A Sentiment Analysis System for the Hindi Language by Integrating Gated Recurrent Unit with Genetic Algorithm

Kush Shrivastava and Shishir Kumar
Department of Computer Science Engineering, Jaypee University of Engineering and Technology, India

Abstract: *The growing availability and popularity of opinion rich resources such as blogs, shopping websites, review portals, and social media platforms have attracted several researchers to perform the sentiment analysis task. Unlike English, Chinese, Spanish, etc. the availability of Indian languages such as Hindi, Telugu, Tamil, etc., over the web have also been increased at a rapid rate. This research work understands the growing popularity of Hindi language in the web domain and considered it for the task of sentiment analysis. The research work analyses the hidden sentiments from the movie reviews collected from the review section of Hindi language e-newspapers. The reviews are multilingual, which makes sentiment analysis a challenging task. To overcome the challenges, this research work proposes a deep learning based approach where a Gated Recurrent Unit network is combined with the Hindi word embedding model. The strategy enables the network to efficiently capture the semantic and syntactic relation between Hindi words and accurately classify them into the sentiment classes. Gated Recurrent Unit network's performance is profoundly dependent upon the selection of its hyper-parameters; therefore, this research work also utilizes a Genetic Algorithm to automatically build a gated recurrent network architecture enabling it to select the best optimal hyper-parameters. It has been observed that the proposed Genetic Algorithm-Gated Recurrent Unit (GA-GRU) model is effective and achieves breakthrough performance results on the Hindi movie review dataset as compared to other traditional resource-based and machine learning approaches.*

Keywords: *Sentiment analysis, Hindi language, multilingual, deep learning, gated recurrent unit, genetic algorithm.*

Received September 28, 2019; accepted May 9, 2020
https://doi.org/10.34028/iajit/17/6/14

1. Introduction

Since the advancement of the Internet, people nowadays commonly use blogs, forums, review portals, and social networking platforms to pour out their opinions and views. Opinion mining is a renowned strategy applied to mine the individual's opinions and reviews from the internet concerning completely different domains. It is a well-known text classification task that deals with categorizing the text based on its objective and subjective nature. Several methods and techniques are developed by researchers for improving the text classification task after understanding its role in several interesting text-based applications [35]. A well-known text classification approach in computer science is the Sentiment Analysis (SA), which studies individual's opinions and views and classifies them based on its type into the different classes. Till now SA task has been conducted on the English, Chinese, Persian, Arabic language, but there are still other languages that can be considered for the SA task. The social media platforms and other online portals have provided the provision to use Indian languages such as Hindi, Telugu, Punjabi, Tamil, Malayalam, Marathi, etc., to express the user's opinion and views on specific topics of their interest.

After Chinese, Spanish, and English language, Hindi is the primarily spoken language around the globe, with 341 million speakers where a large part of them belongs to India [7]. Nowadays, the internet has been cheaply available to every people in India that leads to the tremendous increase of web content in the Hindi language. Various news and government websites, blogs also provide content in the Hindi language. After realizing the popularity of Indians on social media, platforms such as Facebook, Twitter, and other microblogging websites have also enabled the provision of the Hindi language. The widely available Hindi content on the Internet attracted business and government organizations to analyze it and extract meaningful information for their benefits.

However, performing SA in the Hindi language is not as simple as it seems, because on the grounds the nature of Indian language shifts a lot as far as in terms of content, portrayal level, and phonetic qualities, and so on. A few significant challenges while dealing with Hindi languages are discussed beneath:

- *Word order*: word order in Hindi is different from the English language as the subject, verb, and object can come in any form [9]. Therefore, a slight variation in the order of words in Hindi text can influence the polarity of the word.

- *Word sense disambiguation*: it is well known that words have different meanings based on the context of its usage in the sentence. In the Hindi language, some words may look similar but have a different meaning. Take, for example, the word "कुल" (family ancestry/total) which can have two meanings, "वंश" may mean the family ancestry and "सब" may mean the total of some quantity.
- *Spelling variations*: in the Hindi language, there can exist different spellings for a word with the same meaning and sense. So for the better classification task, it is necessary to have all the words with the same spelling in the corpus. The spelling variation in the Hindi word सम्बन्ध (relation) has been shown in Table 1.
- *Limited resources*: the absence of well annotated corpus also makes the SA task in the Hindi language a challenging task.

Table 1. Spelling variations in the Hindi language.

Hindi sentence	**English translation**
भारत और पाकिस्तान के बीच अच्छे संबंध/ सम्बन्ध/सम्बंध नहीं हैं।	India and Pakistan do not have good relations.

The past years have witnessed few authors showing their enthusiasm for performing English to Hindi machine translation [9], Hindi word recognition system [19] and multilingual SA on Hindi language and overcome, the challenges such as word sense disambiguity [30], language structure parsing rules and Part of Speech (POS) tagging. However, there exists a minimal amount of work that deals with challenges faced in SA of the Hindi language. Lo *et al.* [17] has suggested several approaches that are divided into two types, namely, dictionary and machine learning approaches to handle these challenges. However, these approaches are based on lexicons or handcrafted features, which makes them inefficient to cope with these challenges and unable to capture the semantic meaning from the words. The lexicons need better improvement because of the lack of presence of tenses like adjectives and adverbs for a better idea about the nature of the text. The major limitation faced by the subjective lexicons is that the currently available version does not perform Word Sense Disambiguation. The most popular lexicon resource Hindi SentiWordNet (H-SWN) [8] provides information about synonyms and antonyms but lacks the information on the most commonly used senses.

