Indonesian Journal of Electrical Engineering and Computer Science
Vol. 29, No. 3, March 2023, pp. 1758~1768
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v29.i3.pp1758-1768

Text prediction recurrent neural networks using long short-term memory-dropout

Orlando Iparraguirre-Villanueva[1], Victor Guevara-Ponce[2], Daniel Ruiz-Alvarado[1], Saul Beltozar-Clemente[3], Fernando Sierra-Liñan[4], Joselyn Zapata-Paulini[5], Michael Cabanillas-Carbonell[6]
[1]Facultad de Ingeniería y Arquitectura, Universidad Autónoma del Perú, Lima, Perú
[2]Escuela de Posgrado, Universidad Ricardo Palma, Lima, Perú
[3]Dirección de Cursos Básicos, Universidad Científica del Sur, Lima, Perú
[4]Facultad de Ingeniería, Universidad Privada del Norte, Lima, Perú
[5]Escuela de Posgrado, Universidad Continental, Lima, Perú
[6]Vicerrectorado de Investigación, Universidad Privada Norbert Wiener, Lima, Perú

Article Info

Article history:

Received Aug 10, 2022
Revised Oct 25, 2022
Accepted Oct 29, 2022

Keywords:

Dropout
Prediction
Recurrent neural network
Text
Unit short-term memory

ABSTRACT

Unit short-term memory (LSTM) is a type of recurrent neural network (RNN) whose sequence-based models are being used in text generation and/or prediction tasks, question answering, and classification systems due to their ability to learn long-term dependencies. The present research integrates the LSTM network and dropout technique to generate a text from a corpus as input, a model is developed to find the best way to extract the words from the context. For training the model, the poem "*La Ciudad y los perros*" which is composed of 128,600 words is used as input data. The poem was divided into two data sets, 38.88% for training and the remaining 61.12% for testing the model. The proposed model was tested in two variants: word importance and context. The results were evaluated in terms of the semantic proximity of the generated text to the given context.

Corresponding Author:

Saul Beltozar-Clemente
Dirección de Cursos Básicos, Universidad Científica del Sur
Panamericana Sur Km 19, Villa el Salvador, Lima, Perú
Email: sbeltozar@cientifica.edu.pe

1. INTRODUCTION

In recent times, the field of natural language processing (NLP) has undergone a rapid evolution, thanks to the evolution of learning algorithms and the availability of computing resources. Recurrent neural networks (RNN) are sequence-based models for natural language understanding and word prediction. Unit short-term memory (LSTM) is a type of RNN that addresses the general problem of leakage gradient in RNNs through additional cells, input and output gates. RNNs are powerful models, which show high performance on many tasks, but they overfit quickly [1]. The lack of regulation in these models hinders data management.

Dropout is a technique to reduce overfitting in artificial neural networks (ANN), it is a more efficient way to perform model averaging with neural networks [2], [3]. Dropout is a method that deactivates a number of neurons of a neural network randomly, in each iteration of the neural network dropout deactivates different neurons, then, these deactivated neurons are not taken into account in forwarding and backward propagation. In this way, it does not force the nearby neurons to not depend so much on the deactivated neurons [1]. This rationale for Bayesian attrition proposes that the theoretical results obtained can provide insights related to the use of RNN modeling techniques.

Word embedding is a learning word embedding technique in NLP where natural language words are represented as vectors of real numbers. One of the methods for generating this mapping or representation is

NNs; dimensionality reduction with word occurrence matrices, probabilistic models, and explicit representation in terms of the context in which the words appear [4], [5].

For text prediction, the performance of RNN models is usually evaluated between non-duplication of the same text and the generation of new sentences that are syntactically sound. The proposed work aims to find the best way to extract sentence context to predict text from an input context with LSTMs supported with dropout and embedding techniques. The proposed work focuses on evaluating the semantic closeness function of the generated sentences for the new paragraph using the cosine similarity measure. Therefore, the influence of dropout and embedding techniques directly influences the performance of the model with LSTM. The proposed model will analyze the poem "La ciudad y los perros" by the writer Mario Vargas Llosa, and from this, he will predict a new paragraph.

The article is organized as follows. Section 2 describes the main works related to LSTM. Section 3 presents the method and case implementation. Section 4 describes and discusses the results and discussions. Finally, section 5 presents the conclusions.

2. RELATED WORK

Currently, NLP has made exceptional progress, which has allowed the development of models capable of performing high-level automation [6]. There is a variety of research on prediction with neural networks, several of them using existing techniques such as LSTM. Of these, some are reviewed below. Santhanam [7] proposed a model for generating text from a set of input words together with a context vector. They also proposed several methods for extracting context vectors. Similarly, Chang *et al*. [8] proposed an approach that automatically augments the data available for training by generating new text samples based on the GPT-2, and proposed to match the new text samples with the data. The Buddana *et al*. [9] created a model describing the design and operation of text generation, for which they used LSTM as an innovative solution for handling sequential data and text data in particular. Also, Shen *et al*. [10] developed a new hybrid neural network scheme based on a convolutional neural network and LSTM for multi-step wind speed prediction. The scheme was composed of two parts: a data processing module and another model module. Similarly, Abandah *et al*. [11] adopted a machine learning approach using 1,657 K verses of poems and prose to develop neural networks that automatically classify and discretize Arabic poetry. Also, Shedko [12] used an LSTM-based RNN, with the objective of generating a Byron-style poem, achieving, as a result, a guided word sequence generation based on the correspondence between the latent neural network representations and the semantic map. Berrahal and Azizi [13] use unsupervised deep learning networks to generate probabilistic images of human faces from eyewitness statements in text form to assist law enforcement in their investigations.

