Fine-tuning mT5-based Transformer via CMA-ES for Sentiment Analysis

Orlando Grabiel Toledano-López[1,*], Julio Madera[2,*], Hector González[1], Alfredo Simón-Cuevas[3], Thomas Demeester[4] and Erik Mannens[4]

[1]Universidad de las Ciencias Informáticas (UCI), La Habana, Cuba
[2]Universidad de Camagüey, Camagüey, Cuba
[3]Universidad Tecnológica de La Habana "José Antonio Echeverría", Cuba
[4]Ghent University, Belgium

Abstract

In this paper, we describe the methods used to submit our results to the Rest-Mex Sentiment Analysis of the Iberian Languages Evaluation Forum 2022. The addressed challenge proposes a sentiment analysis task of Spanish opinions, categorizing each message into five emotions, and an attraction prediction subtask divided into three categories. Accordingly, our contribution is a hybrid method based on the Estimation of Distribution Algorithms for fine-tuning an mT5-based transformer. For this, we propose the design and development of a deep learning model using the encoder part of the pre-trained mT5-based transformer. The proposed model is trained by dividing the process into two stages using AdamW and the Covariance Matrix Adaptation Evolution Strategy. With this approach, 0.3050 of Mean Absolute Error was obtained for the polarity detection subtask and 0.9781 of Macro F-measure for the attraction prediction subtask, reaching the 10th place out of the 24 teams in competition.

Keywords

mT5-based Transformer, Covariance Matrix Adaptation Evolution Strategy, Sentiment Analysis

1. Introduction

The tourism sector has benefited from recent research in the area of Natural Language Processing (NLP). Some platforms on the web, such as TripAdvisor, allow tourists to write opinion about places and tourist attractions they visit. The collected opinions become a source of data that helps to identify problems based on analysis of semantic aspects of the content of the opinions given by visitors [1]. A priori, the opinion review can provide the visitor an idea about the services offered in the place, their comfort, and whether they have exceeded the customers' expectations. In this way, tourism service destinations use this information to improve customer services since it is known that the opinions given about a place influence the future selection of the destination to visit [2, 1].

IberLEF 2022, September 2022, A Coruña, Spain
ogtoledano@uci.cu (O. G. Toledano-López); julio.madera@reduc.edu.cu (J. Madera)
0000-0001-8263-0425 (O. G. Toledano-López); 0000-0001-5551-690X (J. Madera); 0000-0002-7601-4201 (H. González); 0000-0002-6776-9434 (A. Simón-Cuevas); 0000-0002-9901-5768 (T. Demeester); 0000-0001-7946-4884 (E. Mannens)

Sentiment analysis is a specific task within NLP [3]. This task seeks to categorize a text or comment into positive or negative for binary classification [4], or it can be extended to multi-class problems using a numerical scale ranging from 1 to N, being 1 very negative and N very positive [5]. In this way, the sentiment analysis task can be understood in practice as a text classification problem [6].

Recent research has improved deep learning architectures based on seq2seq models by introducing self-attention mechanisms with transformers, allowing to contextualize each word of a text with other words [7]. A relevant aspect of these architectures is their versatility to adapt to different NLP challenges like text classification, with which additional neural networks on top can be added to the architecture, and fine-tuning the model for learning downstream task-specific knowledge [8, 9, 10, 11]. However, basic approaches for training these deep learning architectures are methods based on partial derivates of the objective function, such as Stochastic Gradient Descent (SGD) [12] and Adam [13]. These methods have theoretical-practical limitations that are evident in their probability of convergence to local minimums, which results in a significant effect on the model generalization [14].

Some approaches followed in the literature seek to improve the results using regularization of the loss function, such as L2 regularization [15] and weight decay regularization [16]. Different results reported in [17, 18, 19] show the effectiveness of applying SGD for image processing tasks with such regularization. However, for Adam, it has not had good results to apply this regularization, for it has been proposed in [16] a modification that decouples the optimal choice of weight decay (AdamW) factor from the setting of the learning rate. However, the regularization focuses on improving the model generalization, reducing its complexity by penalizing high values of its weights.