Recent advancements in the subfield of machine learning called deep learning have outperformed several machine learning approaches in the field of Natural Language Processing (NLP). Deep learning is a type of neural network which consists of several hidden layers that helps the model to extract the hidden information, local and long-distance relations between the words of input data or context, and semantic meaning behind the input data. Theirs exist several deep learning techniques such as a Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) for the classification task. There also exist several words embedding models such as Word2Vec [18], Glove [22], and FastText [13], which enables the text to be represented in a vector form where words with similar context are placed together. The word representation enables the classification techniques to understand the context and semantic relation behind the word and improves the classification performance.

1.1. Contributions

This research work tried to overcome the challenges faced while dealing with Hindi text data and contributes a lot to perform and improve the SA task. Also, the utilization of deep learning and word embedding models in the Hindi language is still generally rare, which is worth to be examined. The main contribution of this research work are mentioned below:

1. Collection of Hindi movie reviews corpus from the e-newspaper sites and manual annotation of them in three sentiment classes.
2. Feature generation as low dimensional vector representation called word embeddings using FastText pre-trained Hindi word vectors.
3. GRU to capture the long term dependency between the words to classify them into positive, negative and neutral classes.
4. The possibility of learning suitable hyper parameters using the genetic algorithm for optimizing the structure of the GRU model.

The rest of the paper is organized as follows. After highlighting the contribution of the research work, section 2 introduces the status of the work done in the field of Hindi text sentiment analysis. Section 3 thoroughly explains the corpus collection and manual annotation process, while section 4 introduces the proposed GA-GRU model, which is a combination of deep learning technique and Genetic Algorithm optimization technique. The experimental setup and results are shown and discussed in section 5, and research work is summarized in section 6.

2. Status of Sentiment Analysis Work for the Hindi Language

Joshi *et al.* [12] first attempted to propose a strategy for SA in the Hindi language. Their strategy consists of in language sentiment analysis, Machine translation, and Resource-based sentiment analysis, where an H-SWN was developed from the existing English SentiWordNet (E-SWN) and achieved an overall accuracy of 78.14%. A method was proposed by combining the N-gram, and POS tagged N-gram approaches to classify Hindi reviews as positive or negative [4]. The SA on a corpus of Hindi movie and product reviews was performed in [3], where the subjective lexicon was utilized along with the N-gram approaches. The H-SWN was extended in [28] by incorporating the bootstrap approach on the existing Hindi wordnet. The approach achieved a classification accuracy of 87%. The sentiment analysis was performed in [34] on the movie reviews by a rule based and fuzzy logic approach in the document level and achieved an accuracy of 56%. H-SWN was also used for evaluating the sentiments from the Hindi movie in association with the Synset replacement algorithm in [21]. A Punjabi subjective lexicon was developed in [14] using the Punjabi dictionary and Hindi subjective lexicon and performed a sentiment analysis using Support Vector Machine (SVM), where they achieved an accuracy of 78.02%. An aspect-based SA from Hindi productreviews was conducted in [1], where SVM was used as a classifier and achieved an accuracy of 54.05%. In [25] a Multinomial Naive Bayes method applied for sentiment classification on Hindi tweets and achieved an average accuracy of 50.75%. A lexicon-based approach and machine learning approaches were used in [27] to predict the Indian election results of 2016 from tweets in the Hindi language.

In the sentiment analysis field, deep learning techniques, along with different word embedding models, has already outperformed the traditional machine learning and rule-based techniques. The few noticeable works for the SA task using deep learning techniques for English and other multilingual languages have been discussed below. A thorough comparison of deep learning algorithms like CNN, Recurrent Neural Network (RNN) and CNN with Naive Bayes (NB) was conducted in [29] for the task SA on the movie review dataset. In [31] authors compared CNN with the existing traditional approaches using the Twitter dataset and achieved an F1-score of 64.85%. A GRU was proposed in [32] to learn continuous document representations for sentiment classification. The Chinese dataset was collected in [36] for the SA task and trained on where authors found that CNN outperformed the existing techniques. Authors in [33] investigated the use of a bi-directional GRU network for sentiment analysis of Twitter data achieved 88% accuracy as compared to 73% accuracy of the CNN architecture. For Arabic sentiment analysis, two deep CNNs using the character level features were tested on the user-generated text in [20]. There model shown a 7% enhancement in accuracy as compared to logistic regression, SVM, and Naïve Bayes technique.

The first work on SA using deep learning on the Hindi language was conducted in [2]. They proposed a hybrid deep learning architecture CNN and achieved an overall accuracy of 65.96% on product review datasets. In [26], the authors proposed an RNN based approach on the data collected from Secure Anonymised Information Linkage (SAIL) 2015 for three different Indian languages. Their approach achieved an overall accuracy of 72.01%, 88%, and 65.16% for three languages. A Hindi dataset was collected and manually annotated in [24] from the online websites and newspaper and trained on CNN. The overall accuracy of 95% was achieved by the model.

3. Corpora Acquisition Phase

In total, 1352 sentences labelled as positive (822) and negative (530) sentiments collected from the available resources Indian Institute of Technology (IIT) Bombay[1] and IIT Patna[2] [2]. In addition to this, the 7000 user-written movie reviews were also manually collected from the film-sameeksha ‘film-review’ section of aajtak[3] and Jagran online newspapers[4]. Table 2 shows the sample reviews collected from the movie review websites.

[1] http://www.cfilt.iitb.ac.in/Sentiment_Analysis_Resources.html

[2] http://www.iitp.ac.in/~ai-nlp-ml/resources.html

[3] http://aajtak.intoday.in/film-review.html

[4] http://www.jagran.com/entertainment/reviews-news-hindi.html

Table 2. Sample reviews from movie review site.

