Maria Mihaela TRUȘCĂ, PhD Candidate
E-mail: maria.trusca@csie.ase.ro
Associate Professor Anamaria ALDEA, PhD
E-mail: anamaria.aldea@csie.ase.ro
Simona Elena GRĂDINARU, PhD Candidate
E-mail: simona.gradinaru@csie.ase.ro
Professor Crișan ALBU, PhD
E-mail: crisan.albu @csie.ase.ro
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies

POST-PROCESSING AND DIMENSIONALITY REDUCTION FOR EXTREME LEARNING MACHINE IN TEXT CLASSIFICATION

Abstract. *Text classification is one of the core technologies of textual analysis, with interesting applications varying from sentiment classification, language identification to online abuse detection and many more. Many approaches have been taken to the machine learning classifiers employed in text categorization, with a focus on boosting model performance and efficiency. This paper proposes a new effective framework for the input processing of the Extreme Learning Machine (EML) algorithm, illustrated on the Reuters-21578 test collection of documents. We employ Glove word embeddings to provide a compact and semantic meaningful representation for each word of the input documents. Given the bi-dimensionality constraint of ELM inputs, we reduce the dimensionality of word embeddings using Latent Semantic Analysis and Principal Components Analysis (PCA). Our results reveal that PCA together with the post-processing operation led to more accurate results with lower computational costs.*

Keywords*: Extreme Learning Machine, Glove Word Embeddings, Post-processing Algorithm, Principal Component Analysis, Latent Semantic Analysis.*

JEL Classification: C38, C45

1. Introduction

Machine Learning and Artificial Intelligence are transforming nearly every industry in the digital society nowadays, and text analysis is a central area of interest. Unstructured text data has experienced a massive increase in the digital era, but it's impractical for humans to analyze it at this pace. Many organizations need to parse and classify documents to make their text data easier to manage and

DOI: 10.24818/18423264/55.4.21.03

exploit. Manual classification is often time-consuming and error-prone, thus automating such processes using machine learning and natural language processing is becoming more popular by virtue of their long-term benefits.

Text classification aims to group text units, typically sentences or documents, into classes. Regardless of the difficulty level, it is one of the core technologies of textual analysis with interesting applications such as sentiment classification, language identification, online abuse detection, or trend detection based on the customers' feedback. Hence considering the steep increase of online content available on the World Wide Web, it is easy to apprehend the constant interest to explore and extend the text classification task. In scientific literature, text classification techniques vary from classic machine learning solutions based on support vector machine (SVM) (Wang, et al., 2006), maximum entropy (Wang, et al., 2010) or random forest (Islam, et al., 2019), to neural networks (Luan & Lin, 2019).

The large variety of advanced classification techniques makes the process of selecting the most performant model highly dependent on the nature of the analyzed problem and the available data. The final performance is constrained by the model architecture and input processing, as well. In the current paper, we focus on the latter and seek to provide a robust approach for the input refinement required to classify documents using EML. This method has been introduced as a much faster learning alternative to the traditional backpropagation algorithm for the single-layer feedforward neural network (SLFNN) (Huang, et al., 2004). We employed ELM as it has proved to be more reliable than the traditional machine learning classifiers (Zheng, et al., 2013), (Roul, et al., 2015), and it provides a good alternative to the backpropagation-based neural networks, as it avoids convergence issues and requires less time and computational resources for training. Since our main interest lies in adjusting the input for ELM, we employ only the basic version of this algorithm.

With regards to the input framework, our work is based on the post-processing algorithm introduced by (Mu & Viswanath, 2018), with the purpose to increase the discriminative nature of word representations and thus capture more information. In addition, we also assess the effect of reducing the input dimensionality by means of the Principal Component Analysis (PCA) (Wold, et al., 1987) and Latent Semantic Allocation (LSA) (Schutze, et al., 2008), for comparison purposes. The combination of ELM for text classification and dimensionality reduction of word embeddings brings an interesting contribution to literature. According to our framework and the case study on Reuters-21578 data collection, dimensionality reduction not only lessens the computational complexity and reduces the training time, but it can also enhance the model accuracy.

The remaining part of the paper is organized into the following parts. Section 2 offers an overview of the related literature. Section 3 presents the EML algorithm and the post-processing operations. Section 4 describes the data, and Sect. 5 is dedicated to the empirical results. Section 5 summarizes our conclusions.

