastText	CNN/LSTM	Multi-domain sentiment corpus	90.75%
[24]	AraVec	CNN/LSTM	ASTD	65.05%
[25]	Word2Vec	LSTM/Bi-LSTM	Arabic tweets	94%
Baly et al. (2017) [26]	ARSENTB	RNTN	Arabic Corpus	81%
[27]	BOW	DNN/DBN/Deep Auto Encoders	LDC ATB	74.3%
[28]	ArSenL	The recursive deeplearning model	ATB/Tweets /QALB	86.5%/79.2% /76.9%
[29]	GloVe	LSTM/Bi-LSTM	Restaurant reviews	95.76% binary /64.03% multi
[9]	5-g	CNN LSTM	Main-AHS /Sub-AHS /Ar-Twitter /ASTD	94.2% /95.7% /88.1% /77.6%
[30]	Word2Vec	DNN CNN	Health Tweets	91.87%
[31]	Word2Vec/CBOW; Word2Vec/SG	CNN/LSTM	ASTD/ ArTwitter	81.63%/ 87.27%
[32]	Several types	LSTM/Bi-LSTM GRU/Bi-GRU	Various datasets	78.71%
[33]	FastText	LSTM	SemEval 2017; ASTD	87.4%/85.2%
[34]	Unigrams	CNN LSTM	Arabic tweets	92.1%/90%
[5]	TF TF-IDF	CNN	Constructed dataset	94.33%
[35]	Word2Vec	CNN	Various datasets	93.28%
[36]	TF-IDF	H_2O Deep Learning model	Arabic tweets	90%
		Language: Urdu		
Naqvi et al. (2021) [52]	SAMAR/ FastText/ coNLL	LSTM/BiLSTM-ATT/CNN/C-LSTM	Constructed dataset	77.9%
		Language: Persian		
Dashtipour et al. (2021) [10]	fastText	CNN/LSTM	Movies review	95.61%
		Language: Lithuanian		
Kapovciute et al. (2019) [53]	Word2Vec/ fastText	CNN/LSTM	Lithuanian Internet comments	70.6%
		Language: English		
Divyapushpalakshmi et al. (2021) [54]	semantic and syntactic functions	ANN	SemEVAL-2017	92.0%
		Language: Albanian		
Vasili et al. (2021) [55]	BOW/TF-ID/ Word2Vec/Glove	LSTM/CNN	Tweets in Albanian	79.2%

3. Materials and Methods

In this section, two word embedding models, Word2Vec and fastText, are discussed. Moreover, three DL techniques, CNN, LSTM, and CNN-LSTM, used in this study are presented.

In this section, the primary system parts are also represented. First, the Arabic sentences were passed through a pre-processing and cleaning phase to delete unwanted symbols and tokens. Then, in order to prepare the dataset to be fed into selected deep neural networks, the cleaned data was prepared for the word embedding step by converting texts to vectors by applying one of the word embedding methods, and finally for the training phase. Various machine learning models were trained: the CNN model, an RNN with LSTM layers model, and the CNN-LSTM hybrid model. The models classified the sentiment of the input sentence as either positive, negative, or neutral, if any. Figure 1 shows the proposed structure and specifics in the next sections.

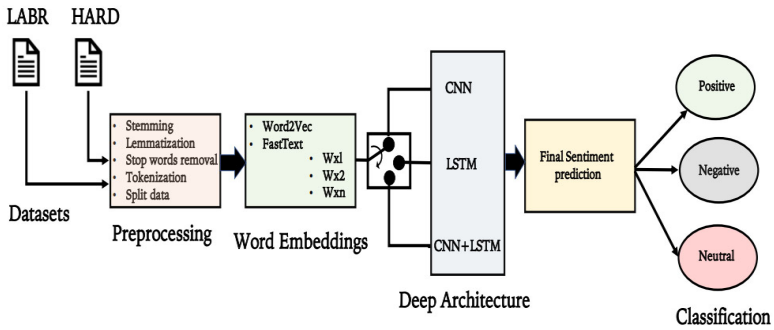

Figure 1. Pipeline: The Large-Scale Arabic Book Reviews (LABR) dataset and the Hotel Arabic-Reviews Dataset (HARD) were included in this study. Arabic sentences have been cleaned, normalized, and lemmatized. The cleaned Arabic sentences were entered into the word embeddings, Word2Vec, and fastText used in this study. To determine the polarity, the data vectors were fed into the deep architecture and sentiment classification. Finally, one of the three labels will be assigned to the polarity. whether positive, negative, or neutral, if any.

3.1. Dataset Details

In this study, two datasets were used in the experiments: the Large-Scale Arabic Book Reviews (LABR) dataset and the Hotel Arabic-Reviews Dataset (HARD) [22], with different setups including balance, imbalance, binary, and multi-classification datasets.

3.1.1. Large-Scale Arabic Book Reviews (LABR) Dataset

Aly et al. [23] manually harvested the largest sentiment analysis dataset for Arabic from the website Goodreads.com in March 2013. The dataset contained over 63,000 book reviews in Arabic, rated from 1 to 5. Some operations and data preparations were performed on the dataset, as presented in the data preprocessing and cleanings section.

3.1.2. Hotel Arabic-Reviews Dataset (HARD)

Elnagar et al. [22] manually built a labeled sentiment analysis dataset from the Booking.com website during June–July 2016. The initial dataset had 981,143 reviews in modern standard Arabic as well as dialectal Arabic, rated from 1 to 5. Each review had positive and negative parts that were combined into one review, resulting in 492,907 reviews. After cleaning, 373,772 reviews remained after reduction, which made up the full unbalanced HARD dataset. The balanced dataset consisted of 94,052 reviews, with 46,968 positive and 47,084 negative reviews. Table 2 shows the number of samples utilized in the present study for each dataset setup.

Table 2. Arabic Reviews Dataset versions.

Data Set	Positive	Negative	Neutral	Total
LABR/2classes/imbalance	42,832	8224	None	63,257
LABR/3classes/imbalance	42,832	8224	12,201	63,257
HARD/2classes/Balance	52,849	52,849	None	105,698
HARD/2classes/imbalance	276,387	52,849	None	409,562
HARD/3classes/imbalance	276,387	52,849	80,326	409,562

3.2. Data Pre-Processing and Cleaning

Datasets usually contain optional dross characters and symbols that may affect the accuracy and quality of the learning models. In this regard, a clean dataset is recommended in a controlled way to avoid missing important data and make the learning process more effective. In order to prepare the two datasets for the cleaning phase, the following procedures were carried out on them:

3.2.1. LABR Preparation

The LABR dataset was used to map ratings 1 and 2 to the negative class, ratings 4 and 5 to the positive class, and rating 3 to neutral. The numbers of negative, positive, and neutral reviews were 8224, 42,832, and 12,201, respectively, giving a total of 63,257 reviews. Another version of LABR with two classes was used by omitting Rating 3.

