Sentiment Analysis on Mixed-Languages Customer Reviews: A Hybrid Deep Learning Approach

Conference Paper · December 2021

DOI: 10.1109/SMART52563.2021.9676277

2 authors:

Ashok Kumar D
Government Arts College, Kulithalai - 639120

Anandan Chinnalagu
Government Arts College, Kulithalai, Karur.

Sentiment Analysis on Mixed-Languages Customer Reviews: A Hybrid Deep Learning Approach

Ashok Kumar Durairaj[1], Anandan Chinnalagu[2]
[1,2]*Dept. of Computer Science, Govt. Arts College (Affiliated to Bharathidasan University Tiruchirappalli), Kulithali, Karur, TN, India*
E-mail: [1]drashoktrichy22@gmail.com, [2]anandanc@hotmail.

***Abstract*—Machine Learning (ML), Natural Language processing (NLP) techniques are used for Sentiment Analysis (SA) problems. Many study results shown predicting more accurate sentiment in mixed language customer reviews are remains challenge. Customer sentiments are classified as positive negative and neutral. Many businesses providing products and services rely on reviews from customers through social media to improve customer experience and increase revenue. Based on our SA research studies and experiments, deep learning neural networks and fastText sequential models are shown better performance and more accurate sentiment prediction in text datasets compare to traditional classification algorithms. In this experiment we build a novel hybrid deep learning SA model using fastText word embedding library and authors multi-layer SAB-LSTM model called fastText-SA-BLSTM. Compare the hybrid model result with traditional linear SVM (LSVM), fastText and authors SAB-LSTM models. For this experiment, data has been collected from online customer reviews and social media posts. This experiment result shows the proposed hybrid model outperforms traditional models and shows more accurate context-based sentiment results. We conclude this paper with more details of our findings and experimental results as well as a proposed future work.**

Keywords: Sentiment Analysis; Text Classification; FastText; BiLSTM; Deep Learning; Machine Learning.

I. Introduction

ML and NLP technologies plays a major role of business process automation, and sentiment analysis, emotion detection, text classification and information extract (IR) applications. Many online businesses are relied on consumer's positive reviews in social media platforms. To improve the customer experience and services analyzing the consumer reviews become a crucial task for business and it's a time-consuming process to predict the sentiment in customers' written reviews in mixed languages. To solve the SA problems, researchers and enterprises are keep doing more research and innovations in improving AI, ML and NLP technologies, and introducing state-of-the-art ML, DL, NLP models, algorithms and easy to use frameworks for model training, testing and validations. Recently, many NLP tasks are performed using RNN, in particular LSTMs, and CNN [12]. Recent SA survey paper identified, there are 112 papers identified, there are 112 papers published for DL. The LSTM and CNN are the most used algorithms for SA [22]. Facebook AI research (FAIR) lab introduced fastText library text classification, representation and it's a lightweight method and work on standard generic hardware with multicore CPU, the fastText models shown that it is often on par with recently proposed DL methods in terms of accuracy, performance, faster during model training and evaluation [9]. The fastText introduce a new word embedding approach and Continuous Bag of Words (CBOW) model like word2vec. The original fastText model is Pre-Trained on Wikipedia and its available for 294 languages [24]. Authors build a novel multilayer sequence processing model called SAB-LSTM model, it consists of two LSTM units and multilayers, one unit taking the input in a forward direction and other unit taking the input in a backward direction [25]. Authors experimental result shown SAB-LSTM model outperformed the traditional LSTM, Bi-LSTM models [13]. In this experiment, our main goal is to build a hybrid SA model utilize the fastText word embedding, SAB-LSTM architecture with proven binary and multi-class algorithm. Our contributions made towards building a multilayer hybrid SAB-LSTM with fastText word embedding (fastText-SAB-LSTM) and solve known Out of Vocabulary (OOV) issues with sentiment predictions made from datasets which contains mixed language texts and Data pre-processing steps are customized to fit the dataset and hybrid models requirements.

This paper details several contributions from various researchers in Section II. The methods used, as well as the pre-processing steps and flow of data pipeline are presented in Section III. The literature reviews on fastText, BiLSTM, and approach taken for the ML and DL models, architecture of various algorithms used for the models are discussed in Section IV. Details of the experimental setup and results are documented throughout experimental research studies are presented in Section V. Results of the concluding model and proposed future work are discussed in Section VI.

