METHODOLOGICAL PROPOSAL FOR THE CLASSIFICATION OF TEXTS WITH CONVOLUTIONAL NEURAL NETWORKS.

Author Evelyn Guindo Betancourt[1] and co Author Armando Plasencia Salgueiro[2]

[1]Department of Applications, Refinery, Cienfuegos, Cuba

[2]Institute of Cybernetics, Mathematics and Physics, Havana, Cuba

ABSTRACT

In this article a methodological proposal for the classification of texts is presented. The main objective is to provide guidance to new researchers, especially Spanish-speaking researchers, who are beginning in the world of machine learning, specifically deep learning using convolutional neural network techniques applied to natural language. It addresses concepts and explanations for a good understanding of the subject. The proposed methodology consists of 6 stages that were carefully carried out in a project to classify scientific texts. The proposal takes as reference the CRISP-DM methodology and the KDD process, in addition to the experiences of projects and scientific articles that apply convolutional neural networks to the processing of natural language, written by elite authors on the subject such as: Yan LeCun, Xiang Zhang and Geoffrey Hinton. The texts used in the training and test of the classifier is a corpus of abstracts of theses of degree, masters and doctorate in the field of computer science, previously classified by specialists *in 11 classes of computer subjects. Each summary was transformed into a vector matrix with the characteristic that the representation was elaborated at the character level from the Python programming language and the libraries for the processing of the language, also with the help of the MXNET framework a convolutional network was designed and then* trained *and get the text classification model.*

KEYWORDS

Methodology, text mining, text classification, deep learning, convolutional neural networks.

1. INTRODUCTION

In recent years positive results have been achieved, seen in projects for the processing of natural language that includes the classification of the text, the analysis of the feeling, the translation of the language, as well as the recognition of speech with personal assistants. These advances are mainly due to the application of a very fashionable technique in artificial intelligence, the Deep Learning.

Deep learning is a way of referring to the simulation of networks of neurons that "learn" to recognize, recognize language and make decisions. These networks do not exactly mimic the

functioning of the brain. Instead, they are based on general mathematical principles, from examples, allows them to learn [8].
The appearance of deep learning algorithms, as a result of the development of automatic learning algorithms and, in particular, of neural networks, has led researchers to think about their application in the classification of texts to improve, mainly, the accuracy of results and their semantic understanding [25].

Yoav Goldberg, in his manual on Deep Learning for the processing of natural language, explains that neural networks in general offer better performance in relation to classical linear classifiers.
Although deep learning has been applied to text mining, there is no clear methodology on how to proceed with the combination of the deep learning technique and the applicable field (in this case, text mining). The procedures, methodology or existing guides are general data mining projects, without addressing technical aspects such as: definition of the architecture of the network to be used or the configuration of the development environment.
The present research proposes a methodology of how to apply convolutional neural networks as part of deep learning in problems of classification of scientific texts. From a study of the art of topics such as: deep learning, text mining as a subset of data mining, CRISP-DM methodology, knowledge discovery process, natural language processing techniques, neural networks and convolutional neural networks, the MCTexto methodology and a classification model of scientific texts are proposed.
MCTexto has 6 stages: stage zero, initial stage, text processing, architecture definition and training of the convolutional neuronal network, validation and interpretation of the classification model and as a last stage application of the model.
Figure 1 explains the relationship of the general concepts taken into account for the design of the proposed methodology. Data mining as a macro process of research and, within this, text mining as a main component of the methodological proposal. The classification of texts constitutes a type of application of text mining and it can be done using several traditional techniques such as Bayesian, decision tree, vectorial machine support among others, however, a less traditional technique such as neural networks was used convolutional.

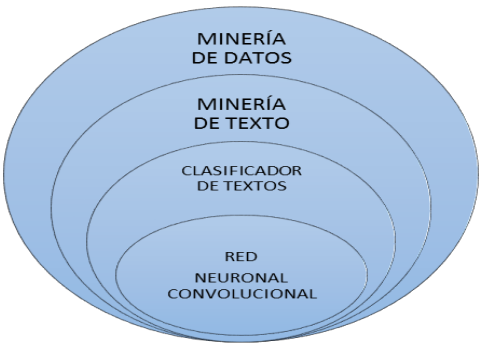


Figure 1. General Concept Relationship employees
for the methodological proposal.

The article is organized by sections, in 2 it shows the mining of texts, the methodologies and processes that exist in this discipline. Section 3 consists of aspects related to the classification of texts and the convolutional neuronal networks technique. Section 4 consists of the proposed methodology and section 5 experiments and results is the application of the methodology proposed in a case study. Finally, the conclusions and bibliography.

2. Text Mining and Methodologies

Text mining, being a subset of data mining, adopts automatic learning techniques for identifying patterns and understanding new information. [1]

The goal of text mining is the discovery of new information from collections of unstructured text documents. By unstructured we refer to free text, usually in natural language, although it could also be source code or other textual information. The most common mining task on these data is the categorization, classification and grouping of the texts. [14]

Text mining and text analysis are general terms that describe a range of technologies for analyzing and processing semi-structured and unstructured text data [5] [18]. Behind each of these technologies is the need to convert text into numbers so that powerful algorithms can be applied to large document databases [5]. Converting text into a structured numerical format and applying analytical algorithms requires knowing how to use and combine techniques to handle text, from individual words to documents to complete document databases [5].