3.2.2. HARD Preparation

For hotel reviews, the balanced dataset was used with two classes that mapped ratings 1 and 2 to negative, ratings 4 and 5 were mapped to positive, and there were no neutral reviews. The experiments' dataset, which was employed, had a size of approximately 52,849 for each positive and negative class. Another version of hotel reviews used two imbalance classes, mapping ratings 1 and 2 to negative and ratings 4 and 5 to positive, with no neutral reviews. The last version had three imbalance classes that conducted the same operations for positive and negative and converted rating 3 to neutral. The data pre-processing and cleaning steps required for the two Arabic datasets are described below:

1. Remove the NAN values.
2. Removing non-required columns by excluding any columns that are not "sentiment" or "polarity".
3. Remove any duplicate entries.
4. Map any "positive", "negative", or "neutral" values into numbers.
5. Tokenize the text column into separate lists using sent_tokenise from NLTK (https://www.nltk.org/, (accessed on 1 January 2022)).
6. Clean the text column using the following steps:
   - Remove the English and Arabic punctuation, e.g., <>, _, (), *, &, . . . etc.
   - Perform Arabic character normalization, as in [56]. This step is widely used in Arabic NLP as it aims to unify the letters that can appear in different forms. e.g., convert any [إأآا] to "ا", "ى" to "ي", "ؤ" to "ء", "ئ" to "ء", "ة" to "ه" and "گ" to "ك"
   - Remove the diacritic "tashkeel"

   e.g.,
   ّ | # Tashdid الشدة, ُ | # Fatha الفتحة, ً | # Tanwin Fath تنوين بالفتحتين . . . etc.
   - Elongation removal: remove the repeating character to keep just one character. e.g.,جمييييييييل will convert to جميل [57]
   - Remove the stop words, e.g., أما, بعد, هل . . . etc.

- Remove the non-Arabic character by using regular expressions to filter out languages that use other alphabets.
- Remove any digit, such as 1234.

7. Convert the lists into tokens using word_tokenize from NLTK
8. Stemming the tokens using ISRIStemmer from NLTK, e.g., نظيفwill be نظف
9. Split the data into training and testing

3.3. Preparing Data for Word Embedding/Text Representation

For converting text into vectors, embedding methods have recently outperformed traditional text-feature extraction methods, such as bag of words, especially when using a deep learning model [58]. With word embedding, related words have similar encodings in an effective dense vector of floating-point value representation. The embedding parameters: some parameters are possible to specify as the vector's length, and some parameters are trainable as weights. In the training phase, weights are learned by the model and for the dense layer too. The resulting embedded vectors contain similar categories of words that are closer in proximity [3].

There are several models available for learning word embeddings from raw text, such as hot vector, Word2Vec developed by Google [59], Glove developed by Stanford [60], fastText developed by Facebook [61], and Bidirectional Encoder Representations from Transformers (BERT) developed by Google [62]. A word set of various lengths composed of each review. Each word in the review was transformed during the data preparation stage to its corresponding word embedding from a pre-trained distributed word representation. In this work, two popular neural models used for learning word embedding [63] were Word2Vec/CBOW and fastText/SG. Table 3 states the setup for both Word2Vec and fastText.

Table 3. Word embedding setup.

Word Embedding	Size	Min Count	Window	Workers	SG
Word2Vec	300	1	5	4	0
FastText	300	1	40	4	1

3.3.1. Word to Vector Embedding Model (Word2Vec)

Word2Vec is a neural network model that converts the semantic information for a given corpus by evaluating the cosine resemblance between the word vectors to understand the semantic similarity and categorize words as similar or dissimilar vectors. It is used in different applications, such as sentiment classification, named-entity recognition (NER), POS-tagging, and document analysis. Word2Vec has two techniques: skip gram (SG) and continuous bag of words (CBOW) [64]. In this study, CBOW was used after conducting exhaust experiments, which concluded that CBOW outperformed the SG. SG was trained to predict the context (surrounding words) of a given word, whereas CBOW was trained to predict the present word using its context (provided words) [65]. For this work, each review was represented as a 2D vector with n $\times$ 300 dimensions, where n is the number of words in the review and 300 is the length of the vector's dimension for each word. This was executed to ensure that all the reviews were of the same size, and each review's representation was padded by zeros to make its length consistent throughout the dataset. This study follows the same approach used for Twitter sentiment analysis with CNNs and LSTMs [66].

3.3.2. FastText Embedding Model

FastText is an efficient, fast, and open-source architecture developed by Facebook for text classification [67] in 2016. It operates on the Word2Vec and n-grams principles, utilizing the information in the sub-words. In fastText, the words are divided into subwords (n-grams) and then fed into the neural network for learning the embedding. As a result, words that do not exist in the vocabulary can be constructed using their constitutive

n-grams, allowing fastText to outperform Word2Vec and Glove in situations where a non-vocal word cannot be obtained after training [67]. FastText has two techniques: skip gram (SG) and continuous bag of words (CBOW). In this study, the experiments showed that SG yields better results and outperforms CBOW.

This study adopted the same approach as [17]. Each review was represented as a 2D vector with n × 300 dimensions, where n is the number of words in the review and 300 is the length of the vector's dimension for each word. This was performed to ensure that all reviews were of the same size, and each review's representation was padded by zeros to keep its length consistent throughout the dataset. Table 3 shows the word embedding setup.

3.4. Deep Learning Models

Implementing a proper model design for deep learning networks such as CNN and LSTM is like a black box because it depends on the problem and the corresponding datasets, which require different pre-processing steps and adjustments. Consequently, the performance is affected by the architecture [54,55]. Many structural hyperparameters (such as the number of filters, the depth of the number of convolutional and fully connected layers, and the number of units in fully connected layers) should be carefully tuned according to a given dataset. However, this is a challenging process since it is unpredictable how these hyperparameters interact to determine how well the developed model performs [68]. The handcrafted design is one methodical approach to developing CNN network architecture; however, it requires much human experience and work to manually adjust the CNN architecture's hyperparameters and use a variety of trial-and-error techniques [69].

The incoming part illustrates the implementation of the three proposed models using CNN, LSTM, and the hybrid CNN/LSTM. In this work, the design phase for all proposed models was handcrafted to explore more model generalization.