II. Review

Reviewed recent NLP, SA related research papers and researchers' contributions towards Transformers, ML, DL models used for text classification and sentiment prediction. Our review focused on Word Embedding, fastText LSTM, and BiLSTM models.

Recent years, most of the sequential SA models are build using neural networks such as RNN, LSTM and GRNN [3]. The encode-decoder architectures and attention layer mechanisms are become a standard integration of sequential and transformation models. The Neural Machine Translation (NMT) is a learning approach used for automated translation and it overcomes conventional phrase-based translation systems. NMT systems lack robustness, computationally expensive to process large data sets and large models [23]. Many research studies shown using the pre-trained language model and word embedding models are improving the effectiveness of many NLP tasks [26] [27]. Google introduce Bidirectional Encoder Representation from Transformers (BERT), its pre-trained from unlabeled text data for language representation.[2]. BERT model can be fine-tuned by adding an additional output layer to BERT. This model used for a wide range of NLP tasks which includes SA models.

Facebook AI Research lab team (Joulin, Grave, Bojanowski, Douze, Jegou, and Mikolov) published a paper (2016) related to compressing text classification model using FastText, this paper specifically addresses the compromise between classification accuracy and the model size, implemented in the fastText library. Overall, fastText.zip is often more accurate [10] compared to recent work based on CNN.

A universal language model called ULMFiT introduced by Jeremy and Sebastian (2018) [20], presenting that the ULMFiT is an effective transfer learning method applicable to NLP tasks, and introduced a variety language model fine-tuning technique. Md, Zia Uddin and Erik G. Nilsson published a paper (2020) related Emotion recognition on audio and video sources. The Neural Structured Learning (NSL) fast deep learning approach used for emotion training and emotion recognition. NSL generated a superior recognition rate than the traditional DBN, CNN, and RNN when using the audio data set of emotion [21].

Hu, Lei, Philip, Yu and Bing presented (2020) an article related to BERT model for SA. This paper analysis hidden representations learned from reviews on BERT pre-trained model, found that pre-trained language models (LMs) BERT yield significant performance gains, and analysis result showed that MLM model learns features and dedicate most of the aspects' features to domains and semantics of aspects themselves rather than opinions [8]. Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss, Balaji Lakshminarayanan from Google Research team presented a paper at Neural Information Processing Systems (NeurIPS) conference (2020) related to estimation of predictive uncertainty of deep learning model [1].

Nalini Chintalapudi, Gopi Battineni and Francesco Amenta published (2021) an article related to Covid-19 SA using DL model [5]. The following emotion text of sad, anger, joy and fears are labeled and data has been analyzed using the BERT model. BERT model results are compared with traditional SVM, LR and LSTM models. BERT model result shows more accuracy 89%, compare to other model's accuracy LR 75%, SVM 74.75% and LSTM 65%. Mrityunjay, Amit and Shivam published (2021) a paper related to Covid19 SA using BERT model and twitter data set. The BERT model experiment result showed 94 % accuracy [4] for given dataset. Akbar Karimi, et al., published a paper (2021) related to Aspect-based SA using BERT model, used 2 BERT language model with a hidden layers for semantic representations of sequence input data and result showed language model performed well compare to out of the box BERT [6].

Anna Kruspe, Matthias Haberle, Iona Kuhn and Xia Xiang Zhu published a paper (2020) related to COVID19 Cross-language SA of European Twitter messages from Italy, Spain, France, and Germany [19]. A pre-trained word embeddings models using skip-gram of word2vec and multilingual version of BERT used and a model constructed with fully-connected Rectified Linear Unit (ReLu) layer to process output from embedded vector and sigmoid activation function in output layer. Analysis over-all 4.6 million tweets in which 79,000 tweets contains one keyword of Covid-19. Researchers took geolocation-based data and trends were varying and these researchers concluded, this study will be continued to collect tweet data from other countries and compare the result.

Marco Pota, Massimo Esposito, Rosario Catelli and Mirko Ventura presented (2020) a case study in Italina Twitter SA using BERT-Based and introduce a different two step approach [7]. The first step is to transform emojis, emoticons as well as jargon in the tweet into plan text, next step is to resulting tweets are classified using BERT model, The results obtained show the effectiveness of the approach and indicate that this approach can be used for other languages.