2. Related works

Like our work, the framework introduced by (Zheng, et al., 2013) relies on LSA to reduce the dimensionality of the input required to train the ELM algorithm. However, instead of using word embeddings to generate new sentence representations, the LSA method is applied to scaled TF-IDF scores. Later, (Li, et al., 2018) generate sentence representations using the average of the word2vec word embeddings (Mikolov, et al., 2013). The method leverages the weighted ELM to solve the problem of imbalanced classes by assigning a relevance score to each document based on the inter-class and intra-class information entropy.

Plain word2vec word embeddings were also embodied in the solution presented by (Waheeb, et al., 2020). Since the proposed method was designed to detect sentiment labels of the discharge summaries, the input is enriched with features like rules for sentiment-shifters or medical concepts. Besides the already mentioned LSA, (Zhang, et al., 2020) include PCA in their analysis with the purpose to reduce the dimensionality of the hidden layer. The reason behind using PCA is to avoid the problem of multicollinearity observed at the level of the hidden layer.

Contrasting with the previous approaches, (Roul, et al., 2015) focused on techniques for selection of the input features, considering options like chi-squared, information gain and bi-normal separation. In addition, (Roul, et al., 2015) also evaluated the multi-layer implantation of ELM based on auto-encoder structures for the task of text classification.

3. Proposed method

Given a Single Layer Feedforward Neural Network (SLFNN), the solution is often provided by the backpropagation algorithm that aims to iteratively adjust the weights to reduce the difference between the iteration-level output and the real one or to minimize the loss function (Rumelhart, et al., 1986). Regarding the weights, gradient-descent algorithms like Adagrad (Duchi, et al., 2011), Adam (Kingma & Ba, 2015) or Adadelta (Zeiler, 2012) are usually utilized. Considering the gradient of the cross-entropy loss function with respect to the weights W, $\frac{\partial\, loss}{\partial\, W}$ computed based on the chain rule and the learning rate η, the batch gradient descent updates the weights for the entire training data as follows:

$$W_{new} = W_{old} - \eta \frac{\delta\, Loss}{\delta\, W} \quad (1)$$

However, despite the popularity of the backpropagation algorithm, some downsides have been elaborated over the last few years. The most important ones relate to the time-consuming nature of the backpropagation algorithm, and to the large number of hyperparameters that require laborious optimizations. In addition,

backpropagation can lead to a local minimum, especially in such cases when the learning rate is too large. To tackle all these disadvantages, (Huang, et al., 2004) propose EML as an alternative learning method for the traditional backpropagation, as further described.

Considering the weights $w = [w_i, \dots, w_{\widetilde{N}}]$ and $b = [b_i, \dots, b_{\widetilde{N}}]$ that connect the input *X* with *N* features with the $\widetilde{N}$ hidden neurons, the output of the hidden layer computed using the activation function *g(·)* is:

$$H(w_1, \dots w_{\widetilde{N}}, b_1, \dots b_{\widetilde{N}}, x_1, \dots x_N) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_{\widetilde{N}} \cdot x_1 + b_{\widetilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_{\widetilde{N}} \cdot x_N + b_{\widetilde{N}}) \end{bmatrix} \quad (2)$$

According to ELM, the weights *w* and *b* are chosen arbitrary, while the weight vector β that connects the hidden layer with the output *T* with *m* features is estimated as follows:

$$\hat{\beta} = H' T \quad (3)$$

where *H'* is the Moore-Penrose generalized inverse of matrix *T* (Serre, 2001).

Since the initial ELM algorithm trains the SLFNN on the entire data, leading to many potential computational errors, (Huang, et al., 2005) introduced the extended version of ELM – OS-ELM that allows feeding data sequentially in an online approach based on the recursive least-squares (Chong & Zak, 2004). This process requires a two steps approach. First, SLFNN is trained on a larger batch whose size should be at least equal to $\widetilde{N}$, and then the remaining training data is split into smaller batches than the initial batch. In the current implementation, only the first batch is different from 1.

Before applying ELM in the text classification field, we need first to select the most appropriate input features. As the input required for the training of ELM is bi-dimensional, we need a one-dimensional representation for each sentence. The two available options we consider refer to the aggregation of word embeddings using sum or average. The new sentence representation will have the same dimensionality as the initial word embeddings.