The LSTM neural network is applicable to different areas of industry. For example, reseachers [14]–[17] proposed an approach for better economic forecasting and decision-making by using the k-means clustering algorithm to group banks with similar price trends, then, to train these grouped stocks, they used a neural network model of short-and long-term memory (LSTM) to perform static and dynamic prediction of stock prices and economic growth rates. Lin *et al*. [18] developed a forecasting model with LSTM using decomposed data to obtain the prediction sequence to forecast the stock index price of the standard & poor's 500 indexes [19] and used the LSTM neural network model with the Adam algorithm to achieve better prediction results. So well in medicine convolutional neural networks are used for data processing, allowing to generate a prediction in time of a certain disease, as in the case of alzheimer's disease, diabetes, covid, cancer, and Parkinson [20]–[24].

Jia *et al*. [25] proposed a hybrid neural network model for labeling open relations, employing the LSTM model of ordered neurons to encode syntactic information and capture associations between arguments. To do so, they built a large-scale, high-quality, fully automated training corpus. Similarly, Javeed [26] developed a machine learning model to understand the syntax of the language and deduce the relationships between the words found, following a sequence modeling approach, in which the model receives a sequence of words and makes use of the different properties to build a hierarchical graph. Finally, Alayba and Palade [27] proposed an approach that combines convolutional neural networks (CNN) with LSTM networks to improve sentiment classification by excluding the maximum clustering layer of CNN, this layer reduces the length of the feature vectors generated after filters on the input data.

3. METHOD AND IMPLEMENTATION OF THE CASE

This section develops the terminology associated with the LSTM network, the dropout and embedding techniques, and the implementation process of the proposed case study, which aims to predict a text paragraph from the poem "La ciudad y los perros".

LSTM networks operate similarly to an RNN cell. Figure1 shows the high-level architecture and the inner workings. The LSTM network is able to add or remove information that it deems relevant to the

processing of the sequence. Compared to a basic RNN cell, the LSTM cell has an additional input and output. This additional element is known as a status cell. This state cell is the key to the operation of LSTM networks. This cell works like a conveyor belt to which data that we do not want to remain in the network can be added or removed. The LSTM network is composed of the following gates: forget gate, input gate, output gate, update gate, and update gate.

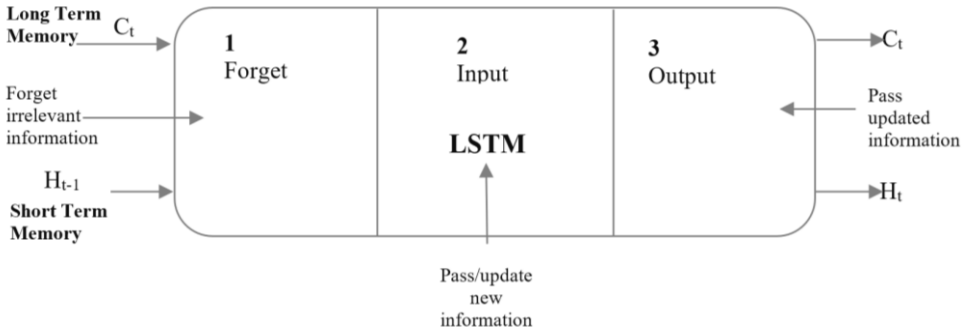

Figure 1. LSTM architecture

It is important to note that, like a basic RNN, the LSTM network also has a hidden state where H(t-1) which is represented by the previous timestamp and Ht by the current timestamp. In addition, Figure 1 shows that the state of the LSTM network cells is represented by C(t-1) and C(t), which are the timestamps, respectively. Forget gate: allows us to remove items from memory, then we move on to use the update gate, which allows us to update the memory of the LSTM cell. Next, we take the previous hidden state and the current input, then transform it and bring it back to a sigmoidal activation function:

$$f_t = \sigma\left(X_t * U_f + H_{t-1} * W_t\right) \quad (1)$$

where: Xt represents the time stamp input; U_f represents the weight associated with the input; H_{t-1} is the hidden state of the time stamp and W_f represents the weight matrix associated with the hidden state. Subsequently, a sigmoid function is applied, this causes f_t to lie between 0 and 1, then it is reproduced with the current state of the previous timestamp, as shown in the (2) and (3).

$$C_{t-1} * f_t = 0 \ldots if\ f_t = 0\ (Forget\ everything) \quad (2)$$

$$C_{t-1} * f_t = C_{t-1} \ldots if\ f_t = 1\ (Forget\ nothing) \quad (3)$$

If f_t is 0 the network will forget everything and if the value is 1, it will forget nothing.