Gary Miner in his book "Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications" It establishes that it is important to collect, organize, analyze and assimilate information. Miner proposes three different activities with subtasks depending on the information that you want to generate. In this book a detailed diagram for text mining is proposed. The first activity comes from a collection of documents. Texts with specific domain must be collected and organized. The corpus of the collection is established. The second activity is about pre-processing the data or structuring the data. This second activity is responsible for introducing a structure to the corpus from stage 1. To conclude, knowledge is extracted. This last activity is responsible for discovering the patterns of the data previously processed. At this stage, feedback can be provided with the first and second activity by providing corrections and / or adjustments. The patterns and associations are represented and visualized. [5]

Text mining approaches are related to traditional data mining and knowledge discovery methods, with some specificities. [3]. Therefore, it was decided to adapt one of the most used methodologies in data mining CRIPS-DM and the KDD process, and thus propose a specific methodology for text classification using the deep learning technique, for its favorable results in this field.

In the following subsections, the stages and phases of both processes are detailed, based on the information obtained in the book "Introduction to Data Mining" by author Orlando Hernández.

2.1. Knowledge Discovery Process

The knowledge discovery process in the database includes several phases, visualized in Figure 2. Data mining is only an essential phase in the process.

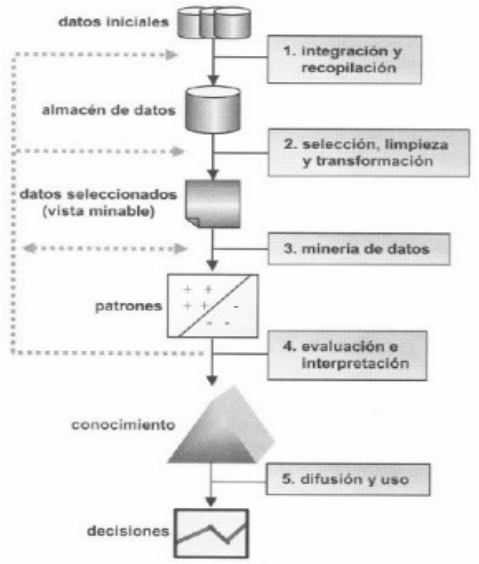


Figure 2. Phases of the KDD Process [14].

Integration phase and data collection determine the sources of information that can be useful and where to get them. Then all the data are transformed into a common format, often through a data warehouse that manages to unify all the collected information in an operative way, detecting and resolving the inconsistencies, as well as facilitating the visualization of the data. In the second **phase, selection, cleaning and transformation** the incorrect data are eliminated or corrected and the strategy to be followed with the incomplete data is decided, in addition to selecting the data that are part of the minable view. The **data mining phase** is the application of automatic learning techniques according to the data mining task to be used and, as a result, descriptive or predictive models are obtained. The **evaluation and interpretation phase** it allows to evaluate the patterns by means of evaluation techniques and they are analyzed by the experts, in case of being necessary it goes back to the previous phases for a new iteration. The **diffusion phase** makes use of the new knowledge for all possible users.

2.2. CRISP-DM

It is a process standard freely available to solve general problems of business strategies or research within data mining. [12]

Standard 1.0 of the CRISP-DM methodology includes a reference model and a guide to carry out a data mining project [14]. Both are structured in six main phases as seen in Figure 3, where some are feedback with another phase.

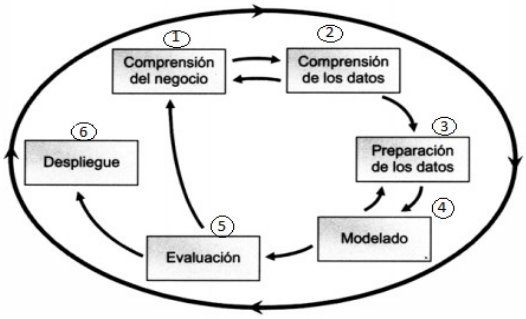


Figure 3. Phases of the CRISP-DM methodology [14].

Business Compression: focuses on understanding the objectives and requirements of the project from a business perspective [14]. **Compression of data:** it is about collecting and becoming familiar with the data, identifying data quality problems and seeing the first potentialities or subsets of data that can be interesting to analyze. **Preparation of the data:** the objective of this phase is to obtain the minable view (data set and descriptions). This includes integration, selection, cleaning and transformation [14]. **Modeling:** is the application of modeling techniques or data mining itself said to the mineable views. **Evaluation:** in this stage it is necessary to evaluate the models of the previous phase, see if the model fits the needs established in the first phase. **Deployment:** during this phase it is about exploring the potential of the models, integrating them into the decision-making processes of the organization, disseminating reports on the extracted knowledge [14].

3. Classification of Texts and Convolutional Neural Networks.

Classification of texts is of great interest at present in the analysis, mainly, of content hosted on social networks, media, search engines, etc. The great interest that the tasks of text classification raise is reflected in different tasks proposed in congress workshops that deal with the topic such as SemEval @ ACL, IberEval @ SEPLN and TASS @ SEPLN [10].
The task of text classification, is to assign a set of classes to a specific document. Formally, given a document d and a fixed set of classes. C = {c 1, c 2, ..., c n} a classifier f must assign the correct class (or classes) to the document, f $(d) = c * d$. Prior to such classification, supervised approaches are often used in which given a training set of m documents labeled, a classifier f is learned [10].