3.4.1. The Proposed Arabic Sentiment Analysis Using CNN

In this section, the implementation of the proposed Arabic Sentiment Analysis model using the CNN architecture has been illustrated. In the proposed CNN architecture, Figure 2 shows the parts of the two key stages of the proposed system, feature extraction and classification.

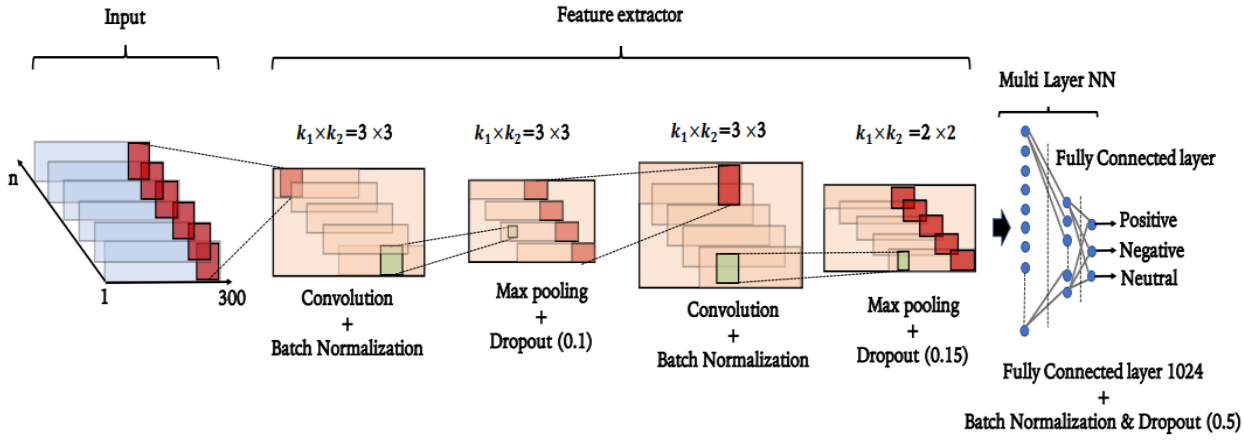

Figure 2. The proposed CNN architecture for Arabic Sentiment Analysis.

The CNN model generally starts with an embedding layer to represent words as dense vector representations. Then, after that, there are blocks of convolution, normalization, max-pooling, and dropout layers that follow each other. Finally, followed by a connected layer producing output results [70]. CNN is motivated by biological research on biological vision mechanisms [71], which necessitates less training data [72]. Initially, it was designed for image recognition. However, the CNN model has been effectively used in NLP tasks such as text classification [73]. It is a convolution-based feed-forward neural network. The

convolutional layer is utilized as a feature extraction layer that is composed of multiple filters to extract local feature vectors inside a multi-dimensional field [74]. Then, the output features from the convolutional layer are usually pooled or sub-sampled to smaller dimensions to reduce the computational complexity. Finally, the output is entirely represented by the last layer in CNN [75].

In this research, the CNN sequential model has been developed by using 18 building blocks, including one embedding layer as an input to the proposed model, one output layer, and two stacked one-dimensional (1D) convolutional layer blocks. The model also includes two fully connected hidden layers (FCLs) for text classification.

For the input layer, two-word embeddings were implemented based on the pre-trained Word2Vec and fastText models. The second layer was made of two one-dimensional (1D) convolution layers in which a padding option called "same" was used to keep all inputs in the same dimensions. Thirty-two feature maps with a kernel size of three were employed in the second convolutional layer as compared with 16 feature maps with a kernel size of three for the first convolutional layer. The ReLU activation function was used for both one-dimensional (1D) convolutional layers. The batch normalization layer following each convolutional layer was used to avoid the overfitting problem [69]. It reduced overfitting because of its regularization effect [76]. Moreover, the dimensionality of the feature maps produced by the convolution layer was minimized using the max pooling 1D layer. A pool size of three and two was used after the first and second convolutions, respectively. The dropout layer, following the previous layer, was used to reduce overfitting. It was configured randomly to remove 10% and 15% of the first and second convolution layers, respectively. The final layer was the dense layer, whereby two dense layers were used, one in the hidden layer with 1024 neurons and another in the output layer. The output layer's number of neurons was configured according to the dataset's number of classes (one for binary and three for multi-class). The binary classification for this last layer was performed using the sigmoid function, and the multi-classification was performed using the Softmax activation function.

3.4.2. The Proposed Arabic Sentiment Analysis Using LSTM

This paragraph demonstrates the implementation of the proposed Arabic Sentiment Analysis using the LSTM architecture. It generally starts with an embedding layer to reflect words as dense vector representations, followed by a feature extraction layer with one LSTM layer of 16 units. The final classification layers are two fully connected layers: the first one is in the hidden layer with 1024 neurons, followed by two normalization layers and a dropout layer. The output layer's number of neurons is configured according to the dataset's number of classes (one for binary and three for multi-class). The proposed LSTM architecture is shown in Figure 3, which demonstrates the basic parts of the proposed system's two key phases: the feature extraction phase and the classification phase.

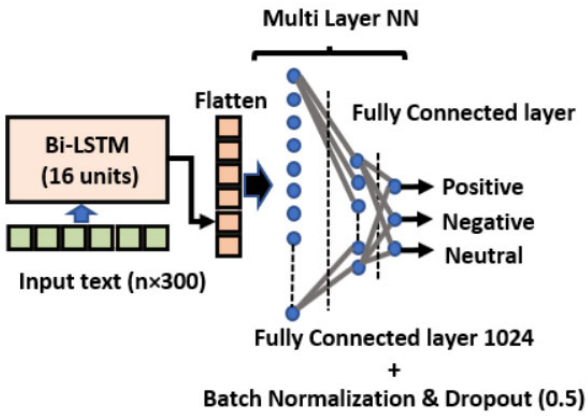

Figure 3. The proposed LSTM architecture for Arabic Sentiment Analysis.

3.4.3. The Proposed Arabic Sentiment Analysis Using CNN-LSTM

The proposed Arabic Sentiment Analysis using a hybrid CNN-LSTM architecture is based on the two previous models discussed in Sections 3.4.1 and 3.4.2. The first part was the CNN model, followed by the LSTM model. It started with an embedding layer to represent words as dense vector representations, followed by the feature extraction layers specified in the CNN model. This was followed by one LSTM model as described in the LSTM model, and finally, two connected layers to produce the output results. The proposed CNN-LSTM architecture is shown in Figure 4, which also depicts the components of the system's two primary phases: the feature extraction phase and the classification phase.