Input gate: this gate functions as the input to the cell state and determines the information be written to the cell state, and is composed of two parts; 1) it passes the previous hidden state H_{t-1} and the current input X_t; to a sigmoid function to determine which values to update. Then, the two inputs are passed to tanh activation to regulate the network. Finally, the output tanh C_t is multiplied by the sigmoid output to decide what information is important to update the cell state. In the following, (4) is presented.

$$i_t = \sigma(X_t * U_i + H_{t-1} * W_i) \quad (4)$$

Where:
Xt: represents the input to the current timestamp t.
Ui: represents the input weight matrix.
Ht-1: is the state hidden in the previous time stamp.
Wi: is the matrix of input weights associated with the hidden state.

In (5), if the obtained value of N_t is negative, then the information has to be subtracted from the current cell state, however, if it is positive, it is returned to the final cell state.

$$N_t = tanh(X_t * U_c + H_{t-1} * W_c) \quad (5)$$

N_t, is not added directly to the state of the cell.

$$C_t = (f_t * C_{t-1} + i_t * N_t) \, (updating \; cell \; state) \quad (6)$$

Output gate: To calculate the hidden state we use the output gate, this output hidden state is simply a filtered version of the cell state just generated. What is done first is, to scale the new cell state to ensure that it is in the range -1 to 1, the tanh function is used to do this. Then, the output gate is used to determine which portions of the cell state will become part of the new hidden state. Finally, the values of the cell state are filtered with the vector generated by the output gate. The following is (7) of the hidden state calculation for prediction.

$$O_t = \sigma(X_t * U_O + H_{t-1} * W_O) \quad (7)$$

In the RNN the values are always between 0 and 1, and this is the case for the value of Ot because the sigmoid function is applied. Also, to obtain the hidden state, (7) and the updated cell state *tanh* are used. The (8) obtains the maximum score as an output for the prediction.

$$H_t = (O_t * tanh(C_t)) \quad (8)$$

It should be noted that the hidden state is a function of the inactive or secondary memory that allows storing memories for a long-term period (C_t). To define the timestamp, we take the output of the current time or we simply apply the activation of the function softMax() H_t. ***Output=Softmax (H_t)***. here, the token with the highest score at the output is the prediction.

Dropout: is a regularization technique to reduce overfitting in ANNs, it is an efficient way to perform NN model averaging. What it means is to randomly drop or omit both hidden and visible neurons during the training process of an NN. In this paper, a LSTM-dropout model is proposed to improve the text prediction fit. In turn, the model is experimented with the development of the case study. For which a number of different techniques are used. Different methods and libraries are used for character counting and training from the "one-hot" encoding, in addition, we adapt the number of parameters to improve accuracy, likewise, we use a certain number of LSTM cells so that the output layer has a better fit and is more accurate, we add dropout to avoid overfitting. Before feeding the LSTM layer, the model (LSTM-dropout) is proposed, then the input layer, the Embedding layer, the dropout layer, the LSTM layer, the dense and activation layer and the model construction are worked on.

3.1. LSTM-dropout model

The proposal is very similar to the implementation of dropout in RNNs with the same network units randomly removed. To obtain a better prediction, the network must train an iterative learning process with the data instances, where the correct output is known. It is there, where the weights associated with the input values in each layer are adjusted each time. The process is repeated with different batches of input data until the weights are adjusted well enough. This is related to the different techniques in network units which eliminates time steps and no dropout is applied in the RNNs [1]. Both RNNs and LSTMs use different gates within the RNN units per axis. Figure 2 shows that the LSTM network is defined by the gates: input, forgetting, output and modulation.

$$\begin{aligned} &\underline{i} = \text{sigm}(h_t - u_i + x_t w_i) \; \underline{f} = \text{sigm}(h_{t-1} u_f + x_t w_i) \\ &\underline{o} = \text{sigm}(h_{t-1} - u_o + x_t w_o) \; \underline{i} = \text{sigm}(h_t - u_i + x_t w_i) \\ &c_t = (f \, o \, c_{t-1} + i \, o \, g) \; h_t = (\underline{o} \, o \tanh(c_t)) \end{aligned} \quad (9)$$

With ɯ={W_i, U_i, W_f, U_f, W_o, U_o, W_g, U_g} are weight matrices and the product element. Also, the internal state C_t can be seen to be updated progressively. Likewise, the model could also be parameterized as in the work [28].

$$\begin{pmatrix} i \\ f \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \left(\left(\frac{x_f}{h_x - 1} \right) w \right) \quad (10)$$

The (10), represents a two- and four-dimensional matrix. This is called parameterization of an LSTM network until the chained weights fit well enough to predict, as compared to the LSTM of unchained weights from (9).

Although the parameterizations result in the same model, they also help to improve the accuracy, however, in order to do so, the parameters that can be adjusted must be managed very well. For example, with a very basic change of the activation function of the input layer, passing through a sigmoid, you can obtain an improvement of the pressure with the same time. One can also increase the number of times, add more neurons in a layer or add more layers. However, in these cases the improvements in accuracy penalize the execution time of the learning process, and the results may be diminished. In order to improve the accuracy, the proposal is developed using the dropout variant with the second parameterization as in (10).

$$\begin{pmatrix} i \\ f \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \left(\left(\frac{x_{fx} \, o \, Zx}{h_{x-1} o \, zh} \right) w \right) \tag{11}$$

With z_x, z_h the random masks are repeated during all training steps, in a very similar way for the parametrization of (9). Comparing with [29], the dropout variant substitutes z_x in (3), on the other hand, in (12), z^t_x depende del tiempo que se genera para cada nuevo paso de tiempo, mientras que z_h se elimina y la conexión recurrente h_{t-1} is not abandoned.

$$\begin{pmatrix} i \\ f \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} \left(\left(\frac{x_t \, o \, z^t_x}{h_{t-1}} \right) \cdot w \right) \tag{12}$$

On the other hand, the dropout variant of [24] changes (9) by adapting the inner cell, where $c_t = c_t$ o z_c. The (12) uses the same z_c mask for all the steps during processing. Whereas, in [30] considers abandonment as an action for all weights. This proposal is developed bassad in the RNNs.