Resource	Movie review	English Translations
Aajtak[3]	सिकंदर को इस तरह के किरदार मे देखना एक सुखद अनुभव रहा।	It was a pleasant experience to see Sikandar in such a character.
	केसरी मे अक्षय कुमार ने एक बार फिर अपनी ऐक्टिंग का लोहा मनवाया।	In Kesari, Akshay Kumar once again proved his acting performance.
	कमजोर कहानी की भेंट चढ़ी फिल्म, नही दिखा तिग्मांशू धुलिया का जादू।	The film was presented with a weak story, the magic of Tigmanshu Dhulia did not worked.
Jagran[4]	तकनीकी स्तर पर मिलिन्द जोन्ग की सिनेमेटो ग्राफि फिल्म को दर्शनीय बनाती है।	Milind Jong's cinematography on a technical level makes the film visible.

3.1. Sentiment Annotation Task

After collecting the movie reviews from the websites, a data annotation task was performed by the annotators. A total of three annotators who are Hindi native speakers were selected and worked independently on the corpus for a month. They were instructed to annotate the reviews in three prominent classes viz positive, negative, and neutral. The moviereview corpus was divided into three batches, where each batch was assigned to one annotator every week. Annotators were instructed to complete the task in one month and submit their annotated files after the period.

3.2. Annotators Performances

The distribution of sentiment classes by all the annotators have been shown in Table 3; it can be observed that all three annotators have precisely annotated the reviews with a minor difference. The reason behind this is, the annotators are native Hindi speakers, and they did not face any difficulty in understanding the hidden sentiments of the reviews.

Table 3. Distribution of sentiment classes w.r.t annotators.

Class Label	Annotation set 1	Ratio (%)	Annotation set 2	Ratio (%)	Annotation set 3	Ratio (%)
Positive	3110	44.3	3006	42.9	3155	45.1
Negative	2779	39.6	2903	41.5	2669	38.1
Neutral	1121	16.1	1091	15.6	1176	16.8

3.3. Inter-Annotator Agreement

The Inter-annotator agreement was applied to the manually annotated movie reviews that measure the degree to which annotators assign a similar class mark to a similar variable. For this research work, Cohen's kappa coefficient [6], is taken as a statistic metric to calculate the inter-annotator agreement using http://vassarstats.net/kappa.html. The value of Cohen's kappa for every two pairs of annotators is shown in Table 4. The average kappa value is 0.90, which shows that the annotator agreement is almost perfect using the kappa statistics scale [16].

Table 4. Inter-annotator agreement between annotators.

Annotator pairs	Observed Agreement	Agreement by chance	Kappa(k)
A1 v/s A2	0.93	0.50	0.91
A2 v/s A3	0.90	0.50	0.89
A1 v/s A3	0.94	0.50	0.92
Average	**0.92**	**0.50**	**0.90**

The annotator agreement between the annotator 1 and annotator 3 was highest and considered as the final corpus. The available dataset and manually annotated dataset were combined to use as training datasets in this research work Table 5.

Table 5. Overall movie review dataset used for training.

Source	Positive	Negative	Neutral	Total
Manually collected dataset	3155	2669	1176	7000
IIT Bombay	127	125	-	252
IIT Patna	822	530	-	1352
Total	4104	3324	1176	**8604**

4. Proposed Sentiment Analysis Framework

The general framework of the proposed sentiment analysis framework named "GA-GRU model" is shown in Figure 1, which consists of a feature enhanced word embedding module and GA-GRU module. In the rest of this section, the GA-GRU model has been elaborated in detail.

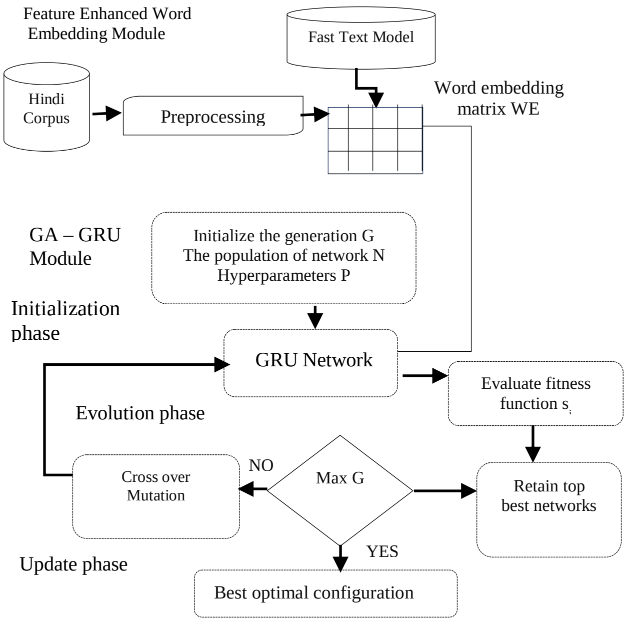

Figure 1. The framework of the proposed GA-GRU Model.

The notations used in the research work are presented in Table 6 with their proper description.

Table 6. Notations and their description.