Continuous and discrete optimization has been addressed via meta-heuristic algorithms in NLP [20, 21, 22]. In [20], Genetic Programming (GP) is used to obtain an optimal multi-task topology in feed-forward neural networks for text classification and the Seq2Seq model for conversational systems. In [21], Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Firefly Algorithm (FA) are introduced for feature selection from documents represented by Latent Semantic Analysis (LSA), and general machine learning approaches are subsequently used. Swarm intelligence and Genetic Algorithms (GA) have allowed optimizing the initial weight matrix in Long Short-Term Memory (LSTM) neural network for text classification [22]. However, these evolutionary algorithms employ general search operators that do not consider relationships between the variables of the optimization problem. In this respect, Estimation of Distribution Algorithms (EDA) modify the search operator through a probabilistic model that can learn these dependency relationships, thus contributing to the exploitation of local information [23]. In contrast to GA, they do not require the classical crossover and mutation operators [24].

Fine-tuning transformers for text categorization is a problem of high dimensionality which requires a large number of parameters to be optimized [25]. As consequence, this leads to a high computational cost of the training process, so some population meta-heuristics are limited to obtaining good results with a reasonable population size. Based on this, our contribution is a new hybrid method that combines AdamW [16] with Covariance Matrix Adaptation-Evolution Strategy (CMA-ES) [26] for fine-tuning an mT5-based transformer [27] in polarity prediction and attraction prediction tasks. CMA-ES is a type of EDA that takes advantage of the

eigenvalue decomposition property of the covariance matrix as a search operator for continuous optimization on non-convex problems.

In the next, we present the design of an mT5-based transformer model for text classification and introduce a hybrid method for fine-tuning this model that combines AdamW and CMA-ES. We use the dataset released for the Rest-Mex 2022 [28], where the polarity and attraction predictions are included in the same dataset. We discuss the comparison between the different approaches used in the solution considering the metrics: Accuracy, Macro F-measure, and Mean Absolute Error. Finally, some conclusions and recommendations are given.

2. Computational methodology

Giving an input text sequence x, the classification model can be considered as a function, i.e. $f(x) = C$ for measuring the conditional probability distributions over all possible labels in the pre-defined category set $C = \{c_1, c_2, c_3, ..., c_L\}$. Text classification works in a instance of space $\mathcal{X}$ where each instance is an input text sequence x. Hence, let $X = \{x_1, x_2, x_3, ..., x_N\}$ be the training set, an each input text sequence $x_i = \{t_1, t_2, t_3, ..., t_P\}$, represents a sequence of tokens. $Y = \{y_1, y_2, y_3, ..., y_N\}$ the set of categories in which each document x_i is classified, each $y_i \in C$.

The model will be trained for the set of parameters $\theta \in \mathbb{R}^q$ minimizing the cross-entropy loss in the Equation 1a:

$$\theta = _{\theta \in \mathbb{R}^q} F(\theta, X) \tag{1a}$$

$$F(\theta, X) = -\sum_{i=1}^{N}\sum_{c=1}^{L} \Upsilon_{x_i|c} \log(P_{x_i|c}(\theta)) \tag{1b}$$

$$\Upsilon_{x_i|c} = \begin{cases} 1 \ if \ y_i = c \\ 0 \ otherwise \end{cases} \tag{1c}$$

Where $P_{x_i|c}(\theta)$ is the predicted conditional probability of x_i giving class c.

2.1. Model description

In this section, we propose an mT5-based transformer model for text classification, shown in Figure 1 that receives an input token sequence. For mT5 model, an input sequence can be represented in two ways by using special tokens, such as a single sequence x_1 </s> or a pair of sequences x_1 </s> x_2 </s>. The special token </s> indicates the end of the sequence. As mT5 is a massive multilingual T5 (Text-to-Text Transfer Transformer) [27] model, it can be used for multiple tasks in NLP, which leads to the input sequence being specified as a task prefix. For the problem addressed, we will use the prefix *"multilabel classification:"*. An example of input is formatted as: $\{multilabel, classification, :\}$ x </s>. mT5 model is an encoder-decoder architecture, in our proposal, we will use the encoder part of the architecture, whose output is the input to a fully-connected head on top.