Raviya K and Dr. Mary Vennila published a paper (2021) related to a hybrid SA system using Spark ML pipeline for improving the machine learning efficiency and computation to solve SA problem, used Apache Spark ML architecture [17]. To measure the efficiency of the proposed system, a Hybrid CNN-SVM model, and traditional SVM, RF, and NB classifiers models are developed for this experiment. The comparison of models result showed the proposed hybrid CNN-SVM model outperformed traditional classifier models.

Mohamed Chiny, Marouane Chihab, Younces Chihab and Omar Bencharel published a paper (2021) related to a hybrid binary classification SA model based on a LSTM network, a rule-based SA lexicon and the TF-IDF weighting method. In addition, the traditional LR, k-Nearest Neighbors (KNN), RF, SVM, NB algorithms are used for experiments and models trained on IMDB and Twitter dataset of USA Airlines' sentiment analysis. Proposed model evaluation results show more Accuracy and F1 score [18]. Arifur, Syful, Ratnadip, Javed, and Mohammed published a paper (2021) related to detecting sarcasm in SA. Proposed DT and RF models, the experimental result show RF (91.90%) outperform DT (91.84%) in terms of accuracy [16].

Zabit and Begonya published a paper (2020) related to a BiLSTM model for binary classification and find the polarity of people's opinions in movie reviews. Used BiLSTM model with a global pooling mechanism and single layer to get more efficient model. The experiment results shown more than 80% accuracy on IMDB, SST2 and MR datasets [14].

Beakcheol, Myeonghwi, Gaspard, Sang-ug and Jong published (2020) an article related to Bi-LSTM+CNN hybrid model with attention layer which provided more accuracy of text classification. To evaluate the model performance, IMDB movie review data used on this proposed model, the test results showed that the proposed hybrid attention Bi-LSTM+CNN model produces more accurate results with higher recall and F1 scores, compare to LSTM, MLP, and CNN [15].

Authors Ashok and Anandan (2020) presented a SA study paper on Covid19 conversations [13]. The authors built and implemented a novel RNN based neural network model using SAB-LSTM with additional layers to overcome problems with accuracy, performance, and model better performance and more accuracy compare to traditional LSTM and Bi-LSTM models. Used COVID19 data sets and results show more than 50% of people emotion is sad about Covid19.

III. Methodologies

Most recent researches are done using deep learning and neural network models for SA and NLP tasks, however neural network approaches cost more compare to traditional baseline methods for supervised, unsupervised and transfer learning. The social media data sources, blogs and online news media text data contents are collected for research studies. The data pre-processing steps are very important to prepare data obtain from publicly available data sources. In general, the customers review data is very often incomplete, inconsistent and filled with noise, contained errors, and data should be cleansed before using it for training the model. In this section, we discussed about our data pre-processing steps and data pipeline used for this experimental research.

A. Data Pre-processing

Authors have built a custom script using Python language for data cleansing processes. The Neural networks and embedding models are not need a lot of preprocessing steps. The following are the data pre-processing tasks: lowercasing all words and removed newlines, special characters, punctuations, stop words and stripping recurring headers and outros from the text. The Fig.1. shows our pre-processing data flow diagram.

The idea of the preprocessing steps is to strip useless information, for example formatting like newlines and characters likely not found in the embeddings or the recurring first string (dpa) which does not carry significant information. We avoid resource and/or time-hungry preprocessing.

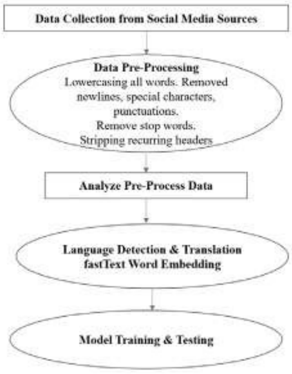

Fig. 1: Data Pre-processing flow diagram

B. Proposed Model Methodology

The following Fig.2. shows the proposed model data pipeline, we collected multilingual customer reviews, our pre-processing steps clean the social media data sets and our custom scripts process the pre-processed data to identify language detection and then it translates to English if the text in other languages. fastText process word embedding process fed the embedded data to BiLSTM model to predict the sentiment,