Convolutional neural networks (CNNs) are a type of neural networks [11] [21] [22] [23] [24] used for deep learning and have obtained good results in the classification of texts. These networks are designed to process data that comes in the form of multiple arrays, for example, a color image [13] [17] composed of three 2-dimensional arrays that contain pixel intensities in the three color channels. Many data modalities are in the form of multiple arrays [13]: 1

dimension for signals and sequences, including language; 2 dimensions for images or audio spectrograms; and 3 dimensions for video or volumetric images [13].

This type of network has been used for the processing of natural language, generating extremely good text classifiers and for it to work the only thing that must be done is vectorize our text so that we have a matrix with height and length (such as an image) and treat it as in the existing projects of image classification in CNNs.

The origins of the CNNs were in 1980 with the article Neocognition of Fukushima explained an algorithm very similar to CNN, however, it was not known how to train it. Then the AI went through a stage of darkness due to the lack of scientific results in the field and the great cost of undertaking an AI investigation. It was not until 1993 that Yan LeCun LeNell implemented the first convolutional network applied to images, in a software that simulated the functioning of neurons and learned to read a handwritten text in check and forms. However, the LeCun project was not successful in front of the community of scientific experts in the field of AI at that time. In 2012 George Hinton and 2 students used a network (like the one LeCun did in 1993) in the main contest of ImageNet's Large Image Recognition Challenge, earning first place in the contest.

With respect to their capacity, CNNs are able to learn, automatically through back-propagation, a hierarchy of characteristics that abstract spatial information and can be used for classification problems [15].

Generally, a convolutional network consisting of convolutional layers, pooling and a reshape layer is called a convolutional network, followed by a classification model, for example, Multilayer Perceptron (MLP for its acronym in English) [10].

In the CNNs the concept of layers is maintained, but each neuron of a layer does not receive incoming connections of all the neurons of the previous layer, but only of some. This favors a neuron to specialize in a region of the list of numbers in the previous layer, and drastically reduces the number of weights and multiplications needed. Typically, two consecutive neurons in an intermediate layer specialize in overlapping regions of the anterior layer. [7]

The architecture of a typical convolutional network can be seen in Figure 4 and it is structured as a series of stages; the first stages are composed of two types of layers: convolutional layers and pooling layers.

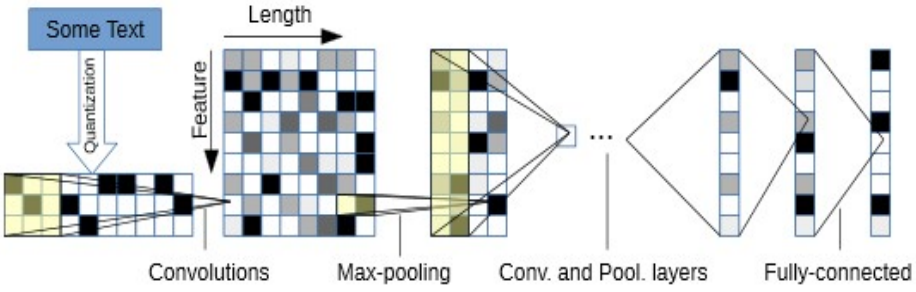


Figure 4. Architecture of CNN text classification. [4]

Units in a convolutional layer are organized into feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called the filter bank. The result of this local weighted sum is passed through a non-linearity function [4] [20], such as a ReLU. All units in a feature map share the same filter bank [13] [17] [18] [19]. Different feature maps in a layer use different filter banks [13]. The reason for this architecture is double [13]:

- In matrix data as images, local groups of values are often highly correlated, forming distinctive local motifs that are easily detected. [13]
- The local statistics of images and other signals are invariant to the location. In other words, if a motif can appear in a part of the image, it could appear anywhere, hence the idea that units in different locations share the same weights and detect the same pattern in different parts of the matrix. [13]

Mathematically, the filtering operation performed by a feature map is discrete convolution, hence the name.

Although the function of the convolutional layer is to detect the local conjunctions of the characteristics of the previous layer, the function of the pooling layer is to combine the semantically similar characteristics in a single layer. Because the relative positions of the characteristics that make up a pattern can vary a bit, reliable detection of the pattern can be done by general granulation of the position of each characteristic. A typical pooling unit (cluster) calculates the maximum of a local patch of units in a feature map (or in a few feature maps). [13]

A pooling layer reduces to: $n * m$ patch in a single value to make the convolutional neuronal network less sensitive to spatial location. The pooling layer is always included after each convolutional layer + activation function.

4. Methodology MCTexto

It is valid to point out that MCTexto eliminates a little documentation generated in relation to other methodologies, however, its stages address the main processes of a text classifier and includes a zero stage for the configuration of the work environment and tools to be used. MCTexto contains 6 stages, visualized in Figure 5 and described later. With MCTexto the application domain is reduced and bringing it to a more specific level, saving time and resource specialists in this type of problem.

Figure 5. Phases of the MCTexto methodology.