The encoder part converts the input sequence to a sequence of hidden states as text representation. This part consists of a stack of blocks formed by a self-attention layer followed by a

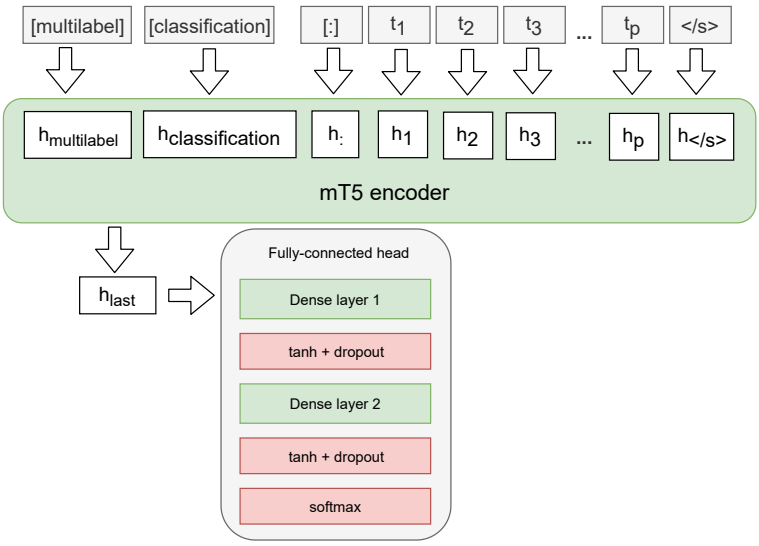

Figure 1: MT5 encoder model overview with additional fully-connected head

small feed-forward network [27, 29]. Only using the encoder part, the complexity of the model is reduced to less than half of the parameters compared to using both encoder and decoder parts.

In our design, we add a head formed by a dense layer of 64 neurons and the output layer whose neurons correspond to the number of classification labels. The first dense layer receives a vector of size 512 that represents the last hidden state h_{last}, both layers have as activation function the hyperbolic tangent (*tanh*), and dropout is applied to both at the output. Finally, the softmax function is applied to the output of the last layer.

The output on the fully-connected head is a simple softmax classifier on top of the mT5 encoder. Let θ be the set of trainable parameters of the model, the fully-connected head turns the vector h_{last} from input sequence x_i into the conditional probability distributions $P(c_l|h_{last}, \theta)$ over all categorical labels C as follow:

$$P(c_l|h_{last}, \theta) = softmax(\tanh(\tanh(h_{last}\theta_1^T)\theta_2^T)) = \frac{exp(P(c_l|h_{last}, \theta))}{\sum_{c=1}^{L} exp(P(c_l|h_{last}, \theta))} \quad (2)$$

Where $\theta_1 \in \mathbb{R}^{K\times|h_{last}|}$ and $\theta_2 \in \mathbb{R}^{L\times K}$ are the trainable parameters of the first dense layer and second dense layer. K is amount of neurons of the first dense layer, both θ_1 and θ_2 are subset of θ such that $\theta_1 \cup \theta_2 = \theta \setminus \{\theta_{mt5}\}$ being θ_{mt5} trainable parameters of mT5 encoder.

Hence, the final prediction of the model $\hat{y}$ giving input x is taken with the maximum label value of the conditional probability distributions:

$$\hat{y}_x = P(c_l|h_{last}, \theta) \quad (3)$$

Algorithm .1 describes how to use a hybrid EDA-based method for fine-tuning our custom mT5-based transformer model. The algorithm receives as input the set of training patterns X

and the set of categories Y of each input. We proposed a combination of the training process in two stages. First, we train the mT5 encoder body via the AdamW method, and after we train the fully-connected head via EDA optimization. In this, an individual represents the set of parameters in θ_2, and the fitness function is evaluated for θ that represents all trainable parameters in the mT5-based transformer model. The fitness function is computed for all trainable parameters in the model as we show in Equation 1a.

An individual $\theta_2 = (u_1, u_2, u_3, ..., u_\gamma)$ is a configuration that corresponds to a distribution $p(\theta_2) = [P_0 = u_1, P_1 = u_2, P_3 = u_3, ..., P_\gamma = u_\gamma]$. Optimization problem consists to find a minimum configuration θ_2 where $F(.) : \mathbb{R}^q \rightarrow \mathbb{R}$.