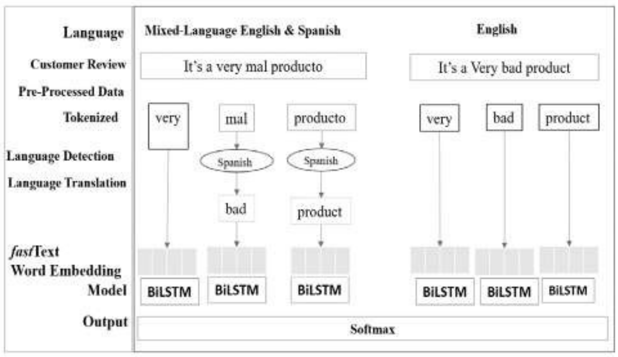

Fig. 2: Proposed model methodology

IV. Literature Review and Proposed Model

In this section, reviewed fastText word embedding, LSVM, LSTM, BiLSTM and proposed fastText-SA-BLSTM model architecture.

A. Linear LSVM

LSVM is a supervised learning algorithm and mostly used for Classification. SVM has two type of classifier

Linear and Non-Linear. In Linear Classifier, a data point considered as a p-dimensional vector and separating data in a liner order, but the best hyperplane consider one which maximize the margin. However, it has maximized the distance between hyperplane and the data points. These data points which influence the hyperplane are known as support vector. The linear and non-linear classifier data separation shown in the Fig.3. for the 2-dimensional dataset [30]. If the dataset is separable then linear kernel works well for classification.

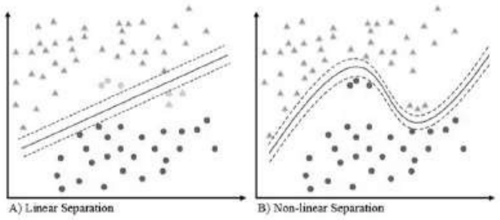

Fig. 3: A) Linear B) Non-linear Data Spearation

The following Fig. 4. shows the linear kernel SVM model [31]. There many kernel functions have been developed over the years [32].

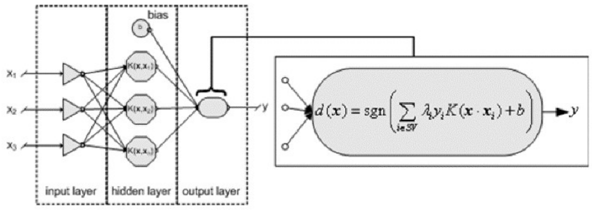

Fig. 4: Linear SVM Architecture

The following Eq. (1) defines the linear function of the kernel $k(x, y)$ for $<x, y>$, here *c* represents as a constant.

$$k(x,y) = x^T + c \tag{1}$$

Linear kernel is mostly used for text classification.

B. *fastText Word Embedding*

The following Fig.5. show the simple linear architecture of the fastText with *N* gram features [9].

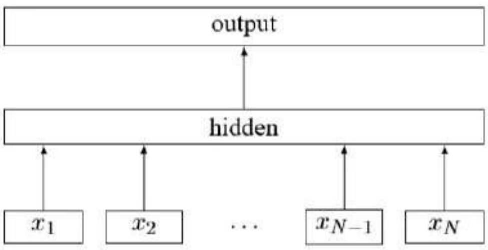

Fig. 5: Linear fastText Architecture

To form hidden variable, the embedded features are averaged by the model. The Bang-of-words (BOW) text is fed into lookup layer first, each word is embedded by model and it obtain an averaged embedding for full text. The hidden layer will have x dim number of parameters for n words, here dim is the embedding size, and *n* words are size of vocabulary.

$$x_1, x_2, \ldots\ldots x_N \tag{2}$$

The averaged single vector is fed to linear classifier and softmax is applied over the linear transformation. Here the linear transformation is dimension1 of $x_N$output, here N output is number of output classes. The Eq (3) shows the final negative log-likelihood:

$$-\frac{1}{N} = \sum_{n=1}^{N} y_n \log\left(f(BAx_n)\right) \tag{3}$$

x_n is the representation of n-gram feature.
A is the lookup matrix of word embedding,
B is the transformation of linear output,
f is the softmax function.