1. **Stage Zero:** This stage is one of the most important because the objectives of the project must be clearly identified. In addition to allowing Configure the Development Environment, defining programming language, library and framework to use.
2. **Initial stage:** Collection, integration, selection and manual classification of texts. In this stage, the source of information to be used in the project is identified. The collection, integration and selection of texts is performed. For this stage the texts must be classified in classes, manually or with the use of a system. Conform the report of the exploration of the texts. Main artifacts:
   - ✓ Report of the Exploration of Texts.
   - ✓ Text Description Report.
3. **Text preprocessing:** This stage helps the accuracy of the classification of the text, in addition to prepare the data for the technique to be used, in article [26] define several techniques of image preprocessing, it is valid to emphasize that the philosophy is equal, which in the context of the texts, i.e. Cleaning of the texts, Normalization, Lemmatization, Tokenization and Transformation to Vector Matrix the texts to be classified, using any of the existing representations Bag of Word representation, N-Grams, Word Embedding and even at the level of character. Main artifacts:
   - ✓ Texts Preprocessing Report.
   - ✓ Vector matrix of the corpus.
4. **Definition of CNN Architecture and Training:** The number of layers and neurons in each layer is defined, in addition to the activation function to be selected. The network is trained from the training corpus, defining in the number of runs or epochs that the algorithm will use. Main artifacts:
   - ✓ Text Classification Model.
   - ✓ Model Training Report.

5. **Validate and Interpret Classification model**: The objective is to validate that the Text Classification model fits the defined objectives; if they are favorable, the results should return to Stage 3, as many times as necessary. Main artifacts:

   ✓ Validated model.

   ✓ Validation Report of the Model.

6. **Application of Result:** in this stage the results are applied, always bearing in mind that the model must be fed back from that same application. Main artifacts:

   ✓ Deployment Plan.

   ✓ Final report.

   ✓ Retro-Feeding Model Report.

5. Experiments and Results

It is important to highlight that the implementation of convolutional neural network models requires a high cost of resources, hardware and software. To obtain faster results, better computational resources should be applied, and in the best case graphic cards (GPU) which opened the way to the techniques of Deep Learning, enabling a result of training the network in a very minimal time. However, non-use does not mean that the results obtained are not favorable.

5.1. Stage Zero

Following the MCTexto methodology proposed, it is explained that the research has a corpus of documents stored in a University Institutional Repository. The RI has all the scientific production of the university community (theses, masters, doctorates, articles, etc.). The objective of the case study is to obtain a text classification model, to be implemented in a technological surveillance system.

The Python programming language was used as a language widely used in deep learning applications and natural language processing, in addition to providing useful libraries that were used, such is the case of **NumPy**, which is the fundamental package for computing. scientific with the Python language. NumPy contains among other things a very powerful N-dimensional array object, it offers sophisticated functions [16].

Another leader in natural language processing library is the **NLTK ,** provides easy to use interfaces to more than 50 corpus and lexical resources, together with a set of text processing libraries for classification, tokenization, labeling, analysis and semantic reasoning [6].

Apache MXNET is one of these free code framework that provides resources for an easy implementation of convolutional neural networks. It was used for its being an efficient and flexible library for deep learning. Being one of the libraries that supports the most language, including the Python language [2].

For the case study, an 8-core CPU was used, with 16 GB of RAM with Ubuntu OS 18.04

5.2. Initial Stage

A corpus was obtained from the abstracts of the thesis with the computer profile of the University Institutional Repository database, which are related to the metadata_value table that represents the metadata of the documents.

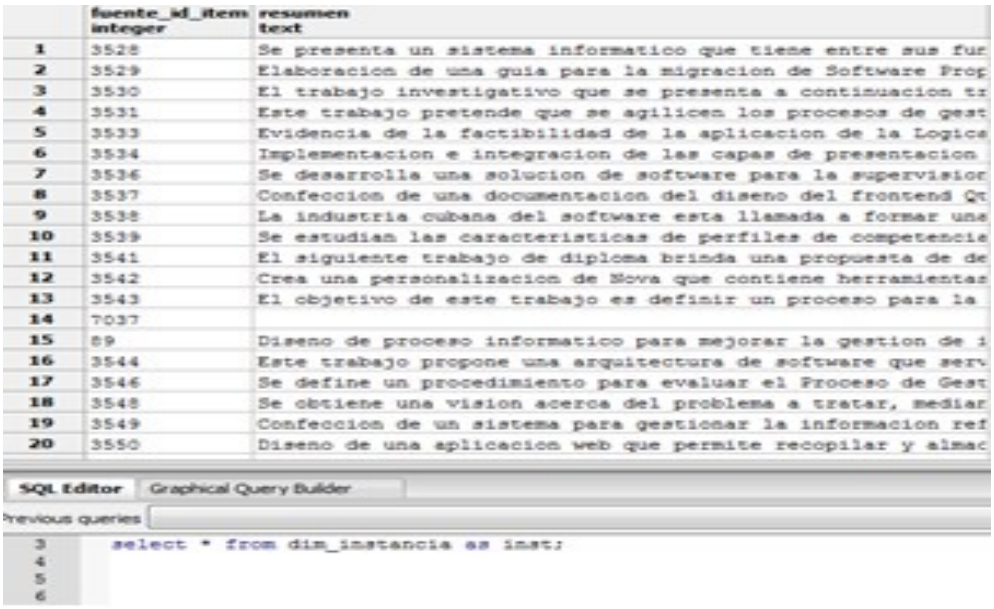
metadata_value_id [PK] integer	item_id integer	metadata_field_id integer	text_value text
1	1	64	Modelacion y visualizacion de superficies de
2	1	3	Valle Martinez, Yusnier autor
3	1	3	Jose Ortiz Rojas tutor
4	1	3	Emilio Escartin Sauleda tutor
5	1	61	INTELIGENCIA ARTIFICIAL
6	1	61	MODELACION
7	1	61	GRAFICOS
8	1	61	TOPOGRAFIA
9	1	61	GEOMETRIA
10	1	61	ANALISIS DE DATOS
11	1	61	ALGORITMOS
12	1	61	TECNICAS
13	1	61	3D
14	1	61	GEOMETRIA DEL ESPACIO
15	1	61	ALMACENAMIENTO DE DATOS
16	1	61	MAESTRIA
17	1	59	006.3-Val-M-TD-2800-09
18	1	27	En el presente documento se exponen nuevas te
19	1	14	2009
20	1	15	2009
21	1	15	7
22	1	11	2011-10-26T15:03:36Z
23	1	12	2011-10-26T15:03:36Z
24	1	25	http://repositorio_institucional.uci.cu/jspui
25	1	28	Made available in DSpace on 2011-10-26T15:03:
26	2	64	Sistema para la integracion del proceso de no
27	2	3	Espinal Martin, Yanet autor

