International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 16 (2024) pp. 010-019

Architecture of Cyberbullying Recognizer in Video Game Chat Using Deep Learning Model with BiLSTM

Gerson R. Carrasco Ledesma[1], H. David Calderón-Vilca [2], Manuel J. Ibarra-Cabrera[3] and Flor C. Cárdenas-Mariño [4]

[1] Software Engineering Department, Universidad Nacional Mayor de San Marcos,
Lima, Carlos Germán Amezaga #375, Perú
gerson.carrasco@unmsm.edu.pe

[2] Software Engineering Department, Universidad Nacional Mayor de San Marcos,
Lima, Carlos Germán Amezaga #375, Perú
hcalderonv@unmsm.edu.pe

[3] Informatique and Systems Department, Universidad Nacional Micaela Bastidas de Apurímac,
Abancay, Arenas 121, Perú
mibarra@unamba.edu.pe

[4] Operations research department, Universidad Nacional Mayor de San Marcos,
Lima, Carlos Germán Amezaga #375, Perú
fcardenasm@unmsm.edu.pe

***Abstract*: Cyberbullying is a social problem that has been growing in recent years through social media and other contexts such as videogames which is the most visited context by children and adolescents in these times. The solutions proposed in recent years to address the prediction of texts with cyberbullying used machine learning algorithms of classification. In view of this, we propose an architecture as a solution, using Natural Language Processing algorithms and a Deep Learning model with BiLSTM for the classification of texts in video game chats. The architecture is composed of 3 modules: Pre-processing, NLP and Clustering. The proposed architecture is sequential, the first Pre-processing module is in charge of cleaning the dataset of numbers, punctuation marks, Stop Words; the second NLP Module is in charge of generating a feature vector and a vocabulary with the Word2Vec algorithm; the third Module is in charge of classifying our dataset if it is cyberbullying thanks to the unsupervised K-means and TF-IDF algorithm; finally the Training Module is in charge of training our BiLSTM model which is composed of an Embedding layer, a BiLSTM layer, we show as a result the architecture test obtaining 97.91% accuracy.**

***Keywords*:** Artificial intelligence, Machine learning, Text analysis, Natural language processing, Classification algorithms**.**

I. Introduction

Artificial intelligence applications are in different fields, such as the recommendation of video games [1], also in the detection of violence in videos [2].

Cyberbullying has become a form of violence, or emotional or psychological abuse perpetrated against children, especially with the rise of technological advances that made the Internet one of the most important tools of information and communication. Reference [3] define cyberbullying as an aggressive and intentional act of a person or several people, using the Internet, repeatedly attempting to attack a victim who cannot easily defend themselves.

About 87% of todays' youth have witnessed some form of cyberbullying which can take various forms such as sexual harassment, hostile environment, revenge or retaliation [4]. Cyberbullying in videogames is an event that has particular characteristics with respect to cyberbullying in general, it depends on the form of communication within each game, the form of communication of some games is by text chat in which players write or express their ideas through the keyboard, in other games the form of communication is by voice in which players express their ideas by voice.

Currently, the detection of cyberbullying has been treated with artificial intelligence for the detection of abusive content using different classification techniques such as SVM [5-6], Naive Bayes, Random Forest, Logistic Regression among others. We are seeking to improve the metrics of this problem with other algorithms such as Deep Learning [7].

Detection of cyberbullying and online harassment is commonly formulated as a classification problem; however, it is inherently more difficult than mere detection of abusive content. Deep neural networks have been shown to be effective in learning nonlinear feature transformations to generate word embeddings. These word embeddings could

be beneficial for detection [8].

Reference [5] presented a text classification approach based on text chat data from a video fame called MovistarPlanet, they used algorithms such as Naive Bayes, J48 Decision Tree, Multilayerperceptron, Logical Regression, k-Nearest neighbor and SVM. They sought to detect sexual predators within the game by creating a subset of data to test different preprocessing strategies such as Bag of words, sentiment features and Rule Breaking Features.

Reference [7] used a CNN model with the objective of detecting harassment in Filipino online game chat logs. For this study, a methodology consisting of 6 steps was used: Data collection, Pre-processing, Word embedding, Model development, Evaluation, Analysis. The problem with this research is the presence of overfitting in the training.

Reference [6] trained a SVM model to predict the winning team on a set of TF-IDF based features of each word. The goal was to use the accuracy of the classifier to measure the importance of the words with respect to the match outcome. This research was not primarily aimed at recognizing sentences with Cyberbullying intent but sought to predict a winning team by taking the words in the texts as a variable.

The use of Deep learning for this type of problems has been progressively benefited along with the improvements that are being made to the current hardware to the point that cloud computing companies provide working environments ready to work with Deep Learning. That is why the deep learning model would have better results in the recognition of texts with cyberbullying than machine learning models.

In this research, a Deep Learning architecture with BiLSTM is proposed for cyberbullying recognition in video game chat texts using NLP.

The rest of this paper is organized as follows. Section 2 discusses the research related to this investigation. Section 3 includes the general architecture design of the model. Section 4 describes our evaluation and experimental results; conclusions are presented in the final part.

II. State of the Art

The review conducted for this study was categorized into two general modules: Models based on natural language processing and machine learning algorithms and Models based on machine learning; each module has two sub-modules which are studies in social media and studies in video games.

A. Models based on natural language processing and machine learning algorithms applied to social media.

Reviewing the cyberbullying detection contributions for social networks [9-12] used Twitter data, [13] used Formspring data,[14] used Kaggle data, [15] used data found on the web. However, the remaining research made use of various datasets such as [16] used Twitter and MySpace; reference [17] used data from Twitter, Formspring and Reddit; reference [18] used data from Wikipedia, Twitter, Facebook and Formspring; reference [19] used data from Facebook and Twitter. We observed that the research that made use of Twitter data were due to the fact that this social network allows obtaining data from a public api for developers making it easier to obtain information for the research.