The word embeddings that we use in the current work are computed based on the Glove model introduced by (Pennington, et al., 2014). Unlike most of the models developed to generate word representations, the Glove model does not aim to predict the probability of observing a word or a set of words in a sentence. Precisely, the model does not use only local information but also relies on global statistics. To compute new word vectors, the Glove model considers that the logarithmic value of the co-occurrence C_{ij} of two words x_i and x_j in the corpus should be equal to the dot product of their representations, w_i and w_j. Also considering the biases b_i and b_j, this relation is defined as:

$$w_i^T w_j + b_i + b_j = log(C_{ij}) \quad (4)$$

To increase the discriminative nature of the word representations and to boost the EML performance, we post-process the word embeddings (Mu & Viswanath, 2018) following a three steps approach:

- Step 1: given the word representation v_i and the mean of all word representations $\underline{v}$, update v_i as follows:

$$v_i = v_i - \underline{v} \tag{5}$$

- Step 2: compute the PCA components $[p_i, \dots, p_d]$ of the word embeddings, where d gives the size of word embeddings.
- Step 3: update the word vector v_i as follows:

$$v_i = v_i - \sum_{i=1}^{D} (p_i^T v_i) p_i \tag{6}$$

In the current framework, we set $D = 7$, as suggested by (Raunak, et al., 2019). Inspired by the work of (Raunak, et al., 2019), we add the fourth step and compute the PCA components of the post-processed word embeddings. However, in our work, the purpose of this step is not only to generate new word representations, but also to reduce their dimensionality.

To achieve this aim, we simply keep the first *P* components given by the eigenvectors of the covariance matrix, that store 95% of the cumulative variance (eigenvalues of the covariance matrix). The main reason behind this additional step consists in increasing the efficiency and assessing effectiveness of working with lower-dimensional word vectors.

In addition to PCA, we also evaluate the LSA method for dimensionality reduction (Schutze, et al., 2008). Given the input *X* of order $V \times d$ storing all word embeddings and a threshold k smaller than the rank of *X*, the core of the LSA method is to determine the low-rank approximation, X_k, of the input *X*, of a rank at most *k*. The new approximation X_k minimizes the Frobenius norm of the difference $F = X - X_k$ and has a lower dimensionality than the input *X*. The Frobenius norm is computed as:

$$F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{d} F_{ij}^2} \tag{7}$$

The computations of the low-rank approximation X_k follow a four-steps procedure:

- Step 1: compute the singular-value-decomposition of the input *X*:

$$X = U \Sigma V^T \tag{8}$$

where the columns of *U* and *V* are the eigenvectors of XX^T and $X^T X$, respectively. Σ is a diagonal matrix with the non-zero elements σ_i are referred to as singular values of *X* and are equal to the square root of the eigenvalues λ_i of the XX^T or $X^T X$ matrices ($\Sigma_{ii} = \sigma_i = \sqrt{\lambda_i}$, with $\lambda_i \geq \lambda_{i+1}$).

- Step 2: compute Σ_k by replacing all singular values σ_i with zeros, where $i > k$.
- Step 3: compute the low-rank approximation X_k:

$$X_k = U \Sigma_k V^T \tag{9}$$

- Step 4: remove all the zero features of the matrix $X_k$to reduce the dimensionality.

Again, we keep only the first components that store 95% of the cumulative variance, according to the eigenvalues λ_i of the XX^Tor X^TX matrices.

4. Data description

The aforementioned methods are employed on the Reuters-21578 collection, which comprises newswire stories published by Reuters in 1987. The documents are classified into more than 90 categories, mostly with regards to business and economy. It is probably the widest used dataset in text categorization research, and it has some particularities that make it interesting to experiment with. First, each document may belong to one or more categories or no category at all, as it is often the case of real data. Moreover, it contains highly skewed categories: some categories have thousands of documents classified under them, while others have very few. Lastly, some categories have hidden semantic relations between them and there is no hierarchy defined based on the categories.

Roughly half of the dataset is used in experiments, as many documents have no assigned topic or spelling errors, a subset referred to as ModApte split is commonly used. Although most of the research papers evaluate their proposed methods on the ModApte dataset, the corpus is usually dubbed Reuters-21578. Therefore, we adopted the same practice in the current paper as well.