Notations	Description
N	a population of network parameters
G	an iterative process, with the population in each iteration
p_c	probability of a selected individual to go through a crossover process (with another selected individual)
p_m	randomly selects children in the offspring's parameter and generates a new value
r	percentage of top parents in the network
U	number of hidden states, i.e., the dimension of h_t in the GRU equations
L	number of GRU hidden layers
A	introduces non-linearity into the output of a neuron
O	adjusts the weights of the GRU network to try and minimize the loss function
lr	determines how quickly or how slowly the weights are updated during the iteration
WE	Hindi Word Embedding Matrix
p	random set of hyper-parameters

4.1. Feature Enhanced Word Embedding Module

4.1.1. Pre-Processing

There exists a minimal number of tools for pre processing the Hindi text data. Therefore different tools from the literature were used in this work for the pre-processing task. The encoding format is kept Unicode Transformation Format (UTF) 8 while performing. the pre-processing task. Few reviews consist of the presence of English words that were converted to Hindi using Google Translator.

A suffix list shown in Figure 2 was created and stored in the form of a dictionary of 5, 4, 3, 2 and 1 suffix. Stemming was performed for the length of suffix 5, then for suffix 4, and so on using the lightweight stemmer [23].

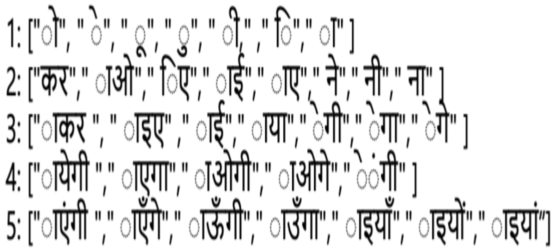
1: ["◌ो", "◌े", "◌ू", "◌ु", "◌ी", "◌ि", "◌ा"]
2: ["कर", "◌ाओ", "◌िए", "◌ाई", "◌ाए", "ने", "नी", "ना"]
3: ["◌ाकर ", " ◌ाइए", " ◌ाई", " ◌ाया", "◌ेगी", "◌ेगा", "◌ेगे"]
4: ["◌ायेगी ", " ◌ाएगा", " ◌ाओगी", " ◌ाओगे", "◌े◌ंगी"]
5: ["◌ाएंगी ", " ◌ाएँगे", " ◌ाऊँगी", " ◌ाउँगा", " ◌ाइयाँ", " ◌ाइयों", " ◌ाइयां"]

Figure 2. Suffix list for stemming.

The second phase handles the spelling variations to overcomes the challenge discussed in section 1. For example, the spelling variation for word सम्बन्ध (relation) is संबंध/सम्बन्ध . In this step, Hindi words were collected from the corpus, and the frequency of all the words was calculated and stored along with the words in a dictionary. The word with maximum frequency was chosen as the standard word among its spelling variants. For example, if the word has the highest frequency than its spelling variant, then the word सम्बन्ध becomes the standard word, and other spelling variants become the non-standard words, at last, the non-standard words were replaced with the standard word in the corpus.

The last phase breaks sentences into tokens using the IndicNLP library tool[5] on the sentence level. However, the tokens cannot be directly fed into the classification model; therefore, these tokens were further processed using a word embedding model, which allows tokens with the same syntactic and semantic meaning to have a similar vector representation. This research work utilizes a pre trained Fast Text Hindi word embedding model [10] trained on small Wikipedia data (39 million tokens) to create an embedding matrix, where the dimension of the embedding matrix is 300. The embedding matrix acts as an input to the GRU model, which helps the GRU to extract more meaningful and semantically similar sentiment features from the corpus data.

4.2. GA-GRU Module

A feature enhanced word embedding module passes its output to the GRU layer, where it learns the sentiment features. GRU is a type of recurrent neural network, which is used in sequence modeling problems and is a perfect fit for the NLP task [5]. The GRU network captures the long short term dependencies in the sentences and involves fewer parameters and is faster to run. Figure 3 shows the basic structure of the GRU network. GRU calculates updates and resets gates, which control the flow of information through each

[5] https://github.com/nisargjhaveri/indicNLP

hidden unit. The past information which contributes to the current state h_t is controlled by the reset gate r_t, whereas the past information is preserved, and the addition of new information is conducted by the update gate z_t. U and W are the weights learned by the GRU gates. At time-step t each hidden state h_t is computed using the following Equations (1), (2), (3), and (4):

Update gate:

$$z_t = \sigma\ (W^{(z)}\, x_t + U^{(z)} h_{t-1}) \qquad (1)$$

Reset gate:

$$r_t = \sigma(W^{(r)} x_t + U^{(r)}\, h_{t-1}) \qquad (2)$$

New memory:

$$h_t = \tanh\ (W x_t + r_t * U\, h_{t-1}) \qquad (3)$$

Final memory:

$$h_t = z_t * h_{t-1} + (1- z_t) * h_{t-1} \qquad (4)$$

Here * is element-wise multiplication, and σ is the sigmoid function.

The other primary task of the GA-GRU module is hyper-parameter optimization. Here, the genetic algorithm is used to optimize the GRU module's hyper-parameters to prevent overfitting and make the classification result of better. The proposed algorithm is shown in algorithm 1, which tries to evolve a GRU architecture and tune hyper-parameters listed in Table 7.

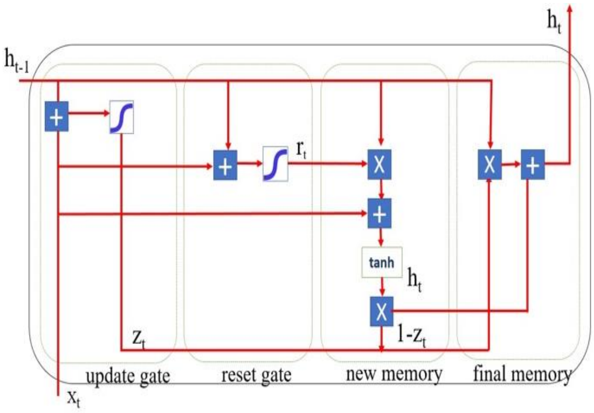

Figure 3. The structure of the GRU model.