The research of [9], [11-19] are aimed at detecting cyberbullying in texts, others like [10] their main objective is to recognize cyberbullying hotspots.

For the processing natural language processing techniques were used, [15] used web mining, stop words and tokenization, [14] used tokenization, stop word, n-gram and TF-IDF; [9] used tokenization and feature extraction, [10] used feature extraction, [11] used feature vector, [16] used semantic dropout noise to emphasize cyberbullying features in sentence, [17] used word embedding, [6] used word2vec, cosine similarity, n-gram, edit-distance and blacklist, [18] used Hierarchical Attention Models (HAN y psHAN), [19] used fastText, [13] used stop words, feature extraction, sentiment analysis and n-gram.

The techniques used by the research were: [14] used SVM and Naive Bayes, [10] used Logistic Regression, [13], [15] used SVM and neural networks, [9] made a comparison between J48, multinomial Naive Bayes, IBK and SVM, [11] used neural networks, [16] made use of smSDA, [12] made use of an unsupervised algorithm based on word2vec and cosine similarity, [17] made use of algorithms such as SVM, Naive Bayes, Random Forest and Logistic Regression and deep learning models such as CNN, LSTM, BLSTM-Attention and BLSTM, [19] made use of deep learning algorithms such as CNN, LSTM-Attention and BERT, [18] made a comparison between traditional machine learning algorithms (Lexicon Based, Linear Regression, SVM, Random Forest), deep learning(GRU, CNN), Hierarchical Attention Models (HAN, psHAN) and pre-trained transformers (BERT, DistilBERT). The presented research used evaluation metrics such as precision, accuracy, recall and F-score. The research that used precision as evaluation metric reported the following results: [9] 84% with IBk, [14] 97.11%, [15] 97%, [19] 81.16%. Regarding the F-Score the presented results were: [11] 91%, [12] 92.09% with Twitter, [13] 91.9% with neural networks, [16] 77.6% on MySpace dataset, [17] 91% with BLSTM, [18] with 97.3% and [19] 75.93% with Twitter. In [10] made a measurement with the probability that sentences contain cyberbullying according to the content, observing that if there are negative emotions or swearing, they have a percentage of 12.5% that the message is cyberbullying.

B. Models based on natural language processing and machine learning algorithms applied to video games

The reviewed research made use of different Video Game datasets: [6] used data from Dota, [7] from Ragnarok and Dota2, [20] obtained their data manually, [21] used data from League of Legends.

Research [5-7] has as main objective the detection of cyberbullying in video game chats; on the other hand, [20] sought to minimize toxic aspects in games not only with text but also wi in-game alerts, [21] sought to detect the most common uni and bi-grams among toxic players.

For data preprocessing they used natural language processing techniques: [6], [20-21] used n-gram, but in addition [6] made use of TF-IDT, [5] used feature extraction, [7] used Word embedding. Among the techniques used, [6], [20] used SVM with the difference that [6] used it for cyberbullying detection, but [20] used it to classify pings and

texts in the game, [7] used CNN, [5] used techniques like Naive Bayes, J48 Decision Tree, Multilayerperceptron, Logical Regression, IBk as k- Nearest neighbor algorithm, and Support Vector Machine, [21] did not make use of a technique as they focused their research on natural language processing. Of the reviewed research that used precision as an evaluation metric, [7] obtained 99.86%, [6] 95%, [5] 92.51%, [20] 96.2%, but the problem with the results of [7] was due to the overfitting that occurred in the training process.

C. Models based on machine learning algorithms in social media

Research [22-23] made use of public data from Instagram and Twitter respectively, [23] built their dataset, [4] used a dataset from a private game platform.

Research [22-24] had the main objective of recognizing cyberbullying unlike [25], which its main objective was to verify the validity of other studieswith other datasets.

For preprocessing [22-23] performed data cleaning and manual labeling tasks, [24] did manual labeling, while the research [23], [25], did not perform labeling because it was already implemented in previous studies in [25] and the algorithm proposed by [23] omitted this process.

In the techniques used we observed that [22], [24] proposed classification algorithms such as SVM and BPM respectively, [23] used a new Deep Learning CNN-CB algorithm which is based on CNN, in [25] used the dame methods of the studies they were reviewing, but with the necessary corrections for its correct implementation. The research that used precision as an evaluation metric reported: [22] 79.412%, [23] 93%. Research [24-25], reported other evaluation metrics, [22] reported at most 62% F1-score in one of the studies, [24] reported 85% specificity.

III. Architecture Design

In this section, the proposed architecture for the detection of cyberbullying (CB) in video game chat texts is presented. It is a Cyberbullying Recognition Model that uses Deep Learning, which is defined in 4 Modules: Pre-processing Module, NLP (Natural Language Processing), Clustering Module and Training Module.