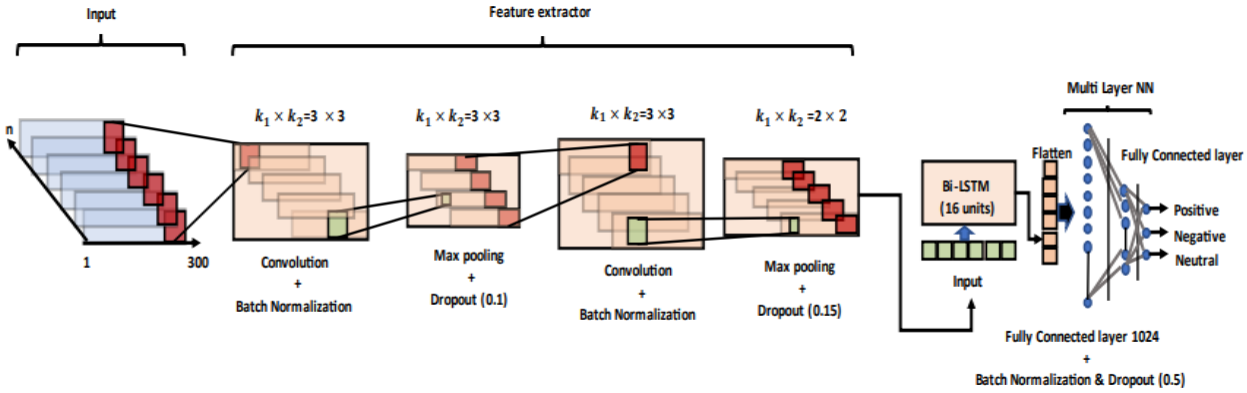

Figure 4. The proposed CNN-LSTM architecture for Arabic Sentiment Analysis.

The hyperparameters used in the basic models were selected after running several experiments using the trial-and-error method by experimenting with various hyperparameters. Exhaust experiments were conducted to select the best hyperparameter values; the hyperparameters that delivered the model's best outcome are shown in Table 4.

Table 4. Tab: The general hyperparameters of deep learning models.

Hyperparameter	Value
Batch size	128
Embedding size	300
Embedding model	Word2Vec/CBOW-FastText/SG
Embedding Trainable	True
Filters	16,32
Epochs	50
Kernel Size	[3]
Pool size	[2,3]
Verbose	1
Optimizer	RMSprop
Activation function	Softmax, Sigmoid
Dropout	0.25, 0.10, 0.15, 0.5
Validation split	0.2
Shuffle	True

4. Experimental Results

In this section, the experimental settings, configuration, and evaluation measurements that were measured in this study are illustrated.

4.1. Experimental Setup

The computer used for the experimental setup had the following features: a Windows 10 operating system, an Intel (R) Core (TM) i7-7700HQ processor clocked at 2.80 GHz, 16 GB of RAM, and an Nvidia GeForce (GTX 1050 Ti) graphics card with 4 GB of RAM that supported the CUDA version 7.1.4 parallel computing platform. Nvidia is an American

multinational technology company. Google's Tensorflow was utilized to conduct the research. It is an open-source framework for building machine learning models that utilizes Keras's higher-level API that is built on top of Tensorflow, their implementation, and deployment, in addition to the open-sourced DL library for Python.

4.2. Evaluation Criteria

To measure the performance of the proposed approaches, two datasets were used with different setups. The data was split into trained 80% and tested 20%. The vectorized training and test datasets were input into the CNN and LSTM classifiers. The parameters of the LSTM and CNN models are as follows: The word embedding dimensions were 300. The number of epochs, the batch sizes, and other parameters took various setups depending on the word embedding and the classifier model that was used. The ML classifiers were trained to determine the polarity of the reviews into either positive, negative, or neutral, if any. Equations (1)–(4) show the definitions of precision, recall, F-measure, and accuracy, respectively, that were implemented to evaluate how well the proposed approaches performed.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F\_measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

5. Results and Discussions

The two performance measurements, accuracy and precision, are considered the main factors in assessing the three models for all databases. Therefore, the accuracy measurement was used for our comparisons. Tables 5–7 show the results obtained by the three models for different versions of the used datasets. Moreover, Table 8 presents the outcomes obtained by other studies using the HARD and LABR datasets. To compare the accuracy attained by the models in this study with other studies that utilized the same dataset versions, a comparative study was carried out.

Table 5. The proposed CNN Experiment Results.

Data Sets	Word Embedding	Precision	Recall	F-Score	Accuracy
HARD/2Balance	Word2Vec	90.41%	90.34%	90.37%	90.36%
	FastText	90.35%	90.34%	90.34%	90.34%
HARD/2Imbalance	Word2Vec	92.43%	87.43%	89.86%	94.68%
	FastText	92.76%	87.08%	89.83%	94.69%
HARD/3Imbalance	Word2Vec	84.00%	75.78%	79.78%	86.37%
	FastText	84.24%	76.20%	80.02%	86.46%
LABR/2Imbalance	Word2Vec	80.08%	61.57%	69.61%	86.33%
	FastText	76.44%	63.33%	69.27%	86.17%
LABR/3Imbalance	Word2Vec	56.23%	43.71%	49.19%	69.72%
	FastText	58.29%	41.34%	48.38%	69.54%

Table 6. The proposed LSTM Experiment Results.

Data Sets	Word Embedding	Precision	Recall	F-Score	Accuracy
HARD/2Balance	Word2Vec	90.28%	90.20%	90.24%	90.20%
	FastText	90.24%	90.19%	90.22%	90.17%
HARD/2Imbalance	Word2Vec	92.86%	86.56%	89.60%	94.58%
	FastText	93.07%	86.58%	89.71%	94.63%
HARD/3Imbalance	Word2Vec	83.68%	76.24%	79.79%	86.33%
	FastText	84.44%	76.19%	80.10%	86.56%
LABR/2Imbalance	Word2Vec	77.75%	62.77%	69.46%	85.85%
	FastText	79.91%	61.55%	69.54%	86.36%
LABR/3Imbalance	Word2Vec	50.69%	41.12%	45.41%	69.35%
	FastText	58.70%	43.71%	50.11%	69.28%

Table 7. The proposed CNN/LSTM Experiment Results.

Data Sets	Word Embedding	Precision	Recall	F-Score	Accuracy
HARD/2Balance	Word2Vec	90.28%	90.24%	90.26%	90.23%
	FastText	90.47%	90.36%	90.42%	90.38%
HARD/2Imbalance	Word2Vec	92.53%	87.07%	89.72%	94.63%
	FastText	92.16%	86.93%	89.47%	94.54%
HARD/3Imbalance	Word2Vec	84.52%	76.09%	80.08%	86.61%
	FastText	83.82%	76.49%	79.99%	86.50%
LABR/2Imbalance	Word2Vec	78.21%	62.50%	69.48%	86.00%
	FastText	77.12%	62.35%	68.95%	86.05%
LABR/3Imbalance	Word2Vec	45.85%	39.84%	42.63%	68.64%
	FastText	54.09%	40.92%	46.59%	69.09%

Table 8. Performance evaluations compared with other approaches.