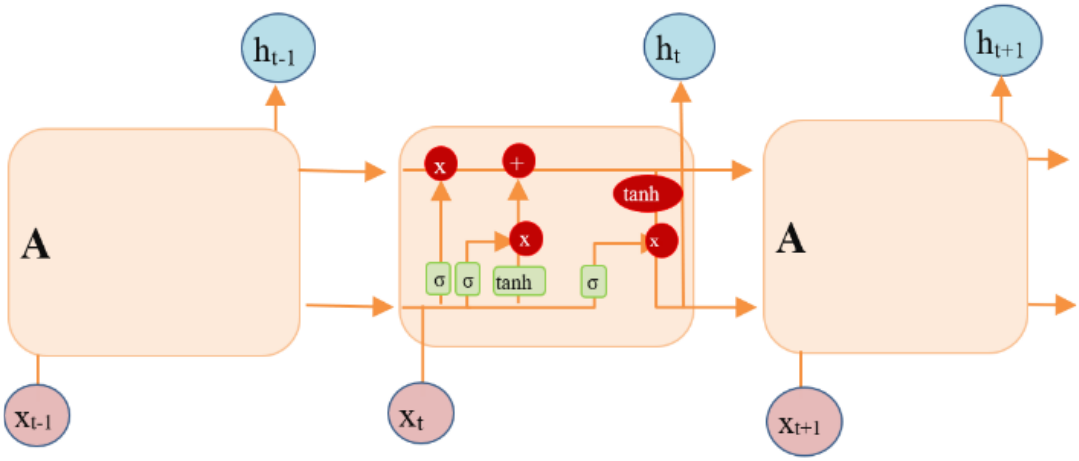

Figure 2. Enhanced LSTM-dropout approach to text prediction

3.2. Text preprocessing

For the preparation of data in the case study, we worked with the novel "*La ciudad y los perros*", by the writer Mario Vargas Llosa. The novel is composed of 128,600 words, of which 50,000 were taken for training and 78,600 for model testing. From previous research, we have found that the corpus used in the generation of new texts does not have a strong influence on the Spanish language. However, to increase the accuracy and efficiency of the model, we updated the dataset with term frequencies and inverse frequency, this is a traditional measure to determine the relevance of words in the corpus. An excerpt from the novel to be trained is shown in Figure 3.

From the frequency of words in the same corpus, the importance of the word is calculated, and the word with the highest importance is selected and chosen as the context word and with that, we start the processing, tokenization, converting the data to a token sequence, as shown in Figure 4. In this model, the context is clearly to choose the corpus keywords. The process is iterative to extract the words from the context

and is applied for all phases of training. Also, for the training of the linguistic model, it is performed with the context vectors and tested with some random words. Further on, the predictors are used as input to predict the next word. For this action, the function generate_padded_sequences() is used to tell the RNN to omit the padding tokens in our encoder. The masking explicitly forces the model to ignore certain values, such as the attention on padded elements.

3.3. Input layer

This layer has as input data the artificial neurons that incorporate a text token, which can be a complete word, a paragraph or the rot of a word [27]. At this stage of training, the tokens are displayed without pre-training, because the corpus is not yet normalized. The input layer was responsible for sending the data to the subsequent layers, where the inputs are weighted and matrices of the same size for each text. To apply word embedding, the raw data, i.e. the original data, was used. Also, three different sizes of vectors were used: 39, 100 and 150, where the input layer is represented as a matrix.

Input text: *- Sí - dice Alberto -. Te ha fregado. - Abre la boca, la cierra. Se lleva una mano a la punta de la lengua, coge con dos dedos una hebra de tabaco, la parte con las uñas se pone en los labios los dos cuerpos minúsculos y escupe. - ¿Tú no has peleado nunca, ¿no? - Sólo una vez - dice el Esclavo. - ¿Aquí? - No. Antes. - Es por eso que estás fregado - dice Alberto- Todo el mundo sabe que tienes miedo. Hay que trompearse de vez en cuando para hacerse respetar. Si no, estarás reventado en la vida. - Yo no voy a ser militar. - Yo tampoco. Pero aquí eres militar, aunque no quieras. Y lo que importa en el Ejército es ser bien macho, tener unos huevos de acero, ¿comprendes? 0 comes o te comen, no hay más remedio. A mí no me gusta que me coman. - No me gusta pelear - dice el Esclavo- Mejor dicho, no - Eso no se aprende - dice Alberto- Es una cuestión de estómago. - El teniente Gamboa dijo eso una vez. - Es la pura verdad, ¿no? Yo no quiero ……*

Figure 3. Input text: excerpt from the poem "*La ciudad y los perros*"

```
1  tokenizador = Tokenizer()
2  def secuencia_tokens(corpus):
3          tokenizador.fit_on_texts(corpus)
4          total_palabras = len(tokenizador.word_index) + 1
5          entrada_secuencia = []
6           for line in corpus:
7                        token_lista = tokenizador.texts_to_sequences([line])[0]
8                         for i in range(1, len(token_lista)):
9                                    n_gram_sequence = token_lista[:i+1]
10                                    entrada_secuencia.append(n_gram_sequence)
11          return entrada_secuencia, total_palabras
```

Figure 4. Tokenization and conversion of data to a token sequence

3.4. Embedding layer

The embedding layer is defined as the first hidden layer of the network. This layer is composed of several arguments, but there are three essential ones that are frequently used: input_dim, output_dim and input_length. This layer uses 3 arguments: Input_dim, Output_dim y Input_length: this last argument represents the length of the input sequence.