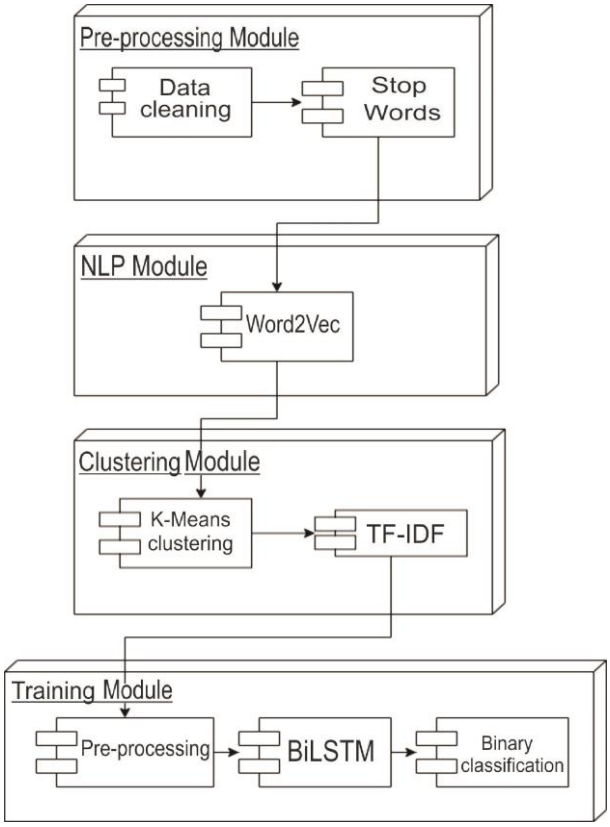


Figure 1. Solution architecture

In the Pre-processing Module the texts of the dataset are cleaned from words that do not contain meaning in the sentences. In the NLP Module, the text goes through the Word2vec component to learn the associations of the words in the corpus through the probability distribution that is generated by the Word2vec model. The Clustering Module is in charge of labeling the sentences of the dataset by the K-means algorithm and the TF-IDF process. These three modules make up the unsupervised learning section. Finally, the Training Module is in charge of classifying, by the means of the BiLSTM model, the texts to see if they contain cyberbullying or not.

A. Dataset

The Dataset named "GOSU.AI Dota 2 Game Chats" was extracted from Kaggle which is an online community of data scientists and machine learning professionals and was published 2 years ago by the user Peter Romov. This dataset contains English messages from Dota 2 game chats and was used to train a bot called Roflan. In total, it presents 21 659 448 messages from almost 1 million games. This dataset is used in this research to analyze the corpus and detect the presence of CB in video game chat messages.

B. Pre-processing module

This module processes the corpus of the "text" column of the dataset and eliminates words or text string that do not add value or meaning to the context of the sentences, the cleaned data is used in the following modules.

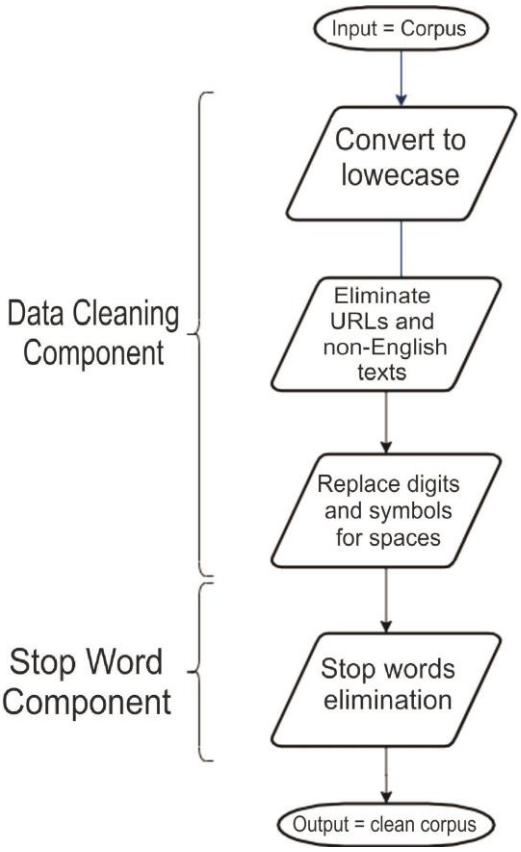


Figure 2. Pre-processing module process

1) Data Cleaning Component

This component is focused on cleaning the 'text' column of words that do not add meaning to the sentences. To start with data cleaning, all the text is transformed into lowercase and unnecessary spaces are removed. Following the above process, elements such as URLs, digits, symbols and non-English words are removed because they do not contribute any meaning to the sentences.

2) Stop Word Component

For this component a Stop Word library is loaded which provides us with a list of predetermined words in English which is the language of the dataset, the words present in this list are eliminated leaving the data with meaningful words which present CB or not CB. The result of this process is a clean corpus with meaningful words ready to pass to the next NLP Module.

C. NLP Module

This component is composed of the Word2Vec model developed by the Google research team led by Tomas Mikolov and for this study the SkipGram algorithm is used as part of the architecture of this model for the detection of CB. With the SkipGram algorithm, the corpus is used to analyze the words and predict which words will be neighbors because it is common that words containing CB are accompanied by more words expressing CB. This model uses a single hidden layer neural network, the objective is to train the neural network with the sentences of the corpus so that, given a word, it tells us the probability that each word in the vocabulary is a neighbor of the first one. Once the network is trained, we use the weights of the hidden layer, since these are the Word vectors that the model learns. For training the word2vec neural network we use word pairs found in the training data. The selection of the word pairs will depend on the Windows size (parameter that determines the number of words in the context).

The input to the neural network will be the corpus, where each word is represented as a "one hot vector", that is, a vector with as many positions as the size of the vocabulary. For example, if we want to represent a word within a corpus of 5000 words, we use a vector with a dimension of 5 000 with a value of 0 in all positions except the one corresponding to the word. The output of the neural network will be a vector with the same dimension, but with probability distribution.

The hidden layer word vectors with 300 features, which is the number of neurons, are trained, therefore, the representation matrix has X rows, one per word and 300 columns, one per neuron. 300 neurons are used in the hidden layer as this is the number of neurons used by Google in its published model.

The output layer uses a Softmax classifier that gets all values to be between 0 and 1 and the sum of all values to be 1, that is, it generates a vector of features, and which is a probability distribution that predicts which word can be the next one according to its probability, which is very convenient because the words that represent CB are accompanied by other words containing CB.