	2 Balanced		2 Imbalanced		3 Imbalanced	
Ref.	Study	Our Result	Study	Our Result	Study	Our Result
[39]	93.58%	90.38%	-	-	93.28%	86.61%
HARD [40]	-	-	90.01%	94.69%	-	-
[20]	-	-	94.2%	-	-	-
LABR [49]	81.14%		80.2%	86.36%	66.42%	69.71%
[3]	-	-	82%	-	-	-

5.1. The Proposed CNN Model Experiments Results

The outcomes of the CNN models are shown in Table 5 for the various dataset setups and the two-word embeddings. The results show that the best results were obtained for the HARD/2 imbalance dataset compared to other datasets by accomplishing 94.68% and 94.69% accuracy when using Word2Vec and fastText word embedding, respectively. The proposed model surpassed the existing best research [40] when using HARD 2-imbalance with a CNN model by accomplishing a 90.01% accuracy compared to our model, which achieved a 94.69% accuracy. However, the worst results were obtained for the LABR

3-imbalance, which achieved 69.71% and 69.54% accuracy when using Word2Vec and fastText, respectively. For this model, Word2Vec outperformed the fastText word embedding in most cases, as in the case of LABR 3-imbalance. However, in the case of HARD 2-imbalance, fastText outperformed Word2Vec. For LABR 3-imbalance, the proposed model outperformed the result of a study [49] that used a hybrid model that accomplished 66.42% accuracy as compared to the proposed model's 69.71% accuracy.

5.2. Experiment Results for the Proposed LSTM Model

Table 6 presents the results obtained for the LSTM model using various dataset setups and two-word embeddings. The outcomes demonstrated that the best outcomes were obtained for the HARD 2-imbalance dataset, which achieved accuracy of 94.58% and 94.63% for the Word2Vec and fastText word embeddings, respectively. It is possible to conclude that fastText outperformed Word2Vec. The worst results obtained for the LABR 3-imbalance dataset achieved 69.35% and 69.28% accuracy for Word2Vec and fastText, respectively. However, for this dataset, Word2Vec outperformed fastText. For this model, fastText word embedding outperformed Word2Vec in most cases. The proposed model outperformed the results obtained by Al-Bayati et al. [3] that used the LSTM model and achieved 82% accuracy for 2imbalance compared with this study's result of 86.36% accuracy for the same dataset.

5.3. The Proposed CNN-LSTM Hybrid Model Experimentation Results

Table 7 shows the summary of results for the hybrid model CNN-LSTM using various datasets and the two-word embedding. The best results were obtained for the HARD 2-imbalance dataset, which achieved accuracy of 94.63% and 94.54% for the Word2Vec and the fastText, respectively. In general, fastText outperformed Word2Vec in most cases. The results of this study outperformed the results obtained by Nassif et al. [41] for their hybrid Bi-LSTM+CNN model and HARD dataset by accomplishing 94.2% accuracy compared to this study's accuracy of 94.63%. The worst results were obtained for LABR 3-imbalance. However, fastText outperformed Word2Vec by accomplishing 69.09% accuracy. The model outperformed the hybrid model in [49] by achieving 86.36% and 69.71% accuracy on the LABR 2-imbalance and LABR/3imbalance, respectively, compared to the study that achieved 80.2% and 66.42% accuracy for the same datasets.

This study's work surpassed most of the recent works that used the same datasets and DLs. Table 8 represents some of the previous work's results compared to this study's results.

For the HARD 2-imbalance dataset, the proposed models outperformed the results obtained by [40,41] by achieving 90.01% and 94.2%, respectively, compared to the 94.69% achieved by this study. For the LABR 2-imbalance dataset, the proposed models outperformed the results obtained by [3,49], by achieving 80.2% and 82%, respectively, compared with this study's model, which achieved an accuracy of 86.36%. For LABR 3-imbalance, this study achieved 69.71% compared to the 66.42% achieved by [49].

In general, the best results were obtained for the HARD 2 - imbalance when using the CNN model and fastText word embedding. The worst results were obtained for the LABR 3-imbalance across the three models. Furthermore, the worst results were obtained when using the CNN-LSTM hybrid model and Word2Vec word embedding. For the remaining datasets, the obtained outcomes were closed according to the three models and the word embedding. Figure 5 illustrates the models with word embedding and their corresponding accuracy.

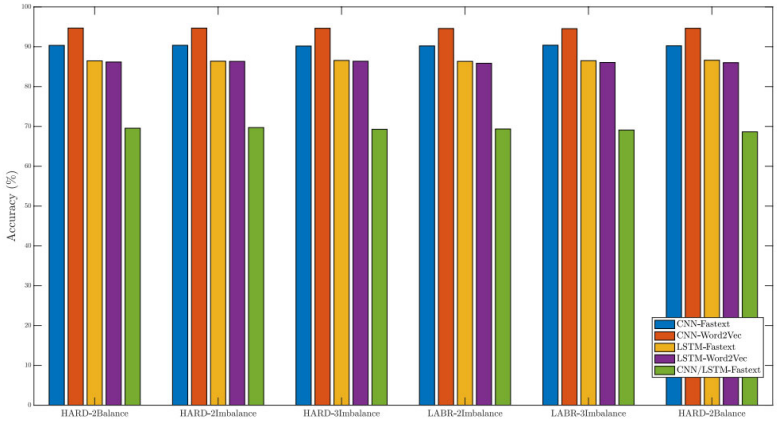

Figure 5. Accuracies of our proposed models for various datasets and the two-word embeddings.

6. Conclusions

In this paper, three DL models (CNN, LSTM, and CNN-LSTM) and two-word embeddings (Word2Vec and fastText) for sentiment analysis were compared. This paper aimed to design DL models that are generalized enough to perform on multiple Arabic datasets using different word embedding techniques. Experiments on HARD and LABR datasets were conducted with many forms of classification, including binary classes and multi-classes with balance and imbalance versions. Additionally, in our datasets, incorporating sentiment information into word embedding during training produced a beneficial outcome. In most cases, the fastText embedding outperformed the Word2Vec embeddings. In this research, word embeddings, feature extraction, and the number of classes were analyzed to investigate how they affected the classification results. Moreover, the outcomes of the classification models were reported. Precision, recall, F1 score, and accuracy were merely some of the performance metrics that were utilized to show how effective the proposed approach was. The findings of all experiences showed that better accuracy scores were achieved with the HARD binary classification-imbalance dataset over the three models. The experiments concluded that fastText word embedding outperformed Word2Vec by accomplishing 94.69% accuracy as compared to the 94.68% obtained by Word2Vec when using the CNN model. The worst results were achieved with the LABR 3-imbalance dataset with the hybrid CNN-LSTM and Word2Vec, which achieved 68.64% accuracy.