$$\text{softmax}(z) = \frac{\exp(z)}{\sum_{k=1}^{K} exp(z)} \tag{4}$$

fastText introduce a new word embedding approach which extend the Continuous Bag of Words (CBOW) and skipgram models, here each word is considered as *n*-grams bag of character. fastText vanilla version is trained on Wikipedia and its available for 294 languages, and it is a simple classification algorithm implemented in C++ [11]. *fast*Text sees words as the sum of their character n-grams and treats as a vector representation of character *n*-gram and words are represented a sum of these representations [29]. A new approach has clear advantages, as it can calculate embeddings even for out-of-vocabulary (OOV) words. To predict the context, distributed representation of the input word is used Skip-gram models, Skip-gram maximize the average log probability for given training words,

$$w_1, w_2, w_3 \ldots\ldots, w_T \tag{5}$$

In the following Equation used for computing probability.

$$\frac{1}{T}\sum_{t=1}^{T} \left[\sum_{j=-k}^{T} \log p\left(w_{t+j} \middle| w_t\right) \right] \tag{6}$$

Here k represents the size of the training window and function of the center word w_t.

-k to k represents inner summation to compute the log probability of correctly predicting the word w_{t+j} given the word in the middle w_t. The outer summation represents over all words in the training corpus. In skip-gram model every word *w* is associated with two learnable parameters vectors $u_{w,}$ and $v_{w,}$ They are input and output vectors of *w* respectively. The probability is correctly predicting the word $u_{i,}$ given the word $w_{j,}$ is defined as Eq. (7).

$$p\left(w_i \mid w_j\right) = \frac{\exp\left(u_{w_i} \top v_{w_j}\right)1}{\sum_{l=1}^{V} \exp\left(u_{w_l} \top v_{w_j}\right)} \tag{7}$$

Where *V* represents the number of words in the vocabulary.

C. *BiLSTM*

The BiLSTM is used in this proposed model. The extended LSTM architecture is shown in Fig. 6.

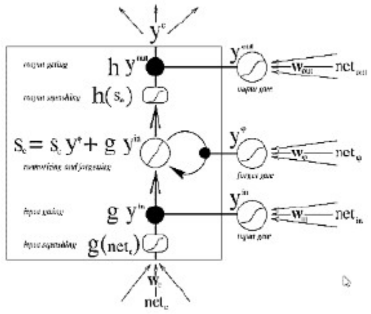

Fig. 6: Extended LSTM Architecture Diagram

The extended LSTM consists of Input, Output, multiplicative Forget gates and the *f* sigmoid function is used for activation [28].

The Constant Error Carousels (CEC) feature solves the vanishing gradient problem of LSTM and CECs provides the constant back flow to cell if no signal of input or error. The forward and backward activation error flow from CECs is protected by both Input and output gates. The irrelevant input to the gate is avoided if 0 activation or when gates are closed. The Forward LSTM computes output from network input and Backward LSTM computes errors from network output and based on gradient descent it updates network weights. The loss function is used to calculate gradient for back propagation (BP) [28]. The sigmoid function *sig(t)* activation is used for input and output control of LSTM Gates. The input and output gates are closed if the sigmoid function value is set to 0 and it allows the input if this function value is set to 1.

Here are the equations for all three gates.

$$i_t = \sigma(\omega_i [h_{t-1}, x_t] + b_i) \tag{8}$$

$$o_t = \sigma(\omega_f [h_{t-1}, x_t] + b_o) \tag{9}$$

$$f_t = \sigma(\omega_o [h_{t-1}, x_t] + b_f) \tag{10}$$

σ is a *sig(t)* sigmoid function,

$$\text{sig}(t) = \frac{1}{1 + e^{-t}} \tag{11}$$

x_t is the current timestamp of input.

h_{t-1} is the previous state of the output timestamp.

ω_i, is weight of the input gate.

ω_f, *and* ω_o are weights of forget and output gates.

b_i, is the bias of input gate.

b_o and b_f are the bias of output, forget gates.

Here are the cell state equations of input, output and forget gates.

Input $$\tilde{c}_t = tanh(c\omega [h_{t-1}, x_t] + b_c) \tag{12}$$

Output $$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \tag{13}$$

Forget $$h_t = o_t * tanh(c^t) \tag{14}$$

$\tilde{c}_t$, c_t , and h_t are input, memory and final output of the gates cell state at timestamp (*t*).

Proposed Hybrid fast-Text-BLSTM

The proposed fastText-SA-BLSTM hybrid model consists of Input layer, Language detection and translation layers, fastText Embedded layer, Bi-Directional LSTM neutral network, Dropout, Dense and Output layers.