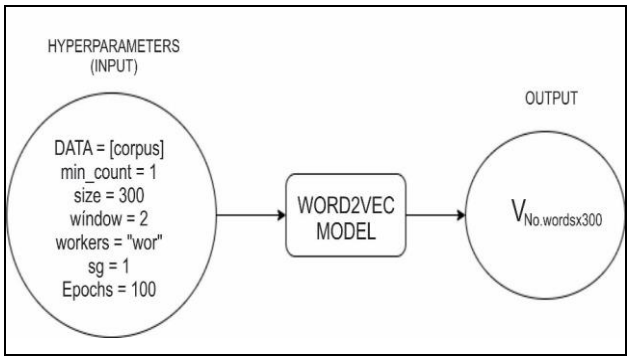


Figure 3. Hyperparameters and output of the Word2vec model

D. Clustering Module

1) K-means Clustering Component

This component makes use of the K-means clustering technique that groups words into clusters. In this study, 2 clusters are needed because it is necessary to predict whether the corpus of each sentence contains CB or not CB, and the output coordinates of the centroids of the computed clusters (center points of discovered clusters). This algorithm is iterative, in the first iteration the data of the centroid coordinates are taken randomly and adjusted until reaching the minimum value of the squared sum of distances between the points assigned of the centroids, and their centroid coordinates which mark the necessary clusters corresponding to words that contain CB and words that do not contain CB.

For the implementation of this component, the scikit-learn with its Kmeans module is used. The hyperparameters required for the implementation of this component are: "n_clusters" is the number of clusters we need which in our case is 2 because we need the clusters of messages with CB and not CB; "max_iter" is the maximum number of iterations the algorithm does, the value in this case is 100;

"random_state" is the Boolean value that tells us if the algorithm initializes the coordinates of the cluster centroids in random or initializes them manually, in this case it is set to true so that the algorithms takes a random value; "n_init" is the value of the number of times that the algorithm initializes with different centroids, in this case it is 50 and data which is the output vector that we obtained in the NLP module. After the training the model with two clusters is obtained, each cluster with the data and coordinates.

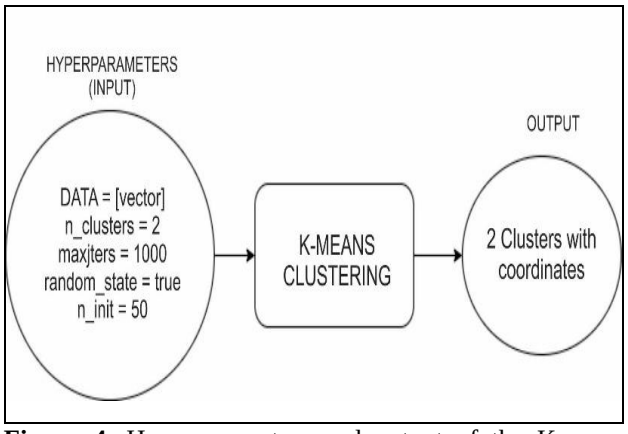


Figure 4. Hyperparameters and output of the K-means Clustering Model

2) TF-IDF Component

In this component, the TF-IDF score of each word in each sentence is calculated to consider how unique each word is in each sentence and to increase the positive or negative signal associated with words for each sentence, this in order to indicate whether the overall sentiment of each sentence is positive or negative and thus label each row of the dataset whether it is CB or not CB.

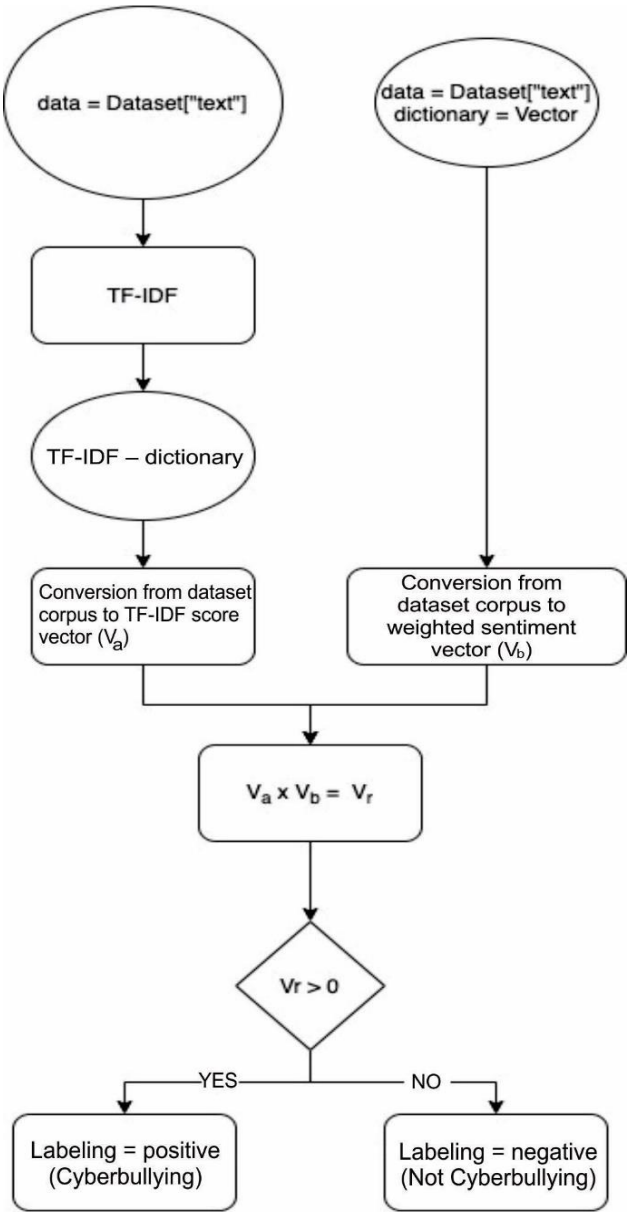


Figure 4. TF-IDF component for dataset labeling

For the implementation of this component, the scikit-learn library and its TfidfVectorizer component are used to convert each word of the corpus to its corresponding TD-IDF score. The input for this component is: the dictionary obtained in the K-means clustering component and the dataset. The dataset goes through a conversion process where each word in the dataset corpus is replaced with its corresponding TF-IDF score and by another with its corresponding weighted sentiment scores, obtaining 2 vectors for each sentence. The scalar product of these vectors if positive indicates that the sentence has positive global sentiment and if the product is negative, it indicates that the sentence has negative global sentiment. With these generated values we proceed to generate the labels for the dataset. Figure 5 shows a graphical representation of the process described above.