For upcoming work, it is intended to utilize several feature extraction techniques to boost the DL's performance, such as bag-of-word (BOW), term frequency-inverse document frequency (TF-IDF), and hybrid techniques. Additionally, it is recommended to investigate the impact of the hyperparameters, such as word embedding size, hidden unit count, and feature engineering technique, on overall performance in the sentiment classification task. DL models with more embeddings must be considered to provide better contextual information, along with increased and balanced datasets such as the BERT-Transformer [62]. The role of pre-processing methods such as lemmatization and stemming needs to be investigated by using effective tools, especially for Arabic pre-processing as developed recently by Obeid Ossama et al. [77]. Moreover, using the character level for sentiment instead of the sentence level is recommended to enhance the results [39]. Furthermore, the model used in this paper can be enhanced to recognize more emotions. In that sense, the model will be more applicable and therefore connected with the multi-model emotion analysis system and assistive technology [78–81]. Finally, it is recommended to explore the effects of applying different SMOTE methods to the LABR3 dataset to enhance the DLs performance.

Author Contributions: Conceptualization, N.E., R.A., K.D. and A.H.; Methodology, N.E., G.V. and R.A.; Software, N.E. and R.A.; Validation, N.E., G.V. and R.A.; Formal analysis, N.E. and R.A.; Investigation, N.E., R.A. and A.H.; Resources, N.E.; Data curation, N.E., R.A. and K.D.; Writing—original draft, N.E. and R.A.; Writing—review & editing, G.V., R.A., M.G., K.D., H.A., M.A.E.-A., B.N.A.-T. and F.A.; Visualization, N.E., K.D. and A.H.; Supervision, A.H.; Project administration, A.H.; Funding acquisition, K.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) programme grant: COG-MHEAR (EP/T021063/1). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source data used in this study was obtained from multiple websites, listed in Section 3.

Acknowledgments: The authors are grateful to the anonymous reviewers for their insightful comments and suggestions, which helped improve the quality of this paper. Hussain acknowledges the support of the UK Engineering and Physical Sciences Research Council (EPSRC) (Grant Nos. EP/M026981/1, EP/T021063/1, and EP/T024917/1).

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

References

1. Liu, B. Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*; Springer: Cham, Switzerland, 2012; Volume 5, pp. 1–167.
2. Alsayat, A.; Elmitwally, N. A comprehensive study for Arabic Sentiment Analysis (challengesand applications). *Egypt. Inform. J.* **2020**, *21*, 7–12. [CrossRef]
3. Al-Bayati, A.Q.; Al-Araji, A.S.; Ameen, S.H. Arabic Sentiment Analysis (ASA) using deep learning approach. *J. Eng.* **2020**, *26*, 85–93. [CrossRef]
4. Ombabi, A.H.; Ouarda, W.; Alimi, A.M. Deep learning CNN–LSTM framework for Arabic Sentiment Analysis using textual information shared in social networks. *Soc. Netw. Anal. Min.* **2020**, *10*, 53. [CrossRef]
5. Omara, E.; Mosa, M.; Ismail, N. Deep convolutional network for Arabic Sentiment Analysis. In Proceedings of the 2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC), Alexandria, Egypt, 17–19 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 155–159.
6. Kolkur, S.; Dantal, G.; Mahe, R. Study of different levels for sentiment analysis. *Int. J. Curr. Eng. Technol.* **2015**, *5*, 768–770.
7. Balaji, P.; Nagaraju, O.; Haritha, D. Levels of sentiment analysis and its challenges: A literature review. In Proceedings of the 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, India, 23–25 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 436–439.
8. Alowaidi, S.; Saleh, M.; Abulnaja, O. Semantic sentiment analysis of Arabic texts. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 256–262. [CrossRef]
9. Alayba, A.M.; Palade, V.; England, M.; Iqbal, R. A combined CNN and LSTM model for Arabic Sentiment Analysis. In Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction, Hamburg, Germany, 27–30 August 2018; Springer: Hamburg, Germany, 2018; pp. 179–191.
10. Dashtipour, K.; Gogate, M.; Adeel, A.; Larijani, H.; Hussain, A. Sentiment analysis of persian movie reviews using deep learning. *Entropy* **2021**, *23*, 596. [CrossRef] [PubMed]
11. Ain, Q.T.; Ali, M.; Riaz, A.; Noureen, A.; Kamran, M.; Hayat, B.; Rehman, A. Sentiment analysis 663 using deep learning techniques: A review. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 424–433.
12. Jhaveri, R.H.; Revathi, A.; Ramana, K.; Raut, R.; Dhanaraj, R.K. A review on machine learning strategies for real-world engineering applications. *Mob. Inf. Syst.* **2022**, *2022*, 1833507. [CrossRef]
13. Varone, G.; Gasparini, S.; Ferlazzo, E.; Ascoli, M.; Tripodi, G.G.; Zucco, C.; Calabrese, B.; Cannataro, M.; Aguglia, U. A comprehensive machine-learning-based software pipeline to classify EEG signals: A case study on PNES vs. control subjects. *Sensors* **2020**, *20*, 1235. [CrossRef]
14. Varone, G.; Ieracitano, C.; Çiftçioğlu, A.Ö.; Hussain, T.; Gogate, M.; Dashtipour, K.; Al-Tamimi, B.N.; Almoamari, H.; Akkurt, I.; Hussain, A. A Novel Hierarchical Extreme Machine-Learning-Based Approach for Linear Attenuation Coefficient Forecasting. *Entropy* **2023**, *25*, 253. [CrossRef]