Figure 6. Metadata_value table

A table dim_instancia, (see Figure 7 and 8) was formed from the sql language query, which selects from the table Metadatavalue the text_value field where the metadata_field_id has a value of 64, which corresponds to the summary field, in addition to other fields that were added. For a total of 6768 initial records.

```
Select text_value from metadata_value where metadata_field_id = 64
```

Figure 7. sql query



	fuente_id_item integer	resumen text
1	3528	Se presenta un sistema informatico que tiene entre sus fur
2	3529	Elaboracion de una guia para la migracion de Software Prop
3	3530	El trabajo investigativo que se presenta a continuacion tr
4	3531	Este trabajo pretende que se agilicen los procesos de gest
5	3533	Evidencia de la factibilidad de la aplicacion de la Logica
6	3534	Implementacion e integracion de las capas de presentacion
7	3536	Se desarrolla una solucion de software para la supervisior
8	3537	Confeccion de una documentacion del diseno del frontend Qt
9	3538	La industria cubana del software esta llamada a formar una
10	3539	Se estudian las caracteristicas de perfiles de competencia
11	3541	El siguiente trabajo de diploma brinda una propuesta de de
12	3542	Crea una personalizacion de Nova que contiene herramientas
13	3543	El objetivo de este trabajo es definir un proceso para la
14	7037	
15	89	Diseno de proceso informatico para mejorar la gestion de i
16	3544	Este trabajo propone una arquitectura de software que serv
17	3546	Se define un procedimiento para evaluar el Proceso de Gest
18	3548	Se obtiene una vision acerca del problema a tratar, mediar
19	3549	Confeccion de un sistema para gesticonar la informacion ref
20	3550	Diseno de una aplicacion web que permite recopilar y almac

SQL Editor | Graphical Query Builder

Previous queries

```
3   select * from dim_instancia as inst;
4
5
6
```

Figure 8. Selection of the Summary Field in the Table dim_instancia

5.3. Text preprocessing

This stage has a very important artifact, which consists in the representation of the documents in a vector matrix. The developers are responsible for selecting the representation to be used, always remembering the relationship with the technique to be used.

We proceeded to filter the values according to the interest of the investigation, focusing only on the summary field. Several sql queries were applied, with the aim of exploring the records, concluding that there were 989 records with the null summary field of 6768 initial records.

Due to the nature of the data, it was decided to divide the data into 2 groups using the option to export to the extension .csv, the first of the data for the training where are the abstracts corresponding to the diploma thesis collection for a total of 4550 records representing 78.90% and a second group with the remaining abstracts with 1229 records of abstracts (during the export 12 records presented an error leaving a total of 1217), these correspond to the test data, representing 21.10%.

Each summary text of the corpus was manually classified into 11 classes defined by several experts in Computer Science, where Figure 9 and Figure 10 show the number of texts per class and the distribution in a histogram:

No.	CLASES	DISTRIBUCIÓN
1	Desarrollo Web	988
2	Desarrollo Escritorio	504
3	Desarrollo Cell	256
4	Inteligencia Artificial	824
5	Base de Datos	163
6	Sistema Operativo	62
7	Seguridad Informática	50
8	Telecomunicaciones	106
9	Ingeniería de Software	420
10	Calidad de Software	225
11	Otros	952
Total		4550

Figure 9. Quantity of texts by classes

Figure 10. Histogram of distribution of items by class

The genfromtxt function of NumPy was used to load the data. Once the data was loaded, the texts were cleaned, the file stopwords.txt containing the frequent words of the Spanish language, as shown in Figure 11, was added and this provides the elimination of these words that do not provide knowledge to the investigation.

```
stopword = cargar_stopword();

def cargar_stopword():
 palabras = np.genfromtxt('stopwords.txt',delimiter='\t', names=True, dtype=None);
 stopwords = []
 for word in palabras:
  #cad = str(word,'utf-8');
  cad = str(word,'ISO-8859-1');
  cad = cad.rstrip('\x00')
  stopwords.append(cad);
 return stopwords
```

Figure 11. Method for work with stopword

Unlike others, neural network and deep learning models do not receive the raw text as input data and they only work with a numerical block, so vectorized text is the process to transform the texts.