GA-GRU consists of three phases: initialization, evaluation, and update phase:

Algorithm 1 GA-GRU algorithm

```
1:  Input: Hindi movie review dataset
2:  Initialization phase: no. of generations G, no. of networks in each generation N, probability $P_m$ and $P_c$ & % of parent networks to retain r.
3:    p ← generate a set of randomized parameters
4:  for g = 1,2,............,G do:      //Evolve the generation
5:  Evolution phase:
6:        for n = 1,2,………,N do:  // Evolve a population of network parameters
7:            perform pre-processing and remove stop words
8:      create WE ∈ $R^{n \times d}$ embedding matrix.
9:      train the GRU network using WE and p
10:   [$s_1$,$s_2$,……,$s_n$] ← fitness function (loss score) for each network.
11:   sort networks on the scores
12:   parent networks ← r% of networks retained.
      // Networks with parameters
13:   Updating phase:
14:   children= [].        // # offspring parameters
15:   for param in the parent networks do:
16:   a crossover between parent network with probability $P_c$
17:         child ← two offspring are selected from parent networks
18:   children.append(child)
19:   end for
20:   mutate some of the children with probability $P_c$
21:    return networks
22:    end for
23: end for
```

Output: a set of optimized hyper-parameters in the final generation with their loss score

- *Step* 1 Initialization Phase: in step 2-3, the parameters shown in Table 8 are initialized. In the next step, a set of hyper-parameters *p* from Table 7 are randomly selected to evaluate the network in order to obtain their fitness function values.
- *Step* 2 Evaluation Phase: this phase starts by building the GRU based on the selected random set of hyper-parameters from *p*. In step 4-9, for each population of the network, the GRU model is trained using the embeddings with *p* parameters. In steps 10-12, the genetic algorithm tries to minimize the fitness function, which is the "loss metric" in this case. All the networks are sorted by the lowest score, and the percentage of the population (parent networks) are retained after each generation. Here the parents are every network that needs to be kept for generating the offspring.
- *Step* 3 Updating Phase: finally, in the updating stage from step 13-22, the parameters of the population are updated using the cross over and mutation operator. Crossover is where the two members of a population (mother and father) are taken, and two children are generated of length equal to their parent. In the case of the GA-GRU module, the parameters from the parents are randomly separated to form the child. For instance, a child may get the "optimizer" from the female and the other parameter such as "learning rate" from the male. For the other child, the combination may be different. At last, the mutation operation is performed to mutate some of the children by setting the mutation probability rate P_m. The number of generations is used as the stop condition, and after step 23 the output of the genetic algorithm is an optimized GRU hyper-parameter with their loss score. These optimized hyper-parameters are used to evaluate the GRU network on the test data for the task of sentiment classification into positive, negative and neutral class labels.

5. Experimental Setup and Results

5.1. Handling Imbalanced Dataset

It can be observed from Table 5 that the number of instances belonging to the neutral class is quite inferior to those observed by the negative and positive instances. This gives rise to class imbalance problems, and therefore special attention has been given to overcome the problem. In this work, a random oversampling has been performed over the inferior class instances by adding the copies of instances as equal to the class with a higher number of instances. For this, a python package called "imbalanced learn" was implemented which helps to deal with the class imbalance issues. This sampling is provided on the 70% training set and 30% test set is kept as it is.

5.2. GRU Architecture and Parameters

For implementing the GA and to learn the optimized hyper-parameters, it is required to have the pre-configuration of the parameters. In total, five different parameters are taken into consideration, shown in Table 7. The parameter values are changed randomly in each generation to minimize the loss function over the training dataset. The optimizer and learning rate are selected with the help of GA from the set of parameters. The model is trained for 50 epochs with batch size kept as 128. The model is prevented from overfitting by using a drop out (0.5) regularization technique. The standard parameters of the Genetic algorithm are presented in Table 8.

Table 7. Hyper-parameters configuration for GRU model.

Hyper-parameters	# of values
U	32,64,128,512
L	2,3,4
A	relu, elu, tanh, sigmoid
O	rmsprop, adam, sgd, adagrad, adadelta
lr	1.0,.01,.001,.0001,.00001

Table 8. Parameters for genetic algorithm.

Parameters	# of values
N	10
G	5
P_c	0.3
P_m	0.1
r	0.4

The proposed GA-GRU model has been run for five generations on Kaggle's online GPU, and top five networks with their optimal hyper-parameter configuration are reported in Table 9. The top configuration where fitness value is 0.235, was selected to test the GRU model.

Table 9. Top five networks that minimized the fitness function.

Optimized hyper-parameters	The fitness function value (loss)
U: 128, L:3 ,A: relu, O: adam, lr: 0.001	0.235
U: 128, L:3 ,A: elu, O: adam, lr: 0.001	0.278
U: 64, L:2 ,A: relu, O: rmsprop, lr: 0.01	0.345
U: 64, L:2 ,A: tanh, O: rmsprop, lr: 0.01	0.395
U: 32, L:2 ,A: tanh, O: sgd, lr: 0.0001	0.423

5.3. Competing Models

A resource-based approach H-SWN lexicon developed by IIT Bombay [12] and machine learning SVM classifier reported in [3] with additional models such as naïve bayes, K-Nearest Neighbors (KNN), Decision Tree classifier is used as competing models for comparing the performance of the proposed GA-GRU model to understand its efficiency and effectiveness. Unigrams are normalized using Term Frequency-Inverse Document Frequency (TF-IDF) for machine learning classifiers and implemented using the Sklearn library. The proposed GA-GRU model has also been compared with the baseline deep learning techniques, such as CNN and LSTM. The baseline CNN reported here is based on Kim's model [15] with two 1-Dimensional convolution layers of filter size 4 and 128 filters. Each layer is followed by a relu activation function, a max-pooling layer of size 2 and 2 fully connected layers with 256 hidden units. The baseline LSTM [11] is based on with 2 layers of 128 cells, followed by a relu activation function with "return-sequences" kept as true. The optimizer in both the models is "adam" and "categorical cross entropy" a well-known loss function to deal with the multi-class problem is applied.