3.5. LSTM layer

The LSTM layer learns in time series and based on sequence data; it also performs additive interactions that help to improve the flow in long sequences during training. The main properties that we find in this layer are: the number of hidden cells, the default output mode, it has an indicator of input and output state to the layer, it defines the input size, with respect to activations, it has an activation function to update the hidden state cell, another function to apply to input gates, among other properties. LSTM is a feedback neural network and is mainly composed of cells. In addition, the layer is able to process single data as well as sequential data, and show better performance on sequential data, such as spoken language, video and text

processing [31]. In this model, we predict a text in the form of a row, taking into account the context of words or sequential features. This layer uses back propagation techniques in order to improve text classification.

3.6. Dropout layer

After the input layer is passed through the dropout layer to avoid overfitting in the neural network, this technique helps to select random neurons to be ignored during training, this means that neurons are temporarily removed in the next step and weight updates are not applied to the neuron in the backward step. This technique helps to regularize model learning by improving generalization techniques so that the training network can consider all inputs in the LSTM layer equally, and not focus on any specific one. This in order to avoid any bias in the training of the RNNs.

3.7. Dense and activation layers

The function of the dense layer is to integrate the outputs of the LSTMs as a single value, and it is tightly connected to the previous layer, which means that each neuron in this layer is connected to each neuron in its previous layer. The function of the activation layer is to transform the input values of the neurons, what it does, is to introduce nonlinearity into the neural networks so that it can learn the relationship between the input and output values, as in [32] used the sigmoid function for the activation layer of this model. Its output is between 0 and 1 to predict the probability of the input text.

3.8. Model construction

So far, we have prepared the training data. In this section we define the main structure of our network and, we create our model that will take the predictors as input. Note that the function embedding (), is who packs and fills our sequence and sends to the RNN, which will return a packed tensor containing all the hidden states of the sequence, finally it unpacks the sequence returning the lengths of each one. The LSTM () function calculates the output using the LSTM units and returns the hidden states of the cell. For the experiment, 150 units have been added to the layer, which can be adjusted. In this case, the dropout function is responsible for the regularization, which means that it prevents overfitting. This is done by turning off the activations of some neurons in the LSTM layer. The dense function (), is responsible for activation, where each neuron is connected to all neurons in the next layer. Table 1 shows the architecture of the model.

After building the model architecture, we can train the model using our predictors. Now it is ready to generate text. Finally, we need to write a function to predict the text based on the input text, for this case, it becomes the novel "*La ciudad y los perros*". We also have to tokenize the sequence and fill in with what we provide for training. An extract of the logic for generating text is shown in Figure 5. Finally, to make predictions with large volumes of text using the LSTM model, it is important to consider the use of a GPU, otherwise the network may simply generate useless results. RNNs can be used as generative and predictive models, and they can learn the sequences of a text and then generate completely new plausible sequences, as in the case study.

```
def generate_text(seed_text, next_words, model, max_sequence_len):
    for _ in range(next_words):
        token_list = tokenizador.texts_to_sequences([seed_text])[0]
        token_list = pad_sequences([token_list], maxlen=max_sequence_len-1, padding='pre')
        predicted=model.predict(token_list, verbose=0)
        predicted=np.argmax(predicted,axis=1)
        output_word = ""
        for word,index in tokenizador.word_index.items():
            if index == predicted:
            output_word = word
            break
         seed_text += " "+ output_word
    return seed_text.title()
```

Figure 5. Code extract to generate new text

Table 1. Model architecture

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 39, 100)	275500
lstm (LSTM)	(None, 150)	150600
dropout (dropout)	(None, 150)	0
dense (Dense)	(None, 2755)	416005
Total params: 842,105		
Trainable params: 842,105		
Non-trainable params: 0		

4. RESULTS AND DISCUSSION

In this section we present the text generated as a result and adjusted to our model. The proposed model merges LSTM and dropout, using Python as a development tool. In the last phase, the function generate_tex(), takes the initial words and the number of words to be predicted, the name of the model and the sequence. From these results, Figure 6 shows the text generated by the proposed model, where, some sentences generated match the word of the context provided. For example, the context "*Alberto*", generated sentences such as "I like the hand of the block" and "I like the hand of the block". "*mundo*" produced words around fear. "*macho*", this word is quite interesting, since it associates it with the bravery of the male. "*defenderse*", this word associates it with the ability to defend oneself from others. It is true, the generated text was not successful in all contexts, even the grammar is not coherent. The following section discusses these results in detail.