Text representation at character level was selected, as shown in Figure 12, to obtain a matrix similar to an image as in image classification projects, then each character encoded is equivalent to one pixel in the image this process is well detailed in Zhang's Character-level Convolutional Networks for Text Classification article where the authors conclude in the article that the most important thing in their experiments is that convolutional neuronal networks at the level of characters they could work for the classification of texts without the need for words.

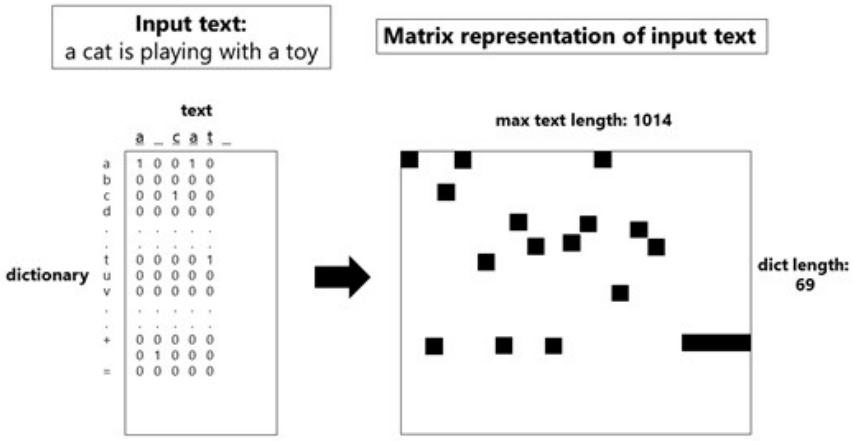


Figure 12. Scheme of the character encoding.
Each sentence is coded as a 69x1014 matrix. [9]

With the encoded function shown in Figure 13, the text corpus was transformed to a vector matrix at the character level and Figure 14 shows the resulting matrix of a summary text. You can check in Figure 15 the representation of a character, in the resulting matrix of the text.

```
alfabeto = list("abcdefghijklmnopqrstuvwxyz0123456789-,;.!?:'\"/\\|_@#$%^&*~`+ =<>()[]{}")

alfabeto_INDEX = {letter: index for index, letter in enumerate(ALPHABET)}

texto_long = 1014


def encode(texto):
    matriz = np.zeros([len(alfabeto), texto_long], dtype='float32')

    i = 0
    for letra in text:
        if i >= texto_long:
            break;
        if letra in alfabeto_INDEX:
            matriz[alfabeto_INDEX[letra]][i] = 1
        i += 1
    return matriz

matriz_encoded = []
for resumen in corpus:
    matriz_encoded.append(encode(resumen))
```

Figure. 13. Implementation of the encode method

```
>>> matriz_encoded[1]
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]], dtype=float32)
```

Figure 14. Vector matrix corresponding to a summary text

```
>>> matriz_encoded[1][0]
array([0., 0., 0., ..., 0., 0., 0.], dtype=float32)


>>> matriz_encoded[1][0][3]
1.0
```

Figure 15. Verifying the representation of the "a" character, active in the 3rd position of the summary text

5.4. Definition of CNN Architecture and Training

For the architecture of the network, it was initially based on the architecture of the Crespe model and its subsequent improvement models. Crespe was the initiator in 2015 of the application of convolutional neural networks in the classification of text at the character level. The Crespe model composed of 9 layers (6 convolutional and 3 fully connected) [9]. It is valid to note that the layer number is adopted according to experience in another model, there is no manual that says the number of layers according to the type of problem.

In the project the architecture of the model was designed as follows: 5 layers (3 convolutional layers and 2 totally connected layers), illustrated in Figure 16. The activation function used is the ReLU very common in function deep learning this has the characteristic of its result is between 0 and 1, causing the gradient to propagate completely backwards if $x > 0$. In the implementation of the network layers, it was carried out with the support of the MXNet framework evidenced in Figures 17 and 18.