The dataset is divided into training (7,769 documents) and testing (3,019 documents) subsets and it is included in the NLTK python library. It contains 90 possible classes, and each document can have one or multiple labels. The class distributions of the training and testing subsets are very similar. Figure 1 displays the most frequent classes per subset.

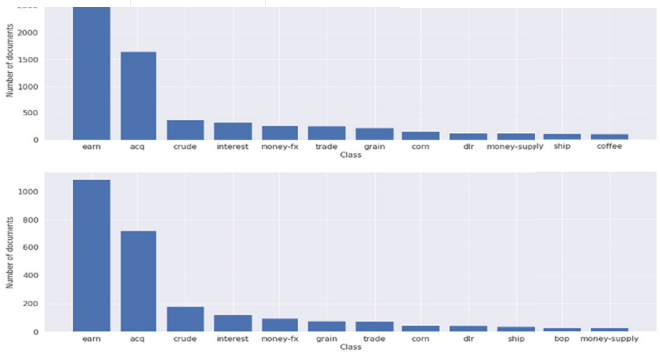


Figure 1. The most frequent classes observed in the training dataset (first plot) and testing dataset (second plot)

Source: our computations - Python

5. Results

The first step of our analysis consists in finding the optimal set of hyperparameters, initial batch size and number of neurons of the hidden layer, for the two aggregation options (sum and average) used to compute one-dimensional sentence representations based on Glove word embeddings. According to our setup, the initial batch size varies between 500 and 1,000 with step 100, and the number of neurons varies between 1 and 450 with step 50. Considering the optimization results (presented as heat maps in the Annexes 1 and 2), the accuracy is reduced as the number of neurons converges to the lower or upper limits of the considered range, given a small initial batch size. The best results are obtained when the first batch size is 1,000 and the number of neurons is 200 for the sum of word embeddings, or 300 for the average of word embeddings. Despite the potential better performances for higher initial batch sizes, we limit the size to 1,000 to avoid running out of computing resources. Likewise, we notice that the average of word embeddings is more suitable for sentence representations than the sum option, generating better results across all hyperparameter sets. As a result, future experiments rely only on this aggregation option.

The best results are obtained when the first batch size is 1,000 and the number of neurons is 200 for the sum of word embeddings, or 300 for the average of word embeddings. Despite the potential better performances for higher initial batch sizes, we limit the size to 1,000 to avoid running out of computing resources. Likewise, we notice that the average of word embeddings is more suitable for sentence representations than the sum option, generating better results across all hyperparameter sets. As a result, future experiments rely only on this aggregation option.

To boost the performance of the OS-ELM model, we apply the post-processing algorithm proposed by (Mu & Viswanath, 2018) to increase the discriminative nature of the Glove word embeddings. Additionally, we consider the LSA and PCA methods for dimensionality reduction.

Baselines:

- OS-ELM+Mean: apply the average Glove word embeddings for sentence representations.
- OS-ELM+MeanPCA: apply the average of the first P components computed by the PCA method. Considering that the principal components should explain at least 95% of the variance of the Glove word embeddings, the optimal P is 216 (Figure 2).
- OS-ELM+MeanLSA: the method is similar to the OS-EML+MeanPCA method, except that the components are computed by the LSA method. The optimal P value is 217 (Figure 2).

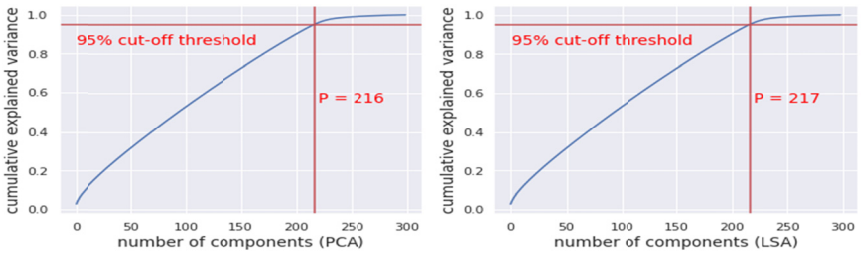


Figure 2. Cumulative explained variance of the Glove word embeddings reported for PCA (first plot) and LSA (second plot)[1]

Source: our computations - Python

Proposed methods:

- OS-ELM+MeanPPA: apply the average of the post-processed Glove word embeddings
- OS-ELM+Mean(PPA+PCA): apply the average of the first *P* PCA components computed using the post-processed Glove word embeddings. After the post-processing, the number of components that explain 95% of the variance is still 216.
- OS-ELM+Mean(PPA+LSA): the method is similar to the above method, except for the dimensionality reduction method (*P* = 217).