15. Al-Azani, S.; El-Alfy, E.S.M. Hybrid deep learning for sentiment polarity determination of Arabic microblogs. In Proceedings of the International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; Springer: Cham, Switzerland, 2017; pp. 491–500.
16. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:1404.2188.
17. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
18. 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]
19. Zahidi, Y.; El Younoussi, Y.; Al-Amrani, Y. A powerful comparison of deep learning frameworks for Arabic Sentiment Analysis. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 745–752. [CrossRef]
20. Nassif, A.B.; Elnagar, A.; Shahin, I.; Henno, S. Deep learning for Arabic subjective sentiment analysis: Challenges and research opportunities. *Appl. Soft Comput.* **2021**, *98*, 106836. [CrossRef]
21. Rudkowsky, E.; Haselmayer, M.; Wastian, M.; Jenny, M.; Emrich, Š.; Sedlmair, M. More than bags of words: Sentiment analysis with word embeddings. *Commun. Methods Meas.* **2018**, *12*, 140–157. [CrossRef]
22. Elnagar, A.; Khalifa, Y.S.; Einea, A. Hotel Arabic-reviews dataset construction for sentiment analysis applications. In *Intelligent natural Language Processing: Trends and Applications*; Springer: Cham, Switzerland, 2018; pp. 35–52.
23. Aly, M.; Atiya, A. Labr: A large scale arabic book reviews dataset. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 5–7 August 2013; Volume 2, pp. 494–498.
24. Heikal, M.; Torki, M.; El-Makky, N. Sentiment analysis of Arabic tweets using deep learning. *Procedia Comput. Sci.* **2018**, *142*, 114–122. [CrossRef]
25. Alahmary, R.M.; Al-Dossari, H.Z.; Emam, A.Z. Sentiment analysis of Saudi dialect using deep learning techniques. In Proceedings of the 2019 International Conference on Electronics, Information, and Communication (ICEIC), Auckland, New Zealand, 22–25 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
26. Baly, R.; Hajj, H.; Habash, N.; Shaban, K.B.; El-Hajj, W. A sentiment treebank and morpho-logically enriched recursive deep models for effective sentiment analysis in arabic. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2017**, *16*, 1–21. [CrossRef]
27. Al Sallab, A.; Hajj, H.; Badaro, G.; Baly, R.; El-Hajj, W.; Shaban, K. Deep learning models for sentiment analysis in Arabic. In Proceedings of the Second Workshop on Arabic Natural Language Processing, Beijing, China, 30 July 2015; pp. 9–17.
28. Al-Sallab, A.; Baly, R.; Hajj, H.; Shaban, K.B.; El-Hajj, W.; Badaro, G. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2017**, *16*, 1–20. [CrossRef]
29. AlSurayyi, W.I.; Alghamdi, N.S.; Abraham, A. Deep Learning with Word Embedding Modeling for a Sentiment Analysis of Online Reviews. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.* **2019**, *11*, 227–241.
30. Alayba, A.M.; Palade, V.; England, M.; Iqbal, R. Arabic language sentiment analysis on health services. In Proceedings of the 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR), Nancy, France, 3–5 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 114–118.
31. Al-Azani, S.; El-Alfy, E.S.M. Using word embedding and ensemble learning for highly imbal-anced data sentiment analysis in short arabic text. *Procedia Comput. Sci.* **2017**, *109*, 359–366. [CrossRef]
32. Al-Azani, S.; El-Alfy, E.S. Emojis-based sentiment classification of Arabic microblogs using deep recurrent neural networks. In Proceedings of the 2018 International Conference on Computing Sciences and Engineering (ICCSE), Kuwait City, Kuwait, 11–13 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6.
33. Al-Laith, A.; Shahbaz, M.; Alaskar, H.F.; Rehmat, A. Arasencorpus: A semi-supervised approach for sentiment annotation of a large arabic text corpus. *Appl. Sci.* **2021**, *11*, 2434. [CrossRef]
34. Oussous, A.; Benjelloun, F.Z.; Lahcen, A.A.; Belfkih, S. ASA: A framework for Arabic Sentiment Analysis. *J. Inf. Sci.* **2020**, *46*, 544–559. [CrossRef]
35. Dahou, A.; Elaziz, M.A.; Zhou, J.; Xiong, S. Arabic sentiment classification using convolutional neural network and differential evolution algorithm. *Comput. Intell. Neurosci.* **2019**, *2019*, 2537689. [CrossRef] [PubMed]
36. Altaher, A. Hybrid approach for sentiment analysis of Arabic tweets based on deep learning model and features weighting. *Int. J. Adv. Appl. Sci* **2017**, *4*, 43–49. [CrossRef]
37. Saeed, R.M.; Rady, S.; Gharib, T.F. Optimizing sentiment classification for Arabic opinion texts. *Cogn. Comput.* **2021**, *13*, 164–178. [CrossRef]
38. Addi, H.A.; Ezzahir, R.; Mahmoudi, A. Three-level binary tree structure for sentiment classifica-tion in Arabic text. In Proceedings of the 3rd International Conference on Networking, Information Systems & Security, Marrakech, Morocco, 31 March–2 April 2020; pp. 1–8.
39. Muaad, A.Y.; Jayappa, H.; Al-antari, M.A.; Lee, S. ArCAR: A novel deep learning computer-aided recognition for character-level Arabic text representation and recognition. *Algorithms* **2021**, *14*, 216. [CrossRef]
40. Mhamed, M.; Sutcliffe, R.; Sun, X.; Feng, J.; Almekhlafi, E.; Retta, E.A. A Deep CNN Architecture with Novel Pooling Layer Applied to Two Sudanese Arabic Sentiment Datasets. *arXiv* **2022**, arXiv:2201.12664.
41. Nassif, A.B.; Darya, A.M.; Elnagar, A. Empirical evaluation of shallow and deep learning classifiers for Arabic Sentiment Analysis. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *21*, 1–25. [CrossRef]