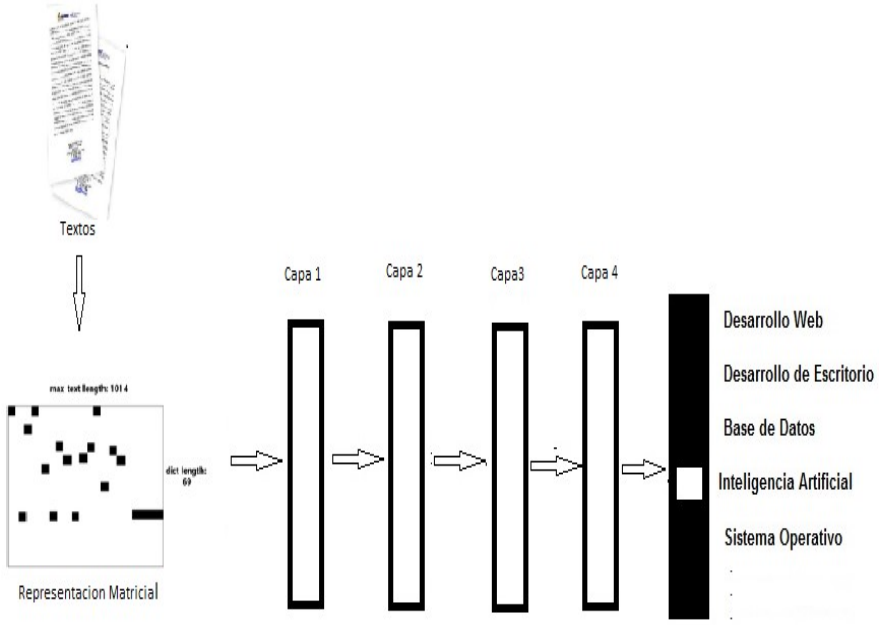


Figure 16. Representation of the Text Classifier Architectures

```
#primera capa convolucional

conv1 = mx.sym.Convolution(data = data, kernel = (7,69), num_filter = 256)
relu1 = mx.sym.Activation(data = conv1, act_type = "relu")
pool1 = mx.sym.Pooling(data = relu1,
                       pool_type='max',
                       kernel=(3,1),
                       stride=(1,1))


#segunda capa convolucional

conv2 = mx.sym.Convolution(data = pool1, kernel = (7,69), num_filter = 256)
relu2 = mx.sym.Activation(data = conv2, act_type = "relu")
pool2 = mx.sym.Pooling(data = relu2,
                       pool_type = "max",
                       kernel = (3,1),
                       stride=(1,1))


#tercera capa convolucional

conv3 = mx.sym.Convolution(data = pool2, kernel = (3,69), num_filter = 256)
relu3 = mx.sym.Activation(data = conv3, act_type = "relu")
pool3 = mx.sym.Pooling(data = relu3,
                       pool_type = "max",
                       kernel = (3,1),
                       stride=(1,1))
```

Figure 17. Implementation of the Convolutional Layer, Activation Function and Layer Pooling using the MXNET framework

```
#primera capa conectada completamente

flatten = mx.sym.flatten(data = pool3)
fc1 = mx.symbol.FullyConnected(data = flatten, num_hidden = 1024)
relu4 = mx.sym.Activation(data = fc1, act_type = "relu")


#segunda capa conectada completamente
fc2 = mx.sym.FullyConnected(data = relu4, num_hidden = 11)
```

Figure 18. Implementation of the 2 Layers Fully Connected

The training time lasted about 2 days with a run of 10 times, using for the optimization of the model the gradient descending method (See Figure 19) in charge of updating the set of parameters of the network in an iterative way to minimize the function of error, and learning radius 0.1

```
model = mx.mod.Module(symbol = lenet, context = mx.cpu())

model.fit(train_ite,
          eval_data = val_iter,
          optimizer = 'sgd',
          optimizer_params = {'learning_rate':0.1},
          eval_metric = 'acc',
          batch_end_callback = mx.callback.Speedometer(batch_size,100),
          num_epoch = 10)
```

Figure 19. Code for CNN training

Figure in chart 20 model accuracy where the x axis represents the number of times used for model training and the y - axis reflects the accuracy.

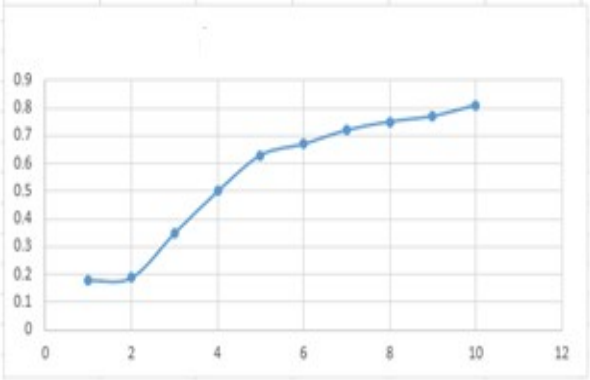


Figure. 20: Precision graph of the model, during each of the periods in training

5.5. Validate and Interpret Classification model

In this stage, the cross-validation algorithm was selected, using the test dataset and obtaining 75% accuracy in the tests, in Figure 21 the validation code used for the model can be seen. Considering the result satisfactorily, although the same can be improved with more training of the model, being able to also use a feedback of the same

```
test_iter = mx.oi.NDArrayIter(corpus_test, None, batch_size)
prob =  model.predict(test_iter)
test_iter = mx.oi.NDArrayIter(corpus_test, label_test, batch_size)
```

Figure 21. Code the validation of the model

5.6 Result Application

The model demonstrates the application of the proposed methodology, where it can be applied to other classification projects. The model was applied to the surveillance system of the Institute of Cybernetics, Mathematics and Physics of Havana, based on an API that links the results of the model to the surveillance system. In the same way, reports could be generated, functionality still pending.

6. CONCLUSIONS

The MCTexto methodology proposal for the implementation of text classifiers with convolutional neural networks was designed and put into practice. A novel "deep learning" technique, specifically convolutional neuronal networks, was applied in a problem of text classification, with representation of the texts at the character level. New concepts of stonemasons on convolutional neuronal networks were addressed.

As a future project, we intend to design and apply a new stage to the MCTexto methodology: knowledge transfer, thus enabling rapid training of the network. A documentary classifier is implemented from the application of the methodology, the case study and some algorithms presented as validation to the MCTexto methodology. The tool may be applicable to document management environments with a computer profile up to now.

ACKNOWLEDGEMENTS

Recognition to the Institute of Cybernetics, Mathematics and Physics, Havana. In addition to the support provided by the AIT Management in Cienfuegos.

REFERENCES

[1] V. Guevara, «Scientific Databases,» 2015. [En línea]. Available: http://www.scientificdatabases.ca/current-projects/english-spanish-text-data-mining/mineria-de-texto-ingles-espanol/mineria-de-texto/.