Table 1. OS-ELM results

	Model	Accuracy
Baselines	OS-ELM_Mean	**0.7387**
	OS-ELM+MeanPCA	0.7193
	OS-ELM+MeanLSA	0.7235
Proposed methods	OS-ELM+MeanPPA	0.7479
	OS-ELM+Mean(PPA+PCA)	***0.7482***
	OS-ELM+Mean(PPA+LSA)	0.7237

Source: our computations - Python

Based on the results reported in Table 1, the post-processing of the word embeddings turns out to be a necessary step that improves all three baselines. Regarding the dimensionality reduction, we notice that model accuracy is usually traded off for efficiency. However, the PCA components of the post-processed word representations have slightly better results than the post-processed word embeddings, proving that a lower-dimensional sentence representation can be effective for a task like text classification with sequential ELM.

[1] The number of PCA / LSA components that explain 95% of the post-processed Glove word embeddings are unchanged, and the plots of cumulative explained variance are similar to the plots of Figure 2.

While the prior analysis was performed considering components that capture 95% of input variance, we are also interested in understanding the impact of the number of PCA or LSA components on the model accuracy. To this objective, we extend the analysis to include lower-dimensional word embeddings computed using a few components that varies between 150 and 250 with a step of 10, as shown in Figure 3. First, we notice that post-processing typically enhances the classification quality, regardless of the number of components. Second, the PCA method seems to be more suitable than LSA for the dimensionality reduction task, especially when the number of discarded components is small. Third, we notice that model performance squares off when at least 200 components are being kept. As a result, even if we increase the 95% threshold for the cumulative explained variance, we do not expect to observe a significant improvement in terms of accuracy.

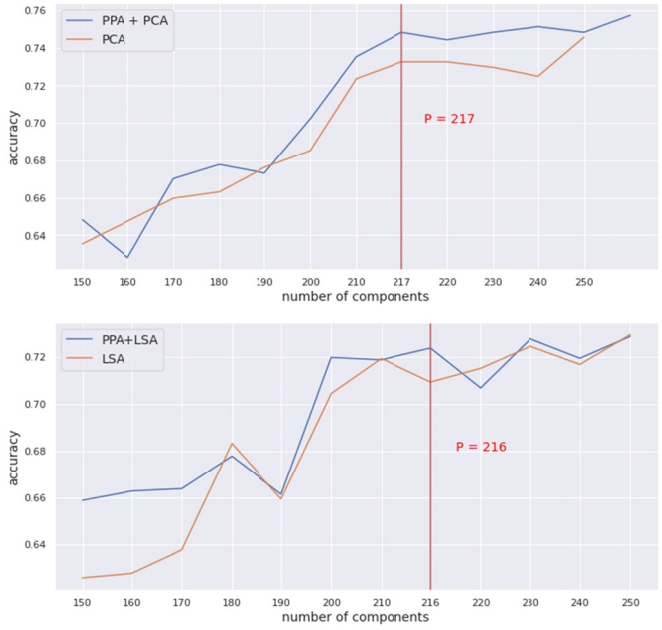


Figure 3. OS-ELM accuracy reported for different numbers of components using (PPA+)PCA (up) and (PPA+)LSA (down)

Source: our computations - Python

To have a better understanding of the quality of the Glove word embeddings, we randomly pick six words ("russia", "china", "germany", "penguin", "building", "economy") and assess their relevance with respect to the word "france" by means of the Euclidean distance and cosine similarity. Given the

word embeddings w_i and w_j of the words x_i and x_j and their size d, the metrics are computed as follows:

$$Euclidean\ distance: d(w_i, w_j) = \sqrt{(w_{i1} - w_{j1})^2 + \cdots + (w_{id} - w_{jd})^2} \quad (9)$$

$$Cosine\ siminarity: s(w_i, w_j) = \frac{w_i \cdot w_j}{||w_i|| \times ||w_j||} = \frac{\sum_{k=1}^{d} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{d} w_{ik}^2} \times \sqrt{\sum_{k=1}^{d} w_{jk}^2}} \quad (10)$$

According to the Figure 4, the baseline word appears to be most related to countries, the most with Germany and the least with China, as one may have expected. Naturally, the remaining three words are significantly less related to "france", especially the word "penguin".