In this model, we increase the number of convolution filters and the number of LSTM cells compared to the model [33]. Also, we divided the data set into two parts: 38.88% for training and, the remaining 61.12% to evaluate the model. In addition, we ran the experiment more than once to measure the prediction accuracy. In Figure 6, the generated text is shown, in it, we will find positive sentences that are correctly predicted. We will also find words that were correctly predicted negative. At the beginning, we examined the model using similar features in terms of words that had been used in [31], and it is evident that there are slight improvements in the accuracy of the results for certain trained sets.

This paper discusses the need to apply context information to train the model for new text generation. LSTM networks have been chosen to predict new text from input and the dropout technique to reduce overfitting. The aim of this work is to find the best way to extract text from sentences, to train a text generation model, in order to obtain a better language model. The results have shown that word embeddings generated from a dataset provide better results. The proposed model used the similarity measures method to calculate the semantic closeness between the generated text and the provided context. As well as, experimentation provided feedback to the training process to know when the model over-fits and thus stop training the model.

Generated text: *me gusta la mano de la cuadra dice (**Alberto**) se mantuvo unos instantes entre la mano que no te has cuenta (**fregado**). Nunca se abandona sus ojos de la mano que me han robado un cordón grita (**Alberto**). Algo van a ser que se abandone los ojos dice el (**esclavo**) nunca salvo que la mente en se distraiga de la mano en la puerta. menos se abandona salvo muerto, se han robado mi pijama que me gusta pelear dice el (**esclavo**) lo suelta se inclina y se aleja de la puerta. La puerta de la cuadra que terminal al final, después los oficiales solo miran por la ventana y la arañan el rostro del barrio y cuando se echa a llorar enfrente de todo el (**mundo**) sabe que tiene miedo. las manos están amarilletas en la oscuridad del recinto, cava comprobó que el jaguar estaba descalzo, eran grandes sus pies, de uñas largas y sucias, olían mal rompí un vidrio dijo sin levantar la voz, la mano de la cuadra (**Alberto**) va a la litera de la estatua que está cerca. El (**militar**) acaba de romper el silencio que los aterroriza», la tia Adelina se moria de miedo. Él se moría de ganas por gritar como un niño, para que la luz de la vida volviera a ese cuarto, donde todo parecía muerto. Pero el esclavo tenía que (**defenderse**) como (**macho**) o morir. Hace frio te hace falta un abrigo, dice el esclavo sin moverse. Los dedos de (**Alberto**) reposan sobre la mano del (**esclavo**). Arrastrando los pies cruza la calle del cuartel con las manos extendidas pidiendo ayuda. .....*

Figure 6. Text generated with LSTM-dropout

5. CONCLUSION

In this paper, the effectiveness of the LSTM model for text generation was studied for the purpose of better representing text features. In addition, the semantics of the words in the novel were examined. "La ciudad y los Perros" using different embedding techniques with different dimensions. The effectiveness of LSTM networks was also studied. We propose a new model that integrates LSTM networks and dropout technique to improve the text prediction fit. Furthermore, in the model we use multiple word embedding methods using the same words from the corpus to measure the influence of a variety of word representations on word

classification. Specifically, the LSTM model is used to learn from long-term temporal dependencies, in addition, the neural network is fully connected and modeled to achieve nonlinear regression. While dropout technique, is used to reduce the overfitting in ANN, it is a more efficient way to perform model averaging with RNN. The results of the proposed model show that it can successfully predict text from an input context. In contrast to other models such as a convolutional neural network and LSTM, compared using the same input context, the proposed model has higher accuracy and slightly outperforms the comparative models.

Future work should consider more complex models than the LSTM network or other alternative machine learning algorithms. In addition, to obtain higher accuracy in data analysis, it is necessary to use larger volume of data and process it with more complex neural networks. The proposed model can also be adapted for different types of question answering systems, where, instead of following a template-based approach, the proposed model could identify the context of the question and answer meaningfully in relation to the question. It could also be adapted to track text in chat-bots. In addition, conversations could be generated based on previously conversed and learned texts. Also, they could be used as linguistic models instead of memory-based networks. Other clustering techniques, such as the spectral technique, could be experimented with for text extraction and dimensionality reduction.

REFERENCES

[1] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," Dec. 2015, [Online]. Available: http://arxiv.org/abs/1512.05287.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Oct. 2013, [Online]. Available: http://arxiv.org/abs/1310.4546.

[3] R. Lebret and R. Collobert, "Word embeddings through hellinger PCA," Dec. 2013, doi: 1312.5542.

[4] S. Patankar, M. Phadke, and S. Devane, "Wiki sense bag creation using multilingual word sense disambiguation," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 1, pp. 319–326, Mar. 2022, doi: 10.11591/ijai.v11.i1.pp319-326.

[5] T. U. Tran, H. T. Thi Hoan, P. H. Dang, and M. Riveill, "Toward a multitask aspect-based sentiment analysis model using deep learning," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 11, no. 2, pp. 516–524, Jun. 2022, doi: 10.11591/ijai.v11.i2.pp516-524.

[6] I. Dhall, S. Vashisth, and S. Saraswat, "Text generation using long short-term memory networks," in *Micro-Electronics and Telecommunication Engineering. Lecture Notes in Networks and Systems*, Singapore: Springer, 2020, pp. 649–657.

[7] S. Santhanam, "Context based text-generation using LSTM networks," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2005.00048.