Fig.7. show the fastText-SA-BLSTM architecture. The input layers process the multilingual mixed customer reviews and speech transcript dataset and vectorizing the data using word embedding technique. In the following Eq. (15, x representing input during the language detection, language translation process [13].

$$x = x_{1.} x_{2.} x_{3.} \ldots\ldots x_T \tag{15}$$

In the detection layer, input text processed to detect the non-English text, here d represents input to detection layer.

$$d = d_{1.} d_{2.} d_{3.} \ldots\ldots d_T \tag{16}$$

If input text identified as non-English text, then the language translation layer converts the text to English, here t represents translation layer input.

$$t = t_{1.} t_{2.} t_{3.} \ldots\ldots t_T \tag{17}$$

The translation layer fed the input to fastText for word embedding and each word converted as vector.

$$S = (w_{1,} \ \ w_3, \ w_3 \ldots\ldots \ w_n) \tag{18}$$

Proposed model has Input, Language detection, and translation for identifying the language in text, fastText library for word embedding used an Input layer. The network optimizer used for setting the number of networks based on input data. There are 196 memory units of BiLSTM, (32,200,128) Embedding and input dimension configured for training the model. There 5 layers of dense and activation function of Softmax are used in the output layer of this model.

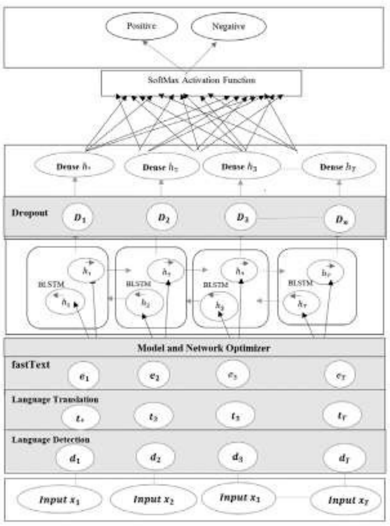

Fig. 7: Hybrid fastText-SA-BLSTM architecuture diagram

V. Experimental Results

In this section, presented environmental setting, data sets and experimental results of proposed fastText-SAB-LSTM model's experiment result and compared results with traditional LSVM, LSTM and Bi-LSTM models for the same data sets.

A. Experiment Setting

For this experiment a standard server configuration used for models' evaluations The details of the software, libraries and tools listed in Table I. for this experiment.

Table 1: Experimental Environment

Environment	Description
Server Configuration	64-bit Windows 10 Operating System, i7 processor, processor, 16 Giga Bytes Memory.
Word Embedding	fastText Library and Keras Encoder
Programming Language & Tool	Python 3.8.8 Jupyter Notebook 6.3.0
Libraries and Frameworks	fastText, Pandas, Numpy, Seaborn, Matplotlib, Nltk, Scikit-learn, Keras

B. Data Source

The sentiment analysis datasets collected from publicly available kaggle.com data source, Total 778631 Twitter Sentiment Analysis dataset, 70% (545041) of data used for Training and 30% (233590) of data used to test the models which includes customer reviews from Facebook posting. Designed and developed fastText-SAB-LSTM hybrid model, fastText used at input layer for word embedding. All models tested with the same dataset. Multiple iterations of the fastText-SA-BLSTM, fastText, LSVM, and SAB-LSTM models training, testing results are recorded and compared the performance measures.

C. Measures

LSVM, fastText, SAB-LSTM and hybrid fastText-SA-BLSTM models are tested with the following parameter setting shown in the Table II.

Table 2: Parameter Setting for Model Evluation

Model	Parameter and Values
SLVM	n=1 (Unigram), n=2 (Bigram), n=3 (Trigram) Kernel=linear
fastText	n=1 (Unigram), n=2 (Bigram), n=3 (Trigram) Epoch 10, Learning Rate 0.01 Loss Function: Softmax
SAB-LSTM and fastText-SA-BLSTM	Neurons 192, Learning Rate 0.01, Epoch 10 Batch Size 20, Embedding Vector 128 Dropout 0.2, Activation: Softmax Optimizer: Adam

The following metrics True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (TN) are used to evaluate the model's testing performance. Table III. show the models evaluation metrics values.