E. Clustering Module

1) Pre-processing Component

The labeled dataset goes through a pre-processing process similar to the Pre-processing Module. Two vectors are created, the first one contains the sentences of the dataset

after being pre-processed and the second vector is their respective labels where 0 is NOT CV and 1 CB. The vector containing the phrases is tokenized with the Tokenizer library, we take the 10 000 most common words, and the other words are represented by a meaningless text string, in this case <OOV>. This tokenization process generates a dictionary of words where the text string <OOV> is the most common one. With this dictionary, all texts in our dataset are converted to a list of sequences represented by their dictionary value.

When training neural networks for NLP it is necessary that the sequences are not of the same size, in our case we take 10 in length which is used for all the sentences, so for example if we have a sentence of 18 in length, the remaining spaces are filled with 0's. All this process mentioned above is applied for the training and validation set. The labels are converted to a matrix to be fed into the BiLSTM model.

2) BiLSTM Component

In this component the BiLSTM model is trained to predict whether the texts present CB or not. To build the BiLSTM model we used the Keras library. The model starts with an embedding layer which stores one vector per word, this layer converts the sequences of words into a sequence of vectors. After training this layer, words with similar meaning have similar vectors. The next layer is the Bidirectional wrapper which is used together with the LSTM layer (BiLSTM), this propagates the input back and forth through the BiLSTM layer and then concatenates the outputs which helps BiLSTM to learn the long-term dependencies. To regulate the overfitting in the model, a Dropout layer with a probability of 0.8 is added and then fit it to a 256-unit dense neural network with ReLU activation and then another Dropout layer with a probability of 0.8 is added. Finally, a 1-unit dense layer with Sigmoid activation is added. The general architecture of the model can be seen in Figure 6.