[8] E. Chang, X. Shen, D. Zhu, V. Demberg, and H. Su, "Neural data-to-text generation with LM-based text augmentation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 758–768, doi: 10.18653/v1/2021.eacl-main.64.

[9] H. V. K. S. Buddana, S. S. Kaushik, P. Manogna, and S. K. P.S., "Word level LSTM and recurrent neural network for automatic text generation," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2021, pp. 1–4, doi: 10.1109/ICCCI50826.2021.9402488.

[10] Z. Shen, X. Fan, L. Zhang, and H. Yu, "Wind speed prediction of unmanned sailboat based on CNN and LSTM hybrid neural network," *Ocean Engineering*, vol. 254, p. 111352, Jun. 2022, doi: 10.1016/j.oceaneng.2022.111352.

[11] G. A. Abandah, M. Z. Khedher, M. R. Abdel-Majeed, H. M. Mansour, S. F. Hulliel, and L. M. Bisharat, "Classifying and diacritizing Arabic poems using deep recurrent neural networks," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3775–3788, Jun. 2022, doi: 10.1016/j.jksuci.2020.12.002.

[12] A. Y. Shedko, "Semantic-map-based assistant for creative text generation," *Procedia Computer Science*, vol. 123, pp. 446–450, 2018, doi: 10.1016/j.procs.2018.01.068.

[13] M. Berrahal and M. Azizi, "Optimal text-to-image synthesis model for generating portrait images using generative adversarial network techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, p. 972, Feb. 2022, doi: 10.11591/ijeecs.v25.i2.pp972-979.

[14] S. Park and J.-S. Yang, "Interpretable deep learning LSTM model for intelligent economic decision-making," *Knowledge-Based Systems*, vol. 248, p. 108907, Jul. 2022, doi: 10.1016/j.knosys.2022.108907.

[15] N. Bacanin, M. Sarac, N. Budimirovic, M. Zivkovic, A. A. AlZubi, and A. K. Bashir, "Smart wireless health care system using graph LSTM pollution prediction and dragonfly node localization," *Sustainable Computing: Informatics and Systems*, vol. 35, p. 100711, Sep. 2022, doi: 10.1016/j.suscom.2022.100711.

[16] Y. Chen, J. Wu, and Z. Wu, "China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach," *Expert Systems with Applications*, vol. 202, p. 117370, Sep. 2022, doi: 10.1016/j.eswa.2022.117370.

[17] O. Iparraguirre-Villanueva, V. Guevara-Ponce, F. Sierra-Linan, S. Beltozar-Clemente, and M. Cabanillas-Carbonell, "Sentiment analysis of tweets using unsupervised learning techniques and the k-means algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130669.

[18] Y. Lin, Y. Yan, J. Xu, Y. Liao, and F. Ma, "Forecasting stock index price using the CEEMDAN-LSTM model," *The North American Journal of Economics and Finance*, vol. 57, p. 101421, Jul. 2021, doi: 10.1016/j.najef.2021.101421.

[19] Y. Li and C. You, "Adaptive trading system based on LSTM neural network," *Journal of Physics: Conference Series*, vol. 1982, no. 1, p. 012091, Jul. 2021, doi: 10.1088/1742-6596/1982/1/012091.

[20] M. T. Younis, Y. T. Younus, J. N. Hasoon, A. H. Fadhil, and S. A. Mostafa, "An accurate Alzheimer's disease detection using a developed convolutional neural network model," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2005–2012, Aug. 2022, doi: 10.11591/eei.v11i4.3659.

[21] T. A. Assegie, T. Karpagam, R. Mothukuri, R. L. Tulasi, and M. Fentahun Engidaye, "Extraction of human understandable insight from machine learning model for diabetes prediction," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 1126–1133, Apr. 2022, doi: 10.11591/eei.v11i2.3391.

[22] M. Alagarsamy, K. Anbalagan, Y. Thangavel, J. Sakkarai, J. Pauliah, and K. Suriyan, "Classification of covid patient image dataset using modified deep convolutional neural network system," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 2273–2279, Aug. 2022, doi: 10.11591/eei.v11i4.3290.
[23] R. R. Kadhim and M. Y. Kamil, "Comparison of breast cancer classification models on Wisconsin dataset," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 11, no. 2, pp. 166–174, Jul. 2022, doi: 10.11591/ijres.v11.i2.pp166-174.
[24] H. A. A. El Aal, S. A. Taie, and N. El-Bendary, "An optimized RNN-LSTM approach for parkinson's disease early detection using speech features," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2503–2512, Oct. 2021, doi: 10.11591/eei.v10i5.3128.
[25] S. Jia, E. Shijia, L. Ding, X. Chen, and Y. Xiang, "Hybrid neural tagging model for open relation extraction," *Expert Systems with Applications*, vol. 200, p. 116951, Aug. 2022, doi: 10.1016/j.eswa.2022.116951.
[26] A. Javeed, "An LSTM model for extracting hierarchical relations between words for better topic modeling," *Journal of Physics: Conference Series*, vol. 1780, no. 1, p. 012019, Feb. 2021, doi: 10.1088/1742-6596/1780/1/012019.
[27] A. M. Alayba and V. Palade, "Leveraging Arabic sentiment classification using an enhanced CNN-LSTM approach and effective Arabic text preparation," *Journal of King Saud University - Computer and Information Sciences*, Dec. 2021, doi: 10.1016/j.jksuci.2021.12.004.
[28] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: representing model uncertainty in deep learning," Jun. 2015, doi: 10.48550/arxiv.1506.02142.
[29] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.2329.
[30] T. Moon, H. Choi, H. Lee, and I. Song, "RNNDROP: a novel dropout for RNNS in ASR," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 65–70, doi: 10.1109/ASRU.2015.7404775.
[31] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient Subtyping via Time-Aware LSTM Networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2017, pp. 65–74, doi: 10.1145/3097983.3097997.
[32] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *From Natural to Artificial Neural Computation. IWANN 1995. Lecture Notes in Computer Science*, Springer, Ed. Berlin, 1995, pp. 195–201.
[33] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," in *Machine Learning and Knowledge Extraction. CD-MAKE 2018. Lecture Notes in Computer Science()*, Cham: Springer, 2018, pp. 179–191.