5.4. Experimental Results

Table 10 shows the experimental results obtained by the proposed GA-GRU model and competing models on the train set. Results suggest that the proposed GA-GRU model performs reasonably well for the Hindi movie reviews. This is because the GRU architecture is well known to capture the long term dependencies between the text data, and word embedding captures the semantic and syntactic relation among the words. For the peer competitors in the first category, GA-GRU obtains 42% improvements in terms of accuracy on the dataset over resource-based approach. The resource-based approach misses an enormous amount of hidden information and is unable to capture them effectively. Also, the context-dependency of words is not captured efficiently by these lexicons. Resource-based approaches look at the token at a time and fail to relate the token with other preceding and successive tokens.

Table 10. Comparison of approaches based on Accuracy (train set).

Approach	Accuracy
Resource-based approach	46.2 %
SVM	75.7 %
Naïve Bayes	79.3 %
KNN	69.1 %
Decision Tree	72.3 %
CNN	80.4 %
LSTM	82.2%
Proposed GA-GRU Model	**88.2 %**

Machine learning-based approaches also perform poorly and fail at many places as compared to the GA-GRU model. The reason being machine learning models ignores the context structure information from the sentences and pays more attention to word frequency features. Also, it was found that the proposed GA-GRU model trained with the Hindi word embeddings overcomes the problem faced by the unigrams. The baseline deep learning approach worked fine as compared to machine learning approaches. However, their performance does not achieve the result as achieved by the proposed GA-GRU model. CNN, along with word embeddings, is proved best to identify the hidden features and word correlation from the by text data; however, it fails to capture the long term dependencies. The LSTM performs better than CNN as it remembers the long term dependencies between the words. The GRU, however, trains faster and perform better than LSTMs on less training data. It was found that reversing the order of words in sentences improves the quality of the GRU model. The word embeddings generated from FastText captures the out of vocabulary words and enables the model to capture the semantic and context from the Hindi words across the multiple sentences. It can be concluded from the obtained results that the involvement of the FastText pre-trained vectors into the GRU network enhances the quality of the text representation for the SA task. The results obtained for the other performance metrics such as precision, recall, and F1 score are listed in Table 11.

Table 11. Comparison of approaches based on precision, recall and F1 score (train set).

Approach	Precision	Recall	F1-score
Resource based approach	0.44	0.49	0.46
SVM	0.72	0.68	0.70
Naïve Bayes	0.81	0.79	0.80
KNN	0.67	0.62	0.64
Decision Tree	0.70	0.67	0.68
CNN	0.79	0.77	0.78
LSTM	0.82	0.79	0.80
Proposed GA-GRU Model	**0.87**	**0.89**	**0.88**

Additionally, the ROC curve for the proposed GA-GRU model and each traditional machine learning classifier has been calculated and shown in Figure 4. As can be seen in the figure, the ROC curve for the proposed GA-GRU model is big as compared to the machine learning classifiers; this shows that the proposed model performs efficiently in distinguishing one class from another. The confusion matrix on the test data formed by the proposed GA-GRU model has been shown in Figure 5. It provides a better idea of what the proposed model is getting right and the types of errors it is making.

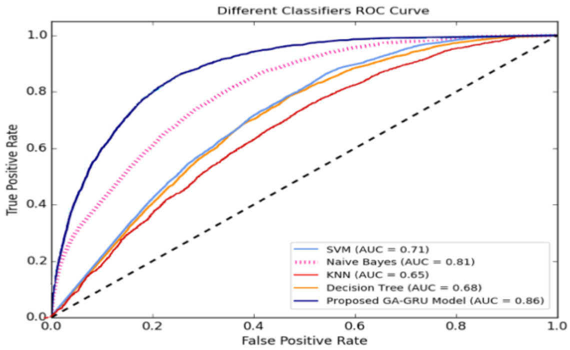

Figure 4. ROC-AUC curve for machine learning approaches and proposed GA-GRU model.

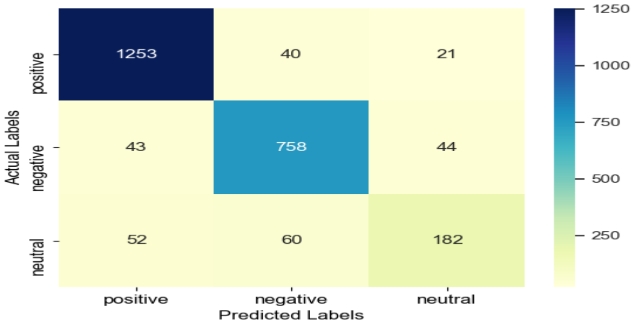

Figure 5. Confusion matrix between the actual and predicted the label for the test set.

The results in Table 12 shows that the proposed GA-GRU model performs far better than the current works on SA on Hindi movie reviews. This improvement in accuracy is because the hyper-parameters are well-tuned and prevents the model from overfitting on the dataset. The optimal hyper-parameters allows the model to generalize on the new sentences that they encounter for the very first time. Also, the word embedding model efficiently captures the sematic and contextual meaning between the words which these models fail to capture.