Table 3: Models Evaluation Metric

Models	TP	FP	TN	FN
LSVM	95772	18687	105116	14015
fastText	107451	11680	98108	16351
SAB-LSTM	88764	25695	91100	28031
fastText-SA-BLSTM	104117	9885	13552	106036

$$Precision = \frac{tp}{tp+fp} \quad (19)$$

$$Recall = \frac{tp}{tp+fn} \quad (20)$$

$$F1\ Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (21)$$

$$\text{Accuracy }\% = \frac{\text{tp+tn}}{\text{tp+fp+tn+fn}} * 100 \quad (22)$$

The following TABLE IV. shows the performance measures of model's Accuracy, Recall, Precision and F1 Score.

Table 4: Models Evaluation Metric

Model	Accuracy	Recall	Precision	F1
LSVM	86 %	0.87	0.84	0.85
fastText	88%	0.87	0.89	0.88
SAB-LSTM	77%	0.75	0.77	0.76
fastText-SA-BLSTM	90%	0.87	0.91	0.89

D. Experimental Results

Proposed fastText-SA-BLSTM model performed better than the other models during the model training. The experimental results showed proposed model showed higher accuracy compare to other models. The traditional LSVM and fastText are showing similar model accuracy results. The following TABLE V shows the models sentiment scores.

Table 5: Sentiment Score

Models	Positive	Negative
LSVM	46.99%	53.00%
fastText	52.99%	47.00%
SA-BLSTM	50.00%	50.00%
fastText-SA-BLSTM	50.37%	49.62%

VI. Conclusion and Future Work

Proposed model (*fastText*-SA-BLSTM) outperformed traditional models in terms of model performance and accuracy. The language detection, language translation, fastText word embedding layers are improved the accuracy of the model. The dense layer, dropout layer and softmax activation function are avoided the model's overfitting problem. The validation result shows fastText-SA-BLSTM and SAB-LSTM models are predicted the sentiment base on context compare to traditional models. Adding translation layer to the hybrid model improve the accuracy on multi-language data sets.

Based on this experimental result, the future work needed to explore pre-trained transformer models and build a framework for data pipe and model training. Fine-tuned BERT transformer model can be used for multiple language dataset, it can be integrated with the SA framework for predicting sentiment from various domains datasets in different languages. The BERT model can be replaced with fastText embedding and language translation layers in fastText-SA-BLSTM model.

References

[1] Jeremiah Zhe Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax-Weiss and Balaji Lakshminarayanan, "Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness", Google Research, 34th Conference NeurIPS, Vancouver, Canada. October 2020.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Google AI Language, May 2019.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakon Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, "Attention is all you need". 31st Conferencence on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, December, 2017.

[4] Mrityunjay Singh, Amit Kumar Jakhar and Shivam Pandey, "Sentiment analysis on the impact of coronavirus in social life using BERT model", Social Network Analysis and Mining, Licence to Springer-Verlag GmBH Austria, part os Springer Nature. February 2021.

[5] Nalini Chinatalapudi, Gopi Battineni and Francesco Amenta, "Sentiment Analysis of COVID-19 Tweets Using Deep Learning Models", Infectious disease reports, MDPI, Basel, Switzerland. April 2021.

[6] Akbar Karimi, Leonardo Rossi, and Andrea Prati, "Improving BERT Performance for Aspect-Based Sentiment Analysis", Department of Engineering and Architecture, University of Parma. Italy. March 2021.

[7] Marco Pota, Mirko Ventura, Rosario Catelli, and Massimo Esposito, "An Effective BERT-BasedPipleline for Twitter Sentiment Analysis: A Case Study in Italina", Sensors, MDPI Article, December 2020.

[8] Hu Xu, Lei Shu, Philip S. and Yu, Bing Liu, "Understanding Pre-trained BERT for Aspect-based Sentiment Analysis", Department of Computer Science, University of Illinois at Chicago, IL, USA. Proceedings of the 28th International Conference on computational Linguistics, Barcelona, Spain, December 2020.

[9] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification", Facebook AI Research arXiv:1607.01759v3 [cs.CL], August 2016.

[10] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jegou, and T. Mikolov, "FastText.Zip: Compressing text classification models", Facebook AI Research, arXiv:1612.03651v1 [cs.CL], December 2016.

[11] Vladimir Zolotov and David Kung, "Analysis and optimization of fastText linear text classifer" IBM T.J.Watson Research Center. Yorktown Heights, NY, USA. arXiv:1702.05531v1 [cs.CL], 2017.

[12] Alexis Conneau, Holger Schwenk, Loic Barrault and Yann Le Cun, "Very Deep Convolutional Network for Text Classification", Facebook AI Research. arXiv:1606.0781v2[cs.CL], January 2017.