Algorithm .1: General approach for fine-tuning mT5-based transformer via EDA

Require: Set of training pattern $X = (x_1, x_2, x_3, ..., x_N)$.
Require: Fitness function $F(\theta, X)$.
* *STAGE 1: Learning parameters of the mT5 encoder by AdamW method.* *\
while Criterion not fulfilled **do**
  for $batch\ in\ X$ **do**
    Forward pass in $batch$ to get output $\hat{y}$.
    Backforward propagation to compute gradients.
    Update parameters θ with computed gradients.
  end for
end while
* *STAGE 2: Learning fully-connected head by EDA optimization.* *\
Initialize $g \leftarrow 1$.
Keep parameters for mT5 encoder θ_{mt5} and θ_1.
Generate initial population $Z = \{\theta_{2,1}, \theta_{2,2}, \theta_{2,3}, ..., \theta_{2,\Lambda}\}$.
while Criterion not fulfilled **do**
  for $\theta_{2,\lambda} \in I$ **do**
    Evaluate the fitness function $F(\theta_{mt5} \cup \theta_1 \cup \theta_{2,\lambda}, X)$
  end for
  Select an intermediate set CS from M individuals.
  Estimate distribution from CS through $p^{CS} = p(\theta_2, g)$.
  Generate new individuals from $p(\theta_2, g+1) \approx p^{CS}(\theta_2, g)$.
  $g = g + 1$.
end while
return Best individual θ_2^* in the population.

In fully-connected head optimization, the most important step is the estimation of the distribution p^{CS} for generating new individuals. In our proposal, covariance matrix $Cov \in \mathbb{R}^\gamma$ is initialized in identity matrix $Cov = I$, this matrix is symmetric and positive definite. We fix the initial centroid $\Omega \in \mathbb{R}^\gamma$ and the initial step-size $\sigma \in \mathbb{R}$ that corresponds to initial variance. Initial Ω is the estimation of the location of the optimum for each parameter in an individual and indicates where to start the evolution. For estimating distribution we sort the individuals by increasing fitness function $F_1 < F_2 < F_3 < ...F_\Lambda$ and select M best individuals in CS. For each generation, we obtain new individuals following Equation 4:

$$\theta_{2,\lambda}^{(g+1)} = \Omega^{(g)} + \sigma^{(g)}\mathcal{N}(0, Cov^{(g)}) \tag{4}$$

The matrix Cov has an orthonormal basis of eigenvectors defined as B as results of eigenvalue decomposition of $Cov = BD^2B^T$, being B an orthogonal matrix, where $B^TB = BB^T = I$. $D^2 = diag(d_1^2, ..., d_\gamma^2)$ is a diagonal matrix with eigenvalues of Cov as diagonal elements. Hence, $D = diag(d_1, ..., d_\gamma)$ is a diagonal matrix with square roots of eigenvalues of D as diagonal elements [26].

Using previous eigenvalue decomposition of Cov, $\mathcal{N}(0, Cov^{(g)})$ can be computed as following Equation 5:

$$\mathcal{N}(0, Cov^{(g)}) = BD\mathcal{N}(0, I) \tag{5}$$

Where $\mathcal{N}(0, I)$ represents a standard normally distributed vector which are realizations from a multivariate normal distribution with zero mean and identity covariance matrix. Selection and recombination is an important step in the evolution strategy to adjust of initial centroid Ω, taking into account initial weights for each point of distribution. Moreover, σ is considered an step-size and we perform its control using property of matrix decomposition Cov and initial evolution path $p \in IR^\gamma$. Finally, we perform the covariance matrix adaptation step as described in [26].

2.2. Dataset

The dataset used was provided by the contest organizers. This collection was obtained from the tourists who shared their opinion on TripAdvisor, between 2002 and 2021, in the most representative places of Mexico. Each opinion's class is an integer between [1, 5], where 1 represents the most negative polarity, and 5 is the most positive. Respect to the previous contest in Rest-Mex 2021 [30], also includes another challenge, which is the type prediction (Attraction). This is another subtask where the examples are classified into three classes: Attractive, Hotel, and Restaurant.

The dataset collects raw texts in the Spanish language and was divided by the event organizers themselves into 70% for training and 30% for testing. Other fields are included to accompany the opinions, so its initial format is structured in the following fields:

- **Title:** The title that the tourist himself gave to his opinion. *Data type:* Text
- **Opinion:** The opinion issued by the tourist. *Data type:* Text
- **Polarity:** The label that represents the polarity of the opinion. *Data type:* [1, 2, 3, 4, 5]
- **Attraction:** The label of the type of place of which the opinion is being issued. *Data type:* [Hotel, Restaurant, Attractive]

The complete dataset contains a total of 30 212 examples. Figure 2 illustrates the distribution of examples per label for both the polarity prediction and attraction prediction subtasks for the training set. It can be noted that for both subtasks the dataset is unbalanced.