[2] The Apache Software Foundation (ASF), «MXNet,» 2018. [En línea]. Available: https://mxnet.apache.org/.

[3] M. Allahyri, S. Safaei, S. Pouriych, M. Assefi, E. Trippe D., J. B. Gutierrez y K. Krys, «A brief Survey of Text Mining: Classification, Clustering and Extraction Techniques,» https://arxiv.org/pdf/1707.02919.pdf, 2017.

[4] X. Zhang y Y. LeCun, «Text Understanding from Scratch,» arXiv:1502.01710v5 [cs.LG], New York, 2016.

[5] G. Miner, J. Elder, A. Fast, T. Hill, R. Nisbet y D. Delen, Practical Text Mining and Statistical Analysis forn Non-structured Text Data Applications, vol. 1st Edition, Academic Press, 2012.

[6] «https://www.nltk.org,» 2018. [En línea]. Available: https://www.nltk.org.

[7] J. M. T. J. D. P. V. Bryan García Navarro, «Implementación de Técnicas de Deep Learning,»

Universidad de La Laguna, Escuela Superior de Ingeniería y Tecnología, Espana, 2015.

[8] T. Coburn, «Aprendizaje profundo,» *Prensa Científica - Muntaner 339, pral 1a. - 08021 Nº 479,* Agosto 2016.

[9] Fierro, «Could-Scale Text Classification with Convolutional Neural Networks on Microsoft Azure,» 2016.

[10] J. Á. Gonzalez Barba, «Aprendizaje Profundo para el Procesamiento del Lenguaje Natural,» Universidad Polítecnica de Valencia, Valencia, 2017.

[11] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning, MIT Press, 2016.

[12] D. T. Larose, Discovering Knowledge in Data, Wiley Interscience.

[13] Y. LeCun, Y. Bengio y G. Hinton, «Deep learning,» NATURE (VOL 521), pp. 2-2, Mayo 2015.

[14] J. H. Orallo, J. Ramírez Quintana y C. Ferri Ramírez, Introduccion a la Mineria de Datos, Pearson Education, 2004.

[15] D. Stutz, «Understanding convolution neural networks,» *InSeminar Report,Fakult at f ur Mathematik, Informatik und Naturwissenschaften Lehr-und Forschungsgebiet Informatik VIII Computer Vision,* 2014.

[16] SciPy, «SciPy,» 2018. [En línea]. Available: https://docs.scipy.org/doc/numpy.

[17] Y. Yu, H. Lin, J. Meng, X. Wei, H. Guo y Z. Zhao, «Deep Transfer Learning for Modality Classification of Medical Images,» *Information,* vol. 8, 2017.

[18] D. Kane, «SlideShare,» 23 Febrero 2015. [En línea]. Available: https://www.slideshare.net/DerekKane/data-science-part-xi-text-analytics. [Último acceso: Junio 2019].

[19] M. D. P. GOMEZ-GIL y B.S., «RECOGNITION OF HANDWRITTEN LETTERS USING A LOCALLY CONNECTED BACK-PROPAGATION NEURAL NETWORK,» Faculty of Texas Tech University, Texas, 1991.

[20] S. Acharya, A. K. Pant y P. K. Gyawali, «Recognition, Deep Learning Based Large Scale Handwritten Devanagari Charecter,» *IEEE,* 2015.

[21] H.-J. Yoo, «Deep Convolution Neural Networks in Computer Vision: a Review,» *IEIE Transactions on Smart Processing and Computing,,* vol. 4, nº 1, Febrero 2015.

[22] J. J. Arguello, «Heterogeneous Multi-Agent Deep Reinforcement Learning for Traffic Lights

Control,» University of Dublin, Trinity College, 2018.

[23] HUGHES SYSTIQUE Corporation, «HUGHES SYSTIQUE,» [En línea]. Available: https://hsc.com/Services/Product-Engineering-Services/Application-Engineering/Convolution-Neural-Networks. [Último acceso: Junio 2019].

[24] S. F. Seixas, «MEDIUM,» 25 Junio 2017. [En línea]. Available: https://medium.com/@seixaslipe/building-a-simpsons-classifier-with-deep-learning-in-keras-36a47fe17f79. [Último acceso: Junio 2019].

[25] A. Plasencia, «Aprendizaje profundo para la clasificación de textos,» INFO 2018, Habana, 2018.

[26] J. Pradeep, S. E. y H. S., «Diagonal Based Feature Extraction for Handwritten Alphabets Recognition System Using Neural Network,» International Journal of Computer Science and Technology (IJCST),, vol. 1, nº 2, 2018.

Authors

Evelyn Guindo Betancourt engineer in computer science. He worked at the University of Computer Science in productive projects playing the role of analyst and developer. Currently working as a developer, especially with technologies and languages for the Web, php, css, bootstrap, veu, angular, etc. He is a student of the Applied Cybernetics Master's Degree in Data Mining, allowing him to learn technologies and tools for the analysis of data and text.In the future, you want to be a data analyst

Armando de Jesús Plasencia Salgueiro Doctor of Science, Researcher at the Institute of Cybernetics, Mathematics and Physics (Icimaf). He works on research topics related to Data Mining, Text Mining, Information Retrieval. Master's and doctoral thesis tutor. It has several publications of scientific articles.