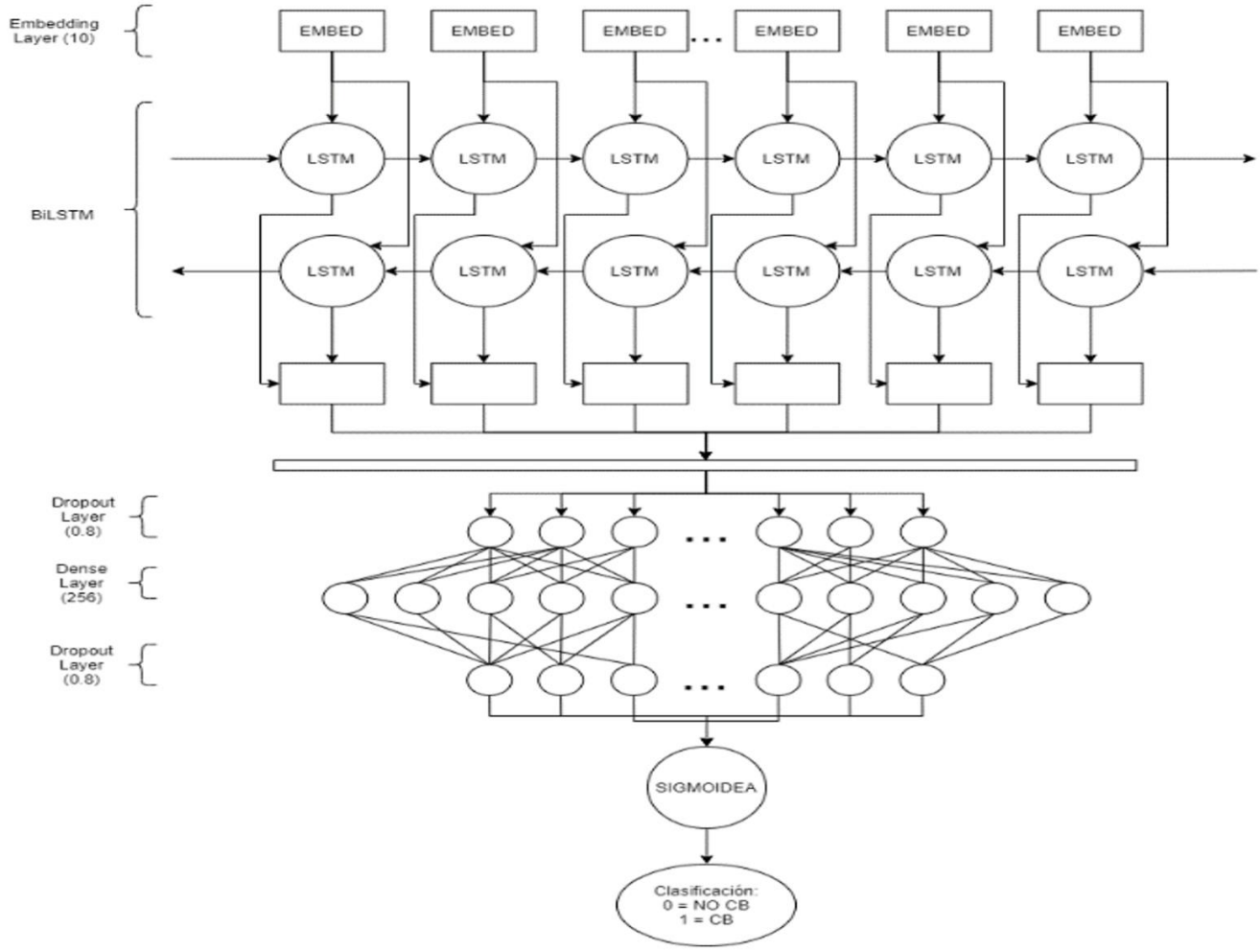


Figure 6. BiLSTM Component

3) Binary classification

The model has a binary output, that is, if the text contains CB or does not contain CB, it makes use in the output layer of a neuron with Sigmoid activation function which, having an output value on a scale between 0 and 1, makes possible the classification of the text introduced at the beginning of the model. This last layer makes use of the binary Crossentropy as a loss function which is used in problems with two possible output options as in our case.

IV. Results and Discussion

In this section we present the results of the proposed process which follows the following steps: first it goes through the Pre-processing Module, second through the

NLP Module, third through the Clustering Module and finally through the Training Module with our Deep Learning model making use of BiLSTM which classifies the chat texts in video games if they present CB or not.

The texts used for training, validation and testing of the BiLSTM model were obtained in Kaggle and each record does not present a classification that's the reason why unsupervised learning is used.

A. Dataset characteristics

The dataset obtained in Kaggle is composed of 21 659 448 English message records from untagged video game chats and the dataset contains 4 fields: 'match' which is the game identifier (int), 'time' is the time in which the message was sent (float), 'slot' is the player's slot (0-4 are Radiant team and 5-9 are Dire team) and 'text' is the message sent; also, the "text" column has texts with maximum size of 127 characters.

For the training, the following hyperparameters were used to create the model:

- Epochs: 20
- Batch size: 1000
- Optimizer: Adam
- Last layer activation function: Sigmoid
- Lost function: Binary CrossEntropy

B. Results and Evaluation of the model using BiLSTM for the classification of texts with cyberbullying

For training evaluation, the training and validation data were used to evaluate the Accuracy and the loss measured in a graph with the number of epochs vs. Accuracy and the number of epochs vs. losses achieved.

In Figure 7, the Accuracy in each epoch is evaluated, it can be observed that the accuracy of the training data reaches a level where it remains stable together with the accuracy of the validation data showing a similar pattern in both trainings which is a sign that our model did not have overtraining.

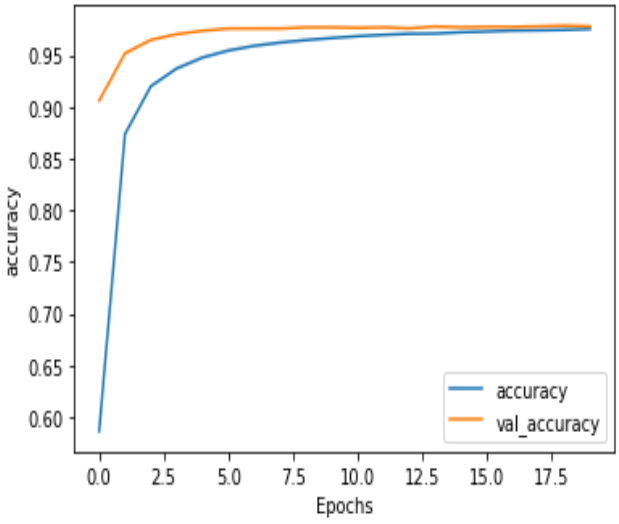


Figure 7. Accuracy evaluation for the proposed model

Figure 8 evaluates the losses in each epoch, it can be observed that the losses in the training and validation data are decreasing following a similar pattern, both reaching stability, which is an additional sign that our model does not present overtraining.

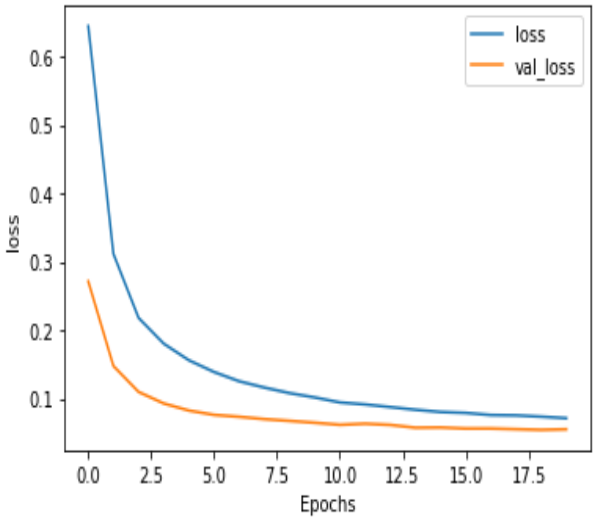


Figure 8. Reached loss evaluation for the proposed model

Table 1. Confusion matrix of the BiLSTM model

Text classification	Cyberbullying	Not Cyberbullying
Cyberbullying	23078	61
Not Cyberbullying	907	22369

From Table 1 the confusion matrix presented, we have obtained results in Table 2 which are divided into "Cyberbullying" or "Not Cyberbullying" and a weighted average of the metrics. The accuracy is 97.91%, Recall is 97.92 and F1-score is 97.91.