Table 12. Comparison between existing works and proposed GA-GRU model based on Accuracy.

Approach	Accuracy
CNN-SVM [2]	65.96 %
RNN [26]	72.01 %
Proposed GA-GRU model	**88.02 %**

5.5. Challenges Overcome

This research work encountered the challenges highlighted in section 1 and proposed a sentiment analysis framework to solve them. The way how these challenges were handled are described below:

- Handling word order: This problem has been overcome by utilizing the Hindi word embeddings.

It represents words in a coordinate system where related words based on a corpus of relationships are placed close together. Therefore if the words are not in order, the word embeddings allow them to maintain the contextual meaning of those words concerning the neighboring words. The deep learning-based GRU also allows Hindi text to keep the word order. The GRU reads the sentence from the first word of the sentence to the last word and preserves the long term dependency between the words. The output of the last words becomes the target of the SA task, i.e., the sentiment label.

- Handling word sense ambiguation: This challenge was again overcome by utilizing the word embeddings and GRU technique. To identify the meaning of a word, it is quite necessary to analyze the context in which the word appears in the sentence. Word embedding allows the GRU to understand the contexts in which a word is expected to appear in the sentence. In word embedding, if two terms are used in a similar context, the embedding model that learns the vectors (numerical representation of word) assigns them to rows that are quite similar, while words that are used in different contexts can have different vector values.
- Handling spelling variation: Spelling variation is handled efficiently during the pre-processing phase. The solution to overcome this challenge is discussed in section 4.1 in the last phase.
- Handling limited resource challenge: The challenge is overcome by manually collecting the movie reviews from the famous e-newspaper movie review sections. The collected movie reviews are manually annotated with the help of annotators. The annotation scheme used in this research work has been discussed broadly in section 3.

6. Conclusions

Over the web, an enormous amount of user-generated content in a native language such as Hindi is available, which needs to be analyzed to generate valuable information. Therefore this research work created a corpus by collecting the Hindi movie reviews from review websites and manually annotated into positive, negative, and neutral classes. This work proposed a GRU architecture designed by using an efficient genetic algorithm to classify the reviews into three separate classes. The Genetic algorithm discovered the optimal hyper-parameters for the GRU and reduced the time needed for optimization as compared to the traditional grid search approach. This work resolved the challenges faced by traditional approaches while dealing with SA in Hindi data. The semantic features have been learned through the Hindi word embeddings and trained in the GRU architecture. Through experimental evaluation, it has been shown that the proposed GA-GRU model can achieve better performance than other popular resource-based and machine learning approaches. The results are also compared with the current deep learning works used for performing the SA in Hindi data and surprisingly performs better than the existing works.

The proposed model can be deployed in the entertainment field, where it can help the filmmakers to understand the people's opinions posted by them on review portals or social media platforms. It can allow them to evaluate the performance of the movies and improve their film making and marketing strategies in the entertainment industry. This research work is restricted to classify the reviews in positive, negative and neutral classes, and however, in future, the reviews can be classified into some other fine-grained emotion classes. This research work can be extended to apply to another type of review, like product reviews, hotel reviews, music reviews, book reviews, etc. In the future, the proposed approach can also be applied to other native Indian languages apart from Hindi for performing the SA task.

References

[1] Akhtar M., Kumar A., Ekbal A., and Bhattacharyya P., "A Hybrid Deep Learning Architecture for Sentiment Analysis," *in Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers*, Osaka, pp. 482-493, 2016.

[2] Akhtar M., Ekbal A., and Bhattacharyya P., "Aspect based Sentiment Analysis in Hindi: Resource Creation and Evaluation," *in Proceedings of the 10th International Conference on Language Resources and Evaluation*, Portorož, pp. 2703-2709, 2016.

[3] Arora P., Sentiment Analysis for Hindi Language, MS Thesis, Research in Computer Science, 2013.

[4] Bakliwal A., Arora P., Patil A., and VarmaV., "Towards Enhanced Opinion Classification using NLP Techniques," *in Proceedings of the Workshop on Sentiment Analysis where AI Meets Psychology*, Chiang Mai, pp. 101-107, 2011.

[5] Cho K., Van Merriënboer B., Gulcehre, C., Bahdanau D., Bougares F., Schwenk H., and Bengio Y., "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, pp. 1724-1734, 2014.

[6] Cohen J., "A Coefficient of Agreement for Nominal Scales.," *Educational and psychological Measurement*, no. 1, pp. 37-46, 1960.

[7] Contributors W., "List of Languages by Number of Native Speakers," Available: https://en.wikipedia.org/wiki/List_of_languages_

by-number-of-native-speakers, Last Visited, 2019.

[8] Das A. and Bandyopadhyay S., "SentiWordNet for Indian Languages," *in Proceedings of the 8th Workshop on Asian Language Resources*, August, Beijing, pp. 56-63, 2010.

[9] Dwivedi S. and Sukhadeve P., "Translation Rules for English to Hindi Machine Translation System: Homoeopathy Domain," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 791-796, 2015.

[10] Grave E., Bojanowski P., Gupta P., Joulin A., and Mikolov T., "Learning Word Vectors for 157 Languages," *in Proceedings of the 11th International Conference on Language Resources and Evaluation*, Miyazaki, 2018.

[11] Hochreiter S. and Schmidhuber J., "Long Short Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[12] Joshi A., Bhattacharyya P., and Balamurali R., "A Fall-Back Strategy for Sentiment Analysis in Hindi: A Case Study," *in Proceedings of ICON 8th International Conference on Natural Language Processing*, Macmillan Publishers, 2010.