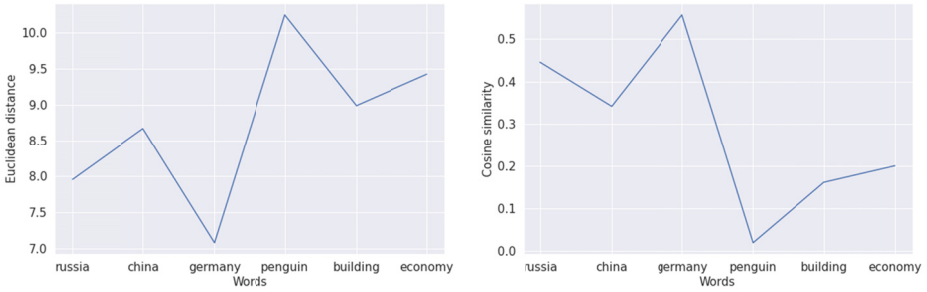


Figure 4. Euclidean distance (left) and cosine similarity (right) with respect to the target word "france"

Source: our computations - Python

Holding the same assumptions as above, we are interested to see if the PPA algorithm enhanced with the PCA dimensionality reduction method that generates 216-dimensional vectors can capture the same information as the initial 300-dimensional Glove word embeddings. While the rankings of the relevance scores are unchanged, we perceive a tendency to overestimate, highlighted through smaller distances and higher similarities (Figure 5). However, the more relevant words are more overestimated than the less relevant words. As a result, the dimensionality reduction differentiates better between relevant and non-relevant words, boosting the performance of the ELM model.

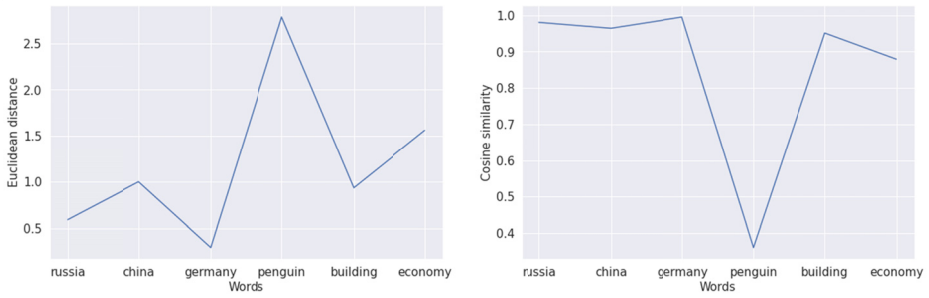


Figure 5. Euclidean distance (left) and cosine similarity (right) with respect to the target word "france". The metrics are computed using the PPA+PCA Glove word embeddings

Source: our computations - Python

6. Conclusions

Text classification is one of the core technologies of textual analysis, with interesting multidisciplinary applications that are helpful to many organizations nowadays. In scientific literature, text classification techniques vary from classic machine learning solutions based on support vector machine, maximum entropy to random forest and neural networks, with a focus on boosting model performance and efficiency. The final model performance is highly dependent on the model architecture and the techniques used to process inputs.

The purpose of our paper is to introduce a new efficient but also effective framework for the input processing of the EML algorithm. First, we provide a compact and semantic meaningful representation for each word of the input documents using Glove word embeddings that have proved higher effectiveness rates than the widely used word2vec word representations. Since the input required to train EML is bi-dimensional, we need to aggregate word embeddings and generate one-dimensional sentence representations. Next, Mu and Viswanath's post-processing algorithm is applied to generate more distinct word embeddings increasing the model performance. As we are also interested in boosting model efficiency, we reduce the dimensionality of word embeddings using LSA and PCA methods.

Our approach brings a few improvements to the post-processing algorithms proposed by scientific literature, and it is also unique by applying PCA for the dimensionality reduction of word embeddings employed in ELM. The case study on Reuters-21758 document collection reveals that while the LSA method leads to poorer results, PCA together with the post-processing operation leads to more accurate results with lower computational costs.