42. Al Shboul, B.; Al-Ayyoub, M.; Jararweh, Y. Multi-way sentiment classification of arabic reviews. In Proceedings of the 2015 6th International Conference on Information and Communication Systems (ICICS), Amman, Jordan, 7–9 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 206–211.
43. Elnagar, A. Investigation on sentiment analysis for Arabic reviews. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; IEEE: Piscataway, NJ, USA, 2016.
44. Aliane, A.; Aliane, H.; Ziane, M.; Bensaou, N. A genetic algorithm feature selection based approach for Arabic sentiment classification. In Proceedings of the 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, Morocco, 29 November–2 December 2016; IEEE: Piscataway, NJ, USA, 2016.
45. Barhoumi, A.; Estève, Y.; Aloulou, C.; Belguith, L. Document embeddings for Arabic Sentiment Analysis. In Proceedings of the Conference on Language Processing and Knowledge Management, LPKM, Kerkennah, Tunisia, 8–10 September 2017.
46. Al-Saqqa, S.; Obeid, N.; Awajan, A. Sentiment analysis for Arabic text using ensemble learning. In Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), Aqaba, Jordan, 28 October–1 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–7.
47. Al-Ayyoub, M.; Nuseir, A.; Kanaan, G.; Al-Shalabi, R. Hierarchical classifiers for multi-way sentiment analysis of arabic reviews. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*, 531–539. [CrossRef]
48. Elzayady, H.; Badran, K.M.; Salama, G.I. Arabic Opinion Mining Using Combined CNN-LSTM Models. *Int. J. Intell. Syst. Appl.* **2020**, *12*, 25–36. [CrossRef]
49. Abu Kwaik, K.; Saad, M.; Chatzikyriakidis, S.; Dobnik, S. LSTM-CNN deep learning model for sentiment analysis of dialectal Arabic. In Proceedings of the International Conference on Arabic Language Processing, Nancy, France, 16–17 October 2019; Springer: Cham, Switzerland, 2019; pp. 108–121.
50. Nouhaila, B.; Habib, A.; Abdellah, A.; El Farouk Abdelhamid, I. Arabic sentiment analysis based on 1-D convolutional neural network. In Proceedings of the Third International Conference on Smart City Applications, Karabuk, Turkey, 7–9 October 2020; Springer: Cham, Switzerland, 2021; pp. 44–55.
51. Al-Dabet, S.; Tedmori, S. Sentiment Analysis for Arabic Language using Attention-Based Simple Recurrent Unit. In Proceedings of the 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
52. Naqvi, U.; Majid, A.; Abbas, S.A. UTSA: Urdu text sentiment analysis using deep learning methods. *IEEE Access* **2021**, *9*, 114085–114094. [CrossRef]
53. Kapočiūtė-Dzikienė, J.; Damaševičius, R.; Woźniak, M. Sentiment analysis of lithuanian texts using traditional and deep learning approaches. *Computers* **2019**, *8*, 4. [CrossRef]
54. Divyapushpalakshmi, M.; Ramalakshmi, R. An efficient sentimental analysis using hybrid 779 deep learning and optimization technique for Twitter using parts of speech (POS) tagging. *Int. J. Speech Technol.* **2021**, *24*, 329–339. [CrossRef]
55. Vasili, R.; Xhina, E.; Ninka, I.; Terpo, D. Sentiment Analysis on Social Media for Albanian Language. *Open Access Libr. J.* **2021**, *8*, 1–31. [CrossRef]
56. Darwish, K.; Magdy, W. Arabic information retrieval. *Found. Trends®Inf. Retr.* **2014**, *7*, 239–342. [CrossRef]
57. Darwish, K.; Magdy, W.; Mourad, A. Language processing for arabic microblog retrieval. In Proceedings of the the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 2427–2430.
58. Terechshenko, Z.; Linder, F.; Padmakumar, V.; Liu, M.; Nagler, J.; Tucker, J.A.; Bonneau, R. A comparison of methods in political science text classification: Transfer learning language models for politics. *SSRN Electron. J.* **2020**, 1–25. [CrossRef]
59. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
60. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
61. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 135–146. [CrossRef]
62. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional trans-formers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
63. Omara, E.; Mosa, M.; Ismail, N. Applying Recurrent Networks For Arabic Sentiment Analysis. *Menoufia J. Electron. Eng. Res.* **2022**, *31*, 21–28. [CrossRef]
64. Sivakumar, S.; Videla, L.S.; Kumar, T.R.; Nagaraj, J.; Itnal, S.; Haritha, D. Review on Word2Vec Word Embedding Neural Net. In Proceedings of the 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 10–12 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 282–290.
65. Khalid, U.; Hussain, A.; Arshad, M.U.; Shahzad, W.; Beg, M.O. Co-occurrences using Fasttext embeddings for word similarity tasks in Urdu. *arXiv* **2021**, arXiv:2102.10957.
66. Cliche, M. BB_twtr at SemEval-2017 task 4: Twitter sentiment analysis with CNNs and LSTMs. *arXiv* **2017**, arXiv:1704.06125.
67. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.

68. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.* **2017**, *18*, 6765–6816.
69. Ahmed, R.; Gogate, M.; Tahir, A.; Dashtipour, K.; Al-Tamimi, B.; Hawalah, A.; El-Affendi, M.A.; Hussain, A. Novel deep convolutional neural network-based contextual recognition of Arabic handwritten scripts. *Entropy* **2021**, *23*, 340. [CrossRef]
70. Rani, S.; Kumar, P. Deep learning based sentiment analysis using convolution neural network. *Arab. J. Sci. Eng.* **2019**, *44*, 3305–3314. [CrossRef]
71. Cheng, Y.; Sun, H.; Chen, H.; Li, M.; Cai, Y.; Cai, Z.; Huang, J. Sentiment analysis using multi-head attention capsules with multi-channel CNN and bidirectional GRU. *IEEE Access* **2021**, *9*, 60383–60395. [CrossRef]
72. Shickel, B.; Tighe, P.J.; Bihorac, A.; Rashidi, P. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 1589–1604. [CrossRef]
73. Minaee, S.; Azimi, E.; Abdolrashidi, A. Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. *arXiv* **2019**, arXiv:1904.04206.
74. Yue, W.; Li, L. Sentiment analysis using Word2vec-CNN-BiLSTM classification. In Proceedings of the 2020 Seventh International Conference on Social Networks Analysis, Management and Security (SNAMS), Paris, France, 14–16 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
75. Rehman, A.U.; Malik, A.K.; Raza, B.; Ali, W. A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis. *Multimed. Tools Appl.* **2019**, *78*, 26597–26613. [CrossRef]
76. Jain, P.K.; Saravanan, V.; Pamula, R. A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–15. [CrossRef]
77. Obeid, O.; Zalmout, N.; Khalifa, S.; Taji, D.; Oudah, M.; Alhafni, B.; Inoue, G.; Eryani, F.; Erdmann, A.; Habash, N. CAMeL tools: An open source python toolkit for Arabic natural 838 language processing. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 7022–7032.
78. Poria, S.; Cambria, E.; Howard, N.; Huang, G.B.; Hussain, A. Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomputing* **2016**, *174*, 50–59. [CrossRef]
79. Poria, S.; Chaturvedi, I.; Cambria, E.; Hussain, A. Convolutional MKL based multimodal emotion recognition and sentiment analysis. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 439–448.
80. Poria, S.; Majumder, N.; Hazarika, D.; Cambria, E.; Gelbukh, A.; Hussain, A. Multimodal sentiment analysis: Addressing key issues and setting up the baselines. *IEEE Intell. Syst.* **2018**, *33*, 17–25. [CrossRef]
81. Poria, S.; Peng, H.; Hussain, A.; Howard, N.; Cambria, E. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing* **2017**, *261*, 217–230. [CrossRef]