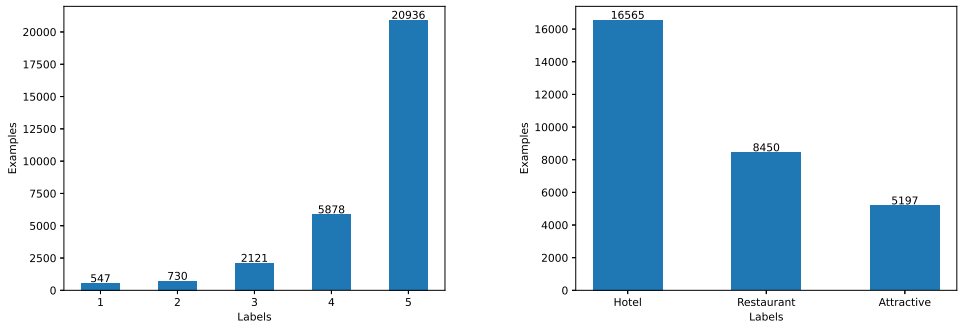

Figure 2: Distribution of examples per label for the polarity and attraction prediction

3. Results and discussion

This section describes the experimental setup and results of the proposed method for mt5-based transformer optimization. As a gradient-based method, we use AdamW as an optimizer. We compare the results with AdamW regarding the combination of our algorithm as a hybrid proposal: AdamW+CMA-ES. We specify network-related hyper-parameters taking into account the training algorithm. However, we do not deal with model selection.

Our model is implemented with Pytorch, it is available online at the link https://github.com/ogtoledano/Rest_mex_DL_EDA. For performing the EDA algorithm we use the Deap library as an evolutionary computation framework that includes implementations of meta-heuristic algorithms. We run the experiments in Google COLAB using a GPU environment.

To perform the experimental analysis, we take 10% of the dataset as the validation set. In the pre-processing phase, we concatenate the **Title** field of the opinion and the **Opinion** into a single text using a colon. The resulting text was prefixed with the task prefix as indicated in section 2.1. After we removed Spanish stop-words, tokenize the text, and the text size is set to 200 tokens using zero as the padding value. The tokenization process was performed using the pre-trained tokenizer T5Tokenizer. The pre-trained tokenizer is part of the mT5 model and is available at: https://huggingface.co/google/mt5-small.

For training the proposed model with AdamW, we train with 3 epochs as maximum and 8 opinions as batch size. Learning rate (α), weight decay(wd) and dropout (dp) were considered as hyperparameters for both subtask, taking the following values: $\alpha = 6e-5$, $wd = 0.01$, and $dp = 0.1$. After, we apply the CMA-ES for learning parameters on the last layer in the fully-connected head making the second stage of the proposed method and comparing the results. To perform the second stage by EDA optimization, we set the initial step-size $\sigma = 0.95$ and the mean of the initial distribution vector $\Omega = 0.05$ for each parameter. For polarity prediction, we use 20 individuals and 25 generations, and for attraction prediction, we use 25 individuals and 15 generations.

Table 1 shows the results of the model with different learning approaches. This table shows the columns of Accuracy (Acc), Mean Absolute Error (MAE), Macro F-measure (F), and Recall. Acc, F, and MAE were used as evaluation measures since it is the measure that the organizers

take as official. It can be appreciated how a determined previous result with AdamW produces with the proposed method better results with the final trained mT5 model. The best result for each metric is marked in bold typeface.

Polarity prediction				
Algorithm	Acc	MAE	F	Recall
AdamW	75,6743	**0,2787**	**0,7487**	0,7567
AdamW+CMA-ES	**75,9722**	0,2820	0,7403	**0,7597**
Attraction prediction				
Algorithm	Acc	MAE	F	Recall
AdamW	97,6336	0,0253	0,9764	0,9763
AdamW+CMA-ES	**97,8984**	**0,0222**	**0,9784**	**0,9790**

Table 1
Validation set results in polarity and attraction prediction

Figure 3 shows the confusion matrix of the results for each subtask using AdamW as an optimizer in the validation set. The color map represents the cell values on a percentage scale concerning the number of examples per label, the same percentage scale is applied in the Figure 4. It can be seen that despite the unbalance in the dataset for each subtask, the model manages to correctly classify instances for all labels. It is worth noting that in the attraction prediction subtask only 133 instances were incorrectly classified.