[13] Ashok Kumar Durairaj and Anandan Chinnalagu, "Sentiment and Emotion in Social Media COVID-19 Conversations: SAB-LSTM Approach", Proceedings of SMART-2020, IEEE Conference. Teerthanker Mahaveer University, Moradabad, India. December 2020.

[14] Zabit Hameed and Begonya Garcia-Zapirain, "Sentiment classification using a single-layered BiLSTM model" eVida Research Group, University of Deusto, Bilbao, Spain. *IEEE Access*, April 2020.

[15] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, Sang-ug Kang, and Jong Wook Kim, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism", Applied Sciences, MDPI, 24 August 2020.

[16] Arifur Rahaman, Ratnadip Kuri, Syful Islam, Md.Javed Hossain and Mohammed Humayin Kabir, "Sarcasm Detection in Tweets: A Feature-based Approach using Supervised Machine Learning Models", International Journal of Advanced Computer Science and Applications (IJACSA). Vol 12, No. 6. 2021.

[17] Raviya K and Dr. Mary Vennila S, "An Implementation of Hybrid Enhanced Sentiment Analysis System using Spark ML Pipeline: A Big Data Analytics Framework", International Journal of Advanced Computer Science and Applications (IJACSA). Vol. 12, No. 5. 2021.

[18] Mohamed Chiny, Marouane Chihab, Younces Chihab and Omar Bencharel, "LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model", International Journal of Advanced Computer Science and Applications (IJACSA). Vol. 12, No. 7. 2021

[19] Anna Kruspe, Matthias Haberle, Iona Kuhn and Xiao Xiang Zhu, "Cross-language sentiment analysis of European Twitter message during the COVID-19 pandemic", arXiv:2008.12172v1 [cs.SI], August 2020.

[20] Jeremy Howard and Sebastian Ruder "Universal Language Model Fine-tuning for Text Classification", Association for Computational Linguistics (Long Papers), Australia. July 2018.

[21] Md, Zia Uddin and Erik G. Nilsson, "Emotion recognition using speech and neural structured learning to facilitate edge intelligence", Engineering Applications of Artificial Intelligence 94(2020) 103775. June 2020.

[22] Alexander Ligthart, Cagatay Catal and Bedir Tekinerdogan, "systematic reviews in sentiment analysis: a tertiary study", Artificial Intelligence Review, February 2021.

[23] Yonghui Wu, Mike schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, et al. "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation", Google, arXiv:1609.08144v2 [cs.CL], October 2016.

[24] Matthias Nitsche, and Stephan Halbritter, "Comparison of Neural Document Classification Models" Hamburg University of Applied Sciences, Department of Computer Science , Hamburg, Germany. 2019.

[25] Karthik Gopalakrishnan and Fathi M. Salem, "Sentiment Analaysis Using Simplified Long Short-term Memory Recurrent Neural Networks", Depatment of Electrical and Computer Engineering, Michigan State University USA. arXiv:2005.03993v1[cs.CL]. 2020.

[26] Matthew E. Peters, Mark Neumann, Luke Zettlemoyer and Wen-tau Yih, "Dissecting Contextual Word Embeddings: Architecture and Representation" Allen Institute for AI, Seattle, WA, USA. Paule G.Allen Computer Science & Engineering, University of Washington.arXiv:1808949v2 [cs.CL] September 2018.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans and Ilya Sutskever, "Improving Lanaguage Understanding by Generative Pre-Training", OpenAI. 2018.

[28] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, "Efficient Estimation of Word Representations in Vector Space", 2013.

[29] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov, "Enriching Word Vector with Subword Information". Facebook AI Research. arXiv:1607.04606v2 [cs.CL]. Jun 2017.

[30] Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes and Donald Brown (2019), Text Classification Algorithms: A Survey, DOI: 10.3390/info10040150, April 2019.

[31] Sven F. Crone, Stefan Lessmann and Rober Stahlbock (2015), Artificial Neural Network – New Potential in Data Mining for Customer Relationship Management? Lancaster University, Humboldt-University, University of Hamburg. 07 January 2015.

[32] Alexandre Kowalczyk, (2014), Support Vector Machine Succinctly", Syncfusion Inc, https://www.svm-tutorial.com/2014/10/svm-linear-kernel-good-text-classification/. 2017