BIOGRAPHIES OF AUTHORS

Orlando Iparraguirre-Villanueva Systems Engineer with a master's degree in Information Technology Management, PhD in Systems Engineering from Universidad Nacional Federico Villarreal-Peru. ITIL® Foundation Certificate in IT Service, Specialization in Business Continuity Management, Scrum Fundamentals Certification (SFC). National and international speaker/panelist (Panama, Colombia, Ecuador, Venezuela, Mexico). Undergraduate and postgraduate lecturer in different universities in the country. Advisor and jury of thesis in different universities. Consultant in information technologies in public and private institutions. Coordinator, Director in different private institutions. Specialist in software development, IoT, Business Intelligence, open-source software, Augmented Reality, Machine Learning, text mining and virtual environments. He can be contacted at email: iv.orlando.c@gmail.com.

Víctor Guevara-Ponce Statistical engineer with a master's degree in university education, graduated with a master's degree in data science from the Ricardo Palma University and applied statistics from the La Molina National Agrarian University-Peru. Specialization in data science ITAM and CIMAT Mexico, Undergraduate and Postgraduate teacher at the National University of San Marcos, Ricardo Palma University and Jose Faustino Sanchez Carrión National University-Peru, Consultant on topics related to artificial intelligence, machine learning, deep learning, and science of data. He can be contacted at email: victor.guevarap@urp.edu.pe.

Daniel Ruiz-Alvarado Systems Engineer with a master's degree in University Teaching and Educational Management, studies for a master's degree in Management and Administration in Information and Communication Technologies from Universidad Nacional de Trujillo-Perú. ITIL® Foundation Certificate in IT and Certificate in information security Based on ISO/IEC 27002 by Official Creditor of the EXIN Company. University teacher in different universities of the country. Consultant in information technologies in public and private institutions. Specialist in open-source software development and virtual environments. He can be contacted at email: druiz.alvarado@gmail.com.

Saúl Beltozar-Clemente Bachelor's Degree in Mathematics and Physics, Master's Degree in University Teaching from the National University of Education UNE-Peru. Certification in Hybrid Teaching from the University of Monterrey-Mexico. Undergraduate teaching at Universidad Científica del Sur, Universidad Privada del Norte, Universidad Tecnológica del Perú. Consultant in information technologies in public and private institutions focused on education. He can be contacted at the following e-mail address: saulbelto@gmail.com.

Fernando Sierra-Liñan Mg. Fernando Sierra-Liñan has a Bachelor's degree in Education, specializing in Science and Technology at USIL, a Master's degree in Edumatics and University Teaching at UTP, a Bachelor's degree in Systems Engineering and Computer Science at UTP, with a technical specialty in Computer Science and Computer Science. He is currently working as a researcher and thesis advisor in the faculty of Computer Engineering and Systems at the Universidad Privada del Norte, Lima-Peru. He has 20 years of teaching experience. His areas of interest are programming, database and data analysis. He can be contacted at email: fernando.sierra@upn.edu.pe, pfsierra.D02052@gmail.com.

Joselyn Zapata-Pauilini Bachelor in Systems Engineering and Computer Science from the Universidad de Ciencias y Humanidades, Master in Science with environmental management and sustainable development at the Universidad Continental, Peru. She has several international publications. Specialized in the areas of augmented reality, virtual reality and internet of things. Author of scientific articles indexed in IEEE Xplore, Scopus and WoS. She can be contacted at email: 70994337@continental.edu.pe.

Michael Cabanillas-Carbonell is Engineer and Master in Systems Engineering from the National University of Callao-Peru, PhD candidate in Systems Engineering and Telecommunications at the Polytechnic University of Madrid. President of the chapter of the Education Society IEEE-Peru. Conference Chair of the Engineering International Research Conference IEEE Peru EIRCON. Research Professor at Norbert Wiener University, Professor at Universidad Privada del Norte, Universidad Autónoma del Perú. Advisor and Jury of Engineering Thesis in different universities in Peru. International lecturer in Spain, United Kingdom, South Africa, Romania, Argentina, Chile, China. Specialization in Software Development, Artificial Intelligence, Machine Learning, Business Intelligence, Augmented Reality. Reviewer IEEE Peru and author of more than 25 scientific articles indexed in IEEE Xplore and Scopus. He can be contacted at mcabanillas@ieee.org