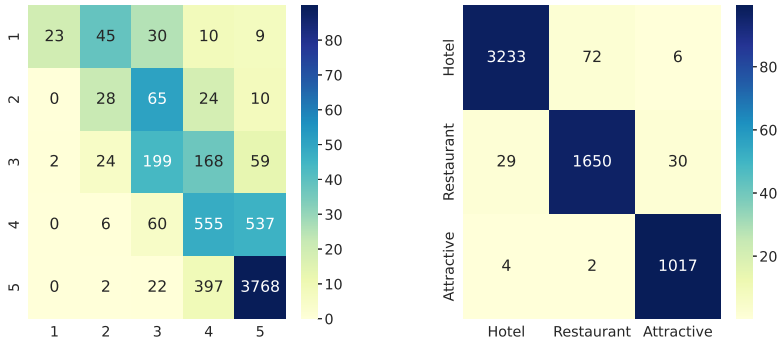

Figure 3: Confusion matrix for AdamW in polarity and attraction prediction

Figure 4 shows the confusion matrix of the results for each subtask using our approach AdamW+CMA-ES in the validation set. In the polarity prediction subtask, better results can be observed in labels 1, 3, and 5, which contributed to better values in the accuracy and recall metrics. For the attraction prediction subtask, only 127 examples were classified incorrectly contributing to better results in all reported metrics.

Table 2 shows the final results in the test set sent by the event organizers. In the general ranking of teams, we reached 10th and 11th places out of a total of 24 teams. For the final result of the challenge, we used the metric proposed by the organizers in Equation 6, which is the

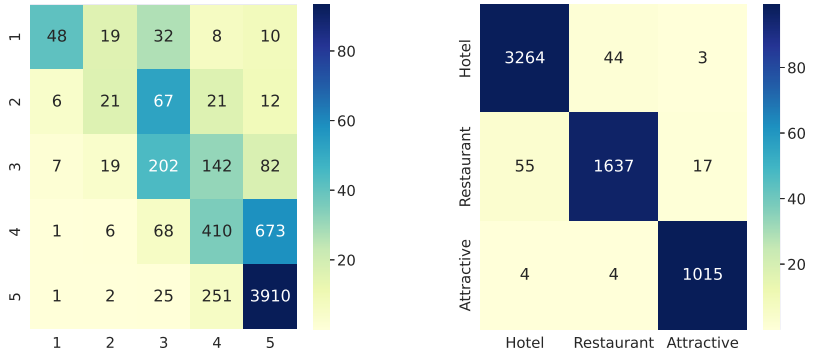

Figure 4: Confusion matrix for AdamW+CMA-ES in polarity and attraction prediction

Polarity prediction				
Algorithm	Acc	MAE	F	Recall
AdamW	73,6858	0,3053	0,4468	**0,5438**
AdamW+CMA-ES	**74,5284**	**0,3050**	**0,4500**	0,5115
Attraction prediction				
Algorithm	Accuracy	MAE	F	Recall
AdamW	97,4412	-	0,9723	0,9681
AdamW+CMA-ES	**97,9051**	-	**0,9781**	**0,9775**

Table 2
Test set results in polarity and attraction prediction

average of the inverse of MAE of polarity prediction and the F of attraction prediction. For AdamW+CMA-ES it is **0,8722** and for AdamW it is **0,8692**.

$$Sentiment_{task} = \frac{\frac{1}{1+MAE_{polarity}} + F_{attraction}}{2} \tag{6}$$

Although population meta-heuristics have a higher computational cost compared to gradient-based methods for training the neural network, they have a higher probability of giving a solution close to the global optimum and can escape local extremes [31, 32]. We apply this advantage in a specific part of our proposed architecture based on the mT5 encoder. The last dense layer optimization of the fully-connected head, using the proposed meta-heuristics, enables the interpretation of the extracted features from the mT5 encoder in terms of a predictive output. Experimental results indicate the influence of learning parameters distribution in this part of the model through the CMA-ES, to find parameters close to the global optimum and obtain a model with greater generalization capacity. Considering our mT5 encoder design in Figure 1, the amount of network parameters for a five-class classification problem (polarity prediction) is 146 973 765, with only 325 parameters for the last dense layer of the fully-connected head. This takes advantage of what the network learned in the first stage, and with the last hidden state

produced by the mT5 encoder, refines the weights of the last layer in the fully-connected head which is a simpler network part.