[13] Joulin A., Grave E., Bojanowski P., and Mikolov T., "Bag of Tricks for Efficient Text Classification," *in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, pp. 427-431, 2016.

[14] Kaur A. and Gupta V., "A Novel Approach for Sentiment Analysis of Punjabi Text Using SVM," *The International Arab Journal of Information Technology*, vol. 14, no. 5, pp. 707-712, 2017.

[15] Kim Y., "Convolutional Neural Networks for Sentence Classification," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, pp. 1746-1751, 2014.

[16] Landis J. and Koch G., "The Measurement of Observer Agreement for Categorical Data," *Biometrics*, vol. 33, no. 1, pp. 159-174, 1977.

[17] Lo S., Cambria E., Chiong R., and Cornforth D., "Multilingual Sentiment Analysis: from Formal to Informal and Scarce Resource Languages," *Artificial Intelligence Review*, vol. 48, no. 4, pp. 499-527, 2017.

[18] Mikolov T., Corrado G., Chen K., and Dean J., "Efficient Estimation of Word Representations in Vector Space Vector Space," *in Proceedings of the International Conference on Learning Representations*, pp. 1-12, 2013.

[19] Mittal T. and Sharma R., "Multiclass SVM Based Spoken Hindi Numerals Recognition," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 666-671, 2015.

[20] Omara E., Mosa M., and Ismail N., "Deep Convolutional Network for Arabic Sentiment Analysis for Arabic sentiment Analysis," *in International Japan-Africa Conference on Electronics, Communications and Computations*, Alexandria, pp. 155-159, 2018.

[21] Pandey P. and Govilkar S., "A Framework for Sentiment Analysis in Hindi using H-SWN," *International Journal of Computer Applications*, vol. 119, no. 19, pp. 23-26, 2015.

[22] Pennington J., Socher R., and Manning C., "GloVe : Global Vectors for Word Representation," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Doha, pp. 1532-1543, 2014.

[23] Ramanathan A. and Rao D., "A Lightweight Stemmer for Hindi," *in Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, pp. 42-48, 2003.

[24] Rani S. and Kumar P., "Deep Learning Based Sentiment Analysis Using Convolution Neural Network," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3305-3314, 2019.

[25] Sarkar K. and Chakraborty S., "A Sentiment Analysis System for Indian Language Tweets," *in Proceedings of International Conference on Mining Intelligence and Knowledge Exploration*, Hyderabad, pp. 694-702, 2015.

[26] Seshadri S., Madasamy A., and Padannayil S., "Article Analyzing Sentiment in Indian Languages Micro Text," *Institute of Integrative Omics and Applied Biotechnology*, vol. 7, no. 1, pp. 313-318, 2016.

[27] Sharma P. and Moh T., "Prediction of Indian election using sentiment analysis on Hindi Twitter," *in Proceedings of IEEE International Conference on Big Data*, Washington, pp. 1966-1971, 2016.

[28] Sharma R. and Bhattacharyya P., "A Sentiment Analyzer for Hindi Using Hindi Senti Lexicon," *in Proceedings of the 11th International Conference on Natural Language Processing*, Goa, pp. 150-155, 2014.

[29] Shirani-mehr H., "Applications of Deep Learning to Sentiment Analysis of Movie Reviews," Technical Report, Stanford University, 2015.

[30] Singh S. and Siddiqui T., "Utilizing Corpus Statistics for Hindi Word Sense Disambiguation," *The International Arab Journal of Information Technology*, vol. 12, no. 6A, pp. 755-763, 2015.

[31] Stojanovski D., Strezoski G., Madjarov G., and Dimitrovski I., "Twitter sentiment analysis using Deep Convolutional Neural Network," *in Proceedings of International Conference on Hybrid Artificial Intelligence Systems*., Bilbao,

pp. 726-737, 2015.

[32] Tang D., Qin B., and Liu T., “Document Modeling with Gated Recurrent Neural Network for Sentiment Classification,” *in Proceedings of Conference on Empirical Methods in Natural Language Processing*, Lisbon, pp. 1422-1432, 2015.

[33] Tang Y. and Liu J., “Gated Recurrent Units for Airline Sentiment Analysis of Twitter Data,” Technical Report, Stanford University, 2011.

[34] Tumsare P., Sambare A., and Jain S., “Opinion Mining In Natural Language Processing Using Sentiwordnet and Fuzzy,” *International Journal of Emerging Trends and Technology in Computer Science*, vol. 3, no. 3, pp. 153-158, 2014.

[35] Zahedi M. and Sorkhi A., “Improving Text Classification Performance Using PCA and Recall-Precision Criteria,” *Arabian Journal for Science and Engineering*, vol. 38, no. 8, pp. 2095-2102, 2013.

[36] Zhang L. and Chen C., “Sentiment Classification With Convolutional Neural Networks: an Experimental Study on A Large-Scale Chinese Conversation Corpus,” *in Proceedings of 12th International Conference on Computational Intelligence and Security*, Wuxi, pp. 165-169, 2017.

Kush Shrivastava is pursuing a Ph.D. at Jaypee University of Engineering and Technology, Guna, M.P, India. Before this, he has completed MTech in Computer Science Engineering from Jaypee University of Engineering and Technology, Guna, M.P., India.

Shishir Kumar is working as a Professor in the Department of Computer Science and Engineering at Jaypee University of Engineering and Technology, Guna, M.P., India. He earned a Ph.D. in Computer Science in 2005. He has twenty-one years of teaching experience in various organizations of repute for PG and UG courses of Computer Science and IT.