Table 2. BiLSTM model metrics

Text classification	Accuracy	Recall	F1-score
Cyberbullying	96.22%	99.74%	97.95%
Not Cyberbullying	99.73%	96.10%	97.88%
Average	**97.97%**	**97.92%**	**97.91%**

The model obtained 97.91% accuracy which indicates that the percentage of cases in our validation dataset was correct. The model obtained 97.97% accuracy which measures the quality of the model and shows what percentage of the texts with cyberbullying are classified as such by the model, it is the number of items correctly identified as positive (cyberbullying) out of the total number of items identified as positive.

The model obtained 97.92% recall which tells us the amount that the model is able to identify, that is, the number of items correctly identified as positive (cyberbullying) out of the total number of true positives. The model obtained a 97.91% F1 score which is the combination of accuracy and recall of the model in a single value.

C. Discussion

Wuth the results obtained with our model for cyberbullying recognition using BiLSTM, a comparative analysis is performed with research that also classified CB

texts in video games. The comparison with the metrics achieved with other studies is shown in Table 3.

In the analysis of the research presented in Table 3 the studies [5-7] sought the detection of cyberbullying or toxicity; in this research we also detected cyberbullying or toxicity in video game chats. In contrast, research [20] seeks to reduce toxicity in games by disabling chats or other tools.

Table 3. Metrics of analyzed studies

Technique	Accuracy	Precision	Recall	F1-Score
[15] n-gram, TF-IDF + SVM	-	95%	-	-
[7] Feature extraction + Perceptron Multiple	-	92.51%	-	-
[9] Word embedding + CNN	-	99.86%	-	-
[20] n-gram + SVM	-	96.2%	-	-
Proposed model Stop words, Word2vec, TF-IDF + K-means clustering + BiLSTM	**97.91%**	**97.97%**	**97.92%**	**97.91%**

Regarding the architecture of the solution and the characteristics of the data obtained, research [5-7], [20] used supervised learning methods, while our research used unsupervised learning methods because our data did not have a label that represents whether the text contains BC.

Research [7] is the only one that made use of a Deep Learning model with CNN and reported that the model achieved 99.86% Precision, however, they concluded that the model presented overfitting according to the generated metrics. The model proposed in this research also made use of a Deep Learning model with BiLSTM which, when analyzing the metrics, it can be observed that it obtained 97.97% Precision which is lower than the one presented by [7] but with the difference that our research does not show signs of overfitting thanks to the Dropout layers that were added to our model.

Research [6], [20] additionally to their models proposed an annotation system in online multiplayer game chats and an agent-based approach to defuse the negative impacts of toxic behaviors in online games respectively and this research also presents a system for cyberbullying recognition which additionally shows statistical data that serves as an aid for the parent to take future actions regarding their child's behavior.

Regarding the metrics reported by research [5], [6], [20] which reached a Precision of 95%, 92.51% and 96.2% respectively; it can be observed that the Precision of the proposed model, which reached 97.97%, is superior to the aforementioned studies.

Conclusions

In this research, an architecture based on Natural Language Processing and Deep Learning techniques was designed with BiLSTM. The proposed architecture for the detection of cyberbullying in video game chat, our results exceeded in metrics to the studies that were analyzed and that focused their research on the context of video games. This architecture can be used as a filter for video game chats, video game messaging. As future work, different clustering techniques can be tested to improve unsupervised classification and also to use other types of Deep Learning models in the training layer.

References

[1] H. Calderon-Vilca, N. M. Chavez, and J. M. R. Guimarey, "Recommendation of Videogames with Fuzzy Logic," in *Conference of Open Innovation Association, FRUCT*, 2020, vol. 2020-Septe, doi: 10.23919/FRUCT49677.2020.9211082.

[2] H. Calderon-Vilca, K. C. Ramos, E. D. Quiroz, J. A. Rojas, R. C. Vilca, and A. A. Tarqui, "The Best Model of Convolutional Neural Networks Combined with LSTM for the Detection of Interpersonal Physical Violence in Videos," in *2021 29th Conference of Open Innovations Association (FRUCT)*, May 2021, pp. 81–86, doi: 10.23919/FRUCT52173.2021.9435563.

[3] R. M. Kowalski, C. A. Morgan, and S. P. Limber, "Traditional bullying as a potential warning sign of cyberbullying," *Sch. Psychol. Int.*, vol. 33, no. 5, pp. 505–519, 2012, doi: 10.1177/0143034312445244.

[4] S. M. Kargutkar and V. Chitre, "A Study of Cyberbullying Detection Using Machine Learning Techniques," *Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020*, no. Iccmc, pp. 734–739, 2020, doi: 10.1109/ICCMC48092.2020.ICCMC-000137.

[5] Y. G. Cheong, A. K. Jensen, E. R. Gudnadottir, B. C. Bae, and J. Togelius, "Detecting Predatory Behavior in Game Chats," *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 3, pp. 220–232, 2015, doi: 10.1109/TCIAIG.2015.2424932.

[6] M. Martens, S. Shen, A. Iosup, and F. Kuipers, "Toxicity detection in multiplayer online games," *Annu. Work. Netw. Syst. Support Games*, vol. 2016-Janua, 2016, doi: 10.1109/NetGames.2015.7382991.

[7] J. A. Cornel *et al.*, "Cyberbullying Detection for Online Games Chat Logs using Deep Learning," *2019 IEEE 11th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2019*, 2019, doi: 10.1109/HNICEM48295.2019.9072811.

[8] S. Salawu, Y. He, and J. Lumsden, "Approaches to Automated Detection of Cyberbullying: A Survey," *IEEE Trans. Affect. Comput.*, vol. 11, no. 1, pp. 3–24, 2020, doi: 10.1109/TAFFC.2017.2761757.

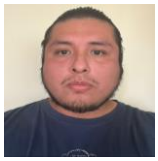
[9] S. A. Özel, S. Akdemir, E. Saraç, and H. Aksu, "Detection of cyberbullying on social media messages in Turkish," *2nd Int. Conf. Comput. Sci. Eng. UBMK 2017*, pp. 366–370, 2017, doi: 10.1109/UBMK.2017.8093411.