4. Conclusions and Further Work

In this paper, we present a hybrid gradient-based method and CMA-ES together for fine-tuning the mT5-based transformer in text classification. The use of AdamW as an optimizer in the first phase takes advantage of the speed of these methods for learning parameters of the mT5 encoder, something that would not be appropriate to apply through population meta-heuristics due to the high parameterization of this type of deep learning model. The application of EDA in the training of a part of the mT5-based transformer model, with fewer parameters, improves the results and maintains good performance on the unbalanced dataset by learning the distribution of each parameter in the last dense layer of the head. Further study should be conducted to make an analysis of dependencies between neural network parameters and their impact in comparison with other methods that do not assume these dependencies.

Acknowledgments

This work has been partially funded by VLIR-UOS Network University Cooperation Programme-Cuba. We gratefully acknowledge the computing time granted through UCI-HPC and Computational Mathematics Study Center at the University of Informatics Sciences supercomputer resources.

References

[1] S. Forouzandeh, K. Berahmand, E. Nasiri, Rostami, Mehrdad, A Hotel Recommender System for Tourists Using the Artificial Bee Colony Algorithm and Fuzzy TOPSIS Model: A Case Study of TripAdvisor, International Journal of Information Technology & Decision Making 20 (2021) 399–429. doi:10.1142/S0219622020500522.

[2] C. Zuheros, E. Martínez-Cámara, E. Herrera-Viedma, F. Herrera, Sentiment Analysis based Multi-Person Multi-criteria Decision Making methodology using natural language processing and deep learning for smarter decision aid. Case study of restaurant choice using TripAdvisor reviews, Information Fusion 68 (2021) 22–36. doi:10.1016/j.inffus.2020.10.019.

[3] R. Guerrero-Rodriguez, M. Á. Álvarez-Carmona, R. Aranda, A. P. López-Monroy, Studying online travel reviews related to tourist attractions using nlp methods: the case of guanajuato, mexico, Current Issues in Tourism (2021) 1–16. doi:https://doi.org/10.1080/13683500.2021.2007227.

[4] X. Fang, J. Zhan, Sentiment analysis using product review data, Journal of Big Data 2 (2015). URL: http://dx.doi.org/10.1186/s40537-015-0015-2. doi:10.1186/s40537-015-0015-2.

[5] J. Tao, X. Fang, Toward multi-label sentiment analysis: a transfer learning based approach, Journal of Big Data 7 (2020) 1–26. URL: https://doi.org/10.1186/s40537-019-0278-0. doi:10.1186/s40537-019-0278-0.

[6] M. Á. Álvarez-Carmona, R. Aranda, R. Guerrero-Rodríguez, A. Y. Rodríguez-González, A. P. López-Monroy, A combination of sentiment analysis systems for the study of online travel reviews: Many heads are better than one, Computación y Sistemas 26 (2022). doi:https://doi.org/10.13053/CyS-26-2-4055.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems 2017-Decem (2017) 5999–6009. arXiv:1706.03762.

[8] S. Yu, J. Su, D. Luo, Improving BERT-Based Text Classification with Auxiliary Sentence and Domain Knowledge, IEEE Access 7 (2019) 176600–176612. doi:10.1109/ACCESS.2019.2953990.

[9] S. Zheng, M. Yang, A New Method of Improving BERT for Text Classification, in: Z. Cui, J. Pan, S. Zhang, L. Xiao, J. Yang (Eds.), Intelligence Science and Big Data Engineering. Big Data and Machine Learning, Springer International Publishing, Cham, 2019, pp. 442–452.

[10] P. Gao, S. Geng, Y. Qiao, X. Wang, J. Dai, H. Li, Scalable Transformers for Neural Machine Translation (2021). URL: http://arxiv.org/abs/2106.02242. arXiv:2106.02242.

[11] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, F. Wu, BertGCN: Transductive Text Classification by Combining GCN and BERT, in: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, 2021, pp. 1456–1462. doi:10.18653/v1/2021.findings-acl.126. arXiv:2105.05727.

[12] J. Kiefer, J. Wolfowitz, Stochastic Estimation of the Maximum of a Regression Function, Annals of Mathematical Statistics 23 (1952) 462–466.