[10] S. M. Ho, D. Kao, M.-J. Chiu-Huang, W. Li, and C.-J. Lai, "Detecting Cyberbullying 'Hotspots' on Twitter: A Predictive Analytics Approach," *Forensic Sci. Int. Digit. Investig.*, vol. 32, p. 300906, 2020, doi:

10.1016/j.fsidi.2020.300906.
[11] A. Bozyigit, S. Utku, and E. Nasiboglu, “Cyberbullying Detection by Using Artificial Neural Network Models,” *UBMK 2019 - Proceedings, 4th Int. Conf. Comput. Sci. Eng.*, pp. 520–524, 2019, doi: 10.1109/UBMK.2019.8907118.
[12] H. S. Lee, H. R. Lee, J. U. Park, and Y. S. Han, *An abusive text detection system based on enhanced abusive and non-abusive word lists*, vol. 113. Elsevier B.V, 2018.
[13] J. Hani, M. Nashaat, M. Ahmed, Z. Emad, E. Amer, and A. Mohammed, “Social media cyberbullying detection using machine learning,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 5, pp. 703–707, 2019, doi: 10.14569/ijacsa.2019.0100587.
[14] Noviantho, S. M. Isa, and L. Ashianti, “Cyberbullying classification using text mining,” *Proc. - 2017 1st Int. Conf. Informatics Comput. Sci. ICICoS 2017*, vol. 2018-Janua, pp. 241–245, 2017, doi: 10.1109/ICICOS.2017.8276369.
[15] I. Castillo *et al.*, “Helping Students Detecting Cyberbullying Vocabulary in Internet with Web Mining Techniques,” *Proc. - 2019 Int. Conf. Incl. Technol. Educ. CONTIE 2019*, vol. 0, pp. 21–27, 2019, doi: 10.1109/CONTIE49246.2019.00014.
[16] R. Zhao and K. Mao, “Cyberbullying Detection Based on Semantic-Enhanced Marginalized Denoising Auto-Encoder,” *IEEE Trans. Affect. Comput.*, vol. 8, no. 3, pp. 328–339, 2017, doi: 10.1109/TAFFC.2016.2531682.
[17] A. Aggarwal, K. Maurya, and A. Chaudhary, “Comparative Study for Predicting the Severity of Cyberbullying Across Multiple Social Media Platforms,” *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, no. Iciccs, pp. 871–877, 2020, doi: 10.1109/ICICCS48265.2020.9121046.
[18] E. Zinovyeva, W. K. Härdle, and S. Lessmann, “Antisocial online behavior detection using deep learning,” *Decis. Support Syst.*, vol. 138, p. 113362, 2020, doi: 10.1016/j.dss.2020.113362.
[19] S. Modha, P. Majumder, T. Mandl, and C. Mandalia, “Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance,” *Expert Syst. Appl.*, vol. 161, p. 113725, 2020, doi: 10.1016/j.eswa.2020.113725.
[20] K. Watanabe and N. Fukuta, “Toward Empathic Agents for Defusing Toxic Behaviors on Team Competition Games,” *Proc. - 2017 6th IIAI Int. Congr. Adv. Appl. Informatics, IIAI-AAI 2017*, pp. 820–824, 2017, doi: 10.1109/IIAI-AAI.2017.81.
[21] H. Kwak and J. Blackburn, “Linguistic Analysis of Toxic Behavior in an Online Video Game,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8852, pp. 209–217, 2015, doi: 10.1007/978-3-319-15168-7_26.
[22] M. Andriansyah *et al.*, “Cyberbullying comment classification on Indonesian Selebgram using support vector machine method,” *Proc. 2nd Int. Conf. Informatics Comput. ICIC 2017*, vol. 2018-Janua, pp. 1–5, 2018, doi: 10.1109/IAC.2017.8280617.
[23] M. A. Al-Ajlan and M. Ykhlef, “Deep learning algorithm for cyberbullying detection,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 9, pp. 199–205, 2018, doi: 10.14569/ijacsa.2018.090927.
[24] K. Balci and A. A. Salah, “Automatic analysis and identification of verbal aggression and abusive behaviors for online social games,” *Comput. Human Behav.*, vol. 53, pp. 517–526, 2014, doi: 10.1016/j.chb.2014.10.025.

[25] A. Arango, J. Pérez, and B. Poblete, "Hate speech detection is not as easy as you may think: A closer look at model validation (extended version)," *Inf. Syst.*, 2020, doi: 10.1016/j.is.2020.101584

Author Biographies



Gerson Carrasco Ledesma Expert in PLN and Frontend development, passionate about new technologies in PLN and frontend. he developed a system for detecting cyberbylining texts in games. It currently covers frontend development roles and he is currently looking to open new projects on NLP.

Hugo D. Calderon-Vilca PhD in Computer Science, research professor of the "Artificial Intelligence" Group of the Universidad Nacional Mayor de San Marcos - Peru, advisor of undergraduate and graduate thesis projects related to Neural Networks, Machine Learning and Natural Language Processing. Professor of doctoral programs in other universities.

Manuel J. Ibarra-Cabrera PhD in Computer Science, a full-time professor at the Professional Academic School of Computer Engineering and Systems of the Micaela Bastidas National University of Apurimac – Peru. My main research area are software engineering, educational games, serious game, mobile computing and collaborative systems.

Flor Cagniy Cárdenas Mariño PhD in Computer Science, research professor of the "Optimización Matemática y Computacional" Group of the Universidad Nacional Mayor de San Marcos - Peru, advisor of undergraduate and graduate thesis projects related to artificial intelligence.