[13] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 2015, pp. 1–15.

[14] J. Rojas-Delgado, R. Trujillo-Rasúa, R. Bello, A continuation approach for training Artificial Neural Networks with meta-heuristics, Pattern Recognition Letters 125 (2019) 373–380.

[15] C. Cortes, G. Research, N. York, L 2 Regularization for Learning Kernels, in: Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, 2004, pp. 109–116.

[16] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR 2019, 2019, pp. 1–8. arXiv:1711.05101.

[17] P. Foret, A. Kleiner, H. Mobahi, B. Neyshabur, Sharpness-Aware Minimization for Efficiently Improving Generalization, in: International Conference on Learning Representations, 2020. URL: http://arxiv.org/abs/2010.01412. arXiv:2010.01412.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, in: International Conference on Learning Representations, 2020. URL: http://arxiv.org/abs/2010.11929. arXiv:2010.11929.

[19] D. Soydaner, A Comparison of Optimization Algorithms for Deep Learning, International Journal of Pattern Recognition and Artificial Intelligence 34 (2020).

[20] I. Chaturvedi, C. L. Su, R. E. Welsch, Fuzzy Aggregated Topology Evolution for Cognitive Multi-tasks, Cognitive Computation 13 (2021) 96–107. doi:10.1007/s12559-020-09807-4.

[21] R. Janani, S. Vijayarani, Automatic text classification using machine learning and optimization algorithms, Soft Computing 25 (2021) 1129–1145. doi:10.1007/

s00500-020-05209-8.

[22] H. K. Maragheh, F. S. Gharehchopogh, K. Majidzadeh, A. B. Sangar, A New Hybrid Based on Long Short-Term Memory Network with Spotted Hyena Optimization Algorithm for Multi-Label Text Classification, Mathematics 10 (2022). doi:10.3390/math10030488.

[23] P. Larrañaga, A Review on Estimation of Distribution Algorithms, in: Estimation of Distribution Algorithms, 1996, 2002, pp. 57–100. doi:10.1007/978-1-4615-1539-5_3.

[24] J. Madera, B. Dorronsoro, Estimation of Distribution Algorithms, in: E. Alba, R. Martí (Eds.), Metaheuristic procedures for training neural networks, 1 ed., Springer US, 2006, pp. 87–108. doi:10.1007/0-387-33416-5.

[25] M. Á. Álvarez-Carmona, R. Aranda, A. Y. Rodríguez-González, L. Pellegrin, H. Carlos, Classifying the mexican epidemiological semaphore colour from the covid-19 text spanish news, Journal of Information Science (2022). doi:https://doi.org/10.1177/01655515221100952.

[26] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies., Evolutionary computation 9 (2001) 159–195. doi:10.1162/106365601750190398.

[27] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, C. Raffel, mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies", 2021, pp. 483–498. doi:10.18653/v1/2021.naacl-main.41.

[28] M. Á. Álvarez-Carmona, Á. Díaz-Pacheco, R. Aranda, A. Y. Rodríguez-González, D. Fajardo-Delgado, R. Guerrero-Rodríguez, L. Bustio-Martínez, Overview of Rest-Mex at IberLEF 2022: Recommendation System, Sentiment Analysis and Covid Semaphore Prediction for Mexican Tourist Texts, Procesamiento del Lenguaje Natural 69 (2022).

[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67. arXiv:1910.10683.

[30] M. Á. Álvarez-Carmona, R. Aranda, S. Arce-Cárdenas, D. Fajardo-Delgado, R. Guerrero-Rodríguez, A. P. López-Monroy, J. Martínez-Miranda, H. Pérez-Espinosa, A. Rodríguez-González, Overview of Rest-Mex at IberLEF 2021: Recommendation System for Text Mexican Tourism, Procesamiento del Lenguaje Natural 67 (2021). doi:https://doi.org/10.26342/2021-67-14.

[31] V. K. Ojha, A. Abraham, V. Snásel, Metaheuristic Design of Feedforward Neural Networks : A Review of Two Decades of Research, Engineering Applications of Artificial Intelligence 60 (2017) 97–116.

[32] R. García-Ródenas, L. J. Linares, J. A. López-Gómez, Memetic algorithms for training feedforward neural networks: an approach based on gravitational search algorithm, Neural Computing and Applications 33 (2021) 2561–2588. doi:10.1007/s00521-020-05131-y.