Regarding future work, we are interested to evaluate our framework on more recent context-dependent word embeddings that are by default more discriminative than Glove representation. As standard ELM has already proved

satisfactory good results, we consider that a stacked structure where the word embeddings are also refined by an ELM auto encoder-based model (Lauren, et al., 2017) might boost the performance even more.

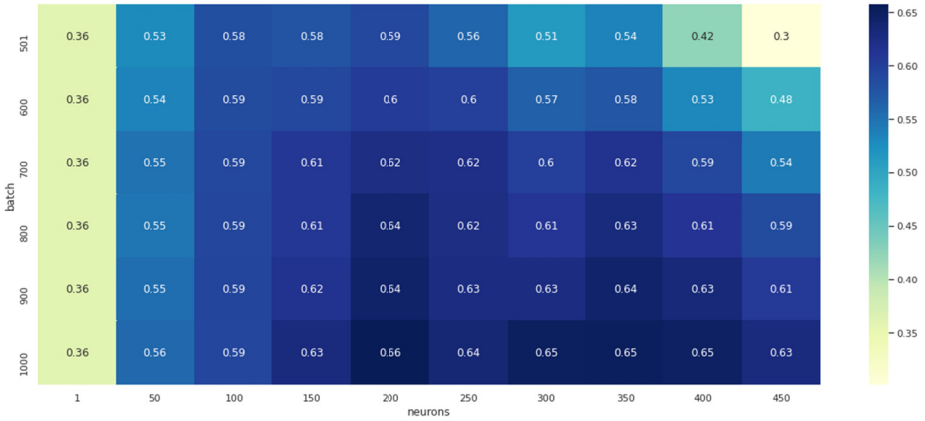
REFERENCES

[1] **Chong, E. K., Zak, S. H. (2004),** ***An Introduction to Optimization.*** *John Wiley & Sons*;
[2] **Duchi, J., Hazan, E., Singer, Y. (2011),** ***Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.*** *Journal of machine learning research,* 12(7), 2121-2159;
[3] **Huang, G.-B. et al. (2005),** ***On-line Sequential Extreme Learning Machine****. Computational Intelligence*, 232-237;
[4] **Huang, G.-B., Zhu, Q.-Y., Siew, C.-K. (2004),** ***Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks****. IEEE International joint conference on neural networks, 2*;
[5] **Islam, M. Z., Liu, J., Li, J., Liu, L., Kang, W. (2019),** ***A Semantics Aware Random Forest for Text Classification****. Proceedings of the 28th ACM International Conference on Information and Knowledge Management,* 1061-1070;
[6] **Kingma, D. P., Ba, J. (2015),** ***Adam: A Method for Stochastic Optimization****. arXiv preprint arXiv:1412.6980*;
[7] **Lauren, P. et al. (2017),** ***A low-Dimensional Vector Representation for Words Using an Extreme Learning Machine****. International joint conference on neural networks (IJCNN), IEEE, 1817-1822*;
[8] **Li, M., Xiao, P., Zhang, J. (2018),** ***Text classification based on ensemble extreme learning machine****. arXiv preprint arXiv:1805.06525.*
[9] **Luan, Y., Lin, S. (2019),** ***Research on Text Classification based on CNN and LSTM****. IEEE*, 352-355;
[10] **Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013),** ***Efficient Estimation of Word Representations in Vector Space****. arXiv preprint arXiv:1301.3781*;
[11] **Mu, J., Bhat, S., Viswanath, P. (2017),** ***All-but-the-Top: Simple and Effective Postprocessing for Word Representations****. arXiv preprint arXiv:1702.01417*;
[12] **Pennington, J., Socher, R., Manning, C. D. (2014),** ***Glove: Global Vectors for Word Representation****. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532-1543;
[13] **Raunak, V., Gupta, V., Metze, F. (2019),** ***Effective Dimensionality Reduction for word Embeddings****. Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, 235-243;

[14] **Roul, R. K., Nanda, A., Patel, V., Sahay, S. K. (2015),** ***Extreme Learning Machines in the Field of Text Classification***. *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 1-7;
[15] **Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986),** ***Learning Representations by Back-propagating Errors***. *Nature,* 323.6088, 533-536;
[16] **Schutze, H., Manning, C. D., Raghavan, P. (2008),** ***Introduction to Information Retrieval***. *Cambridge University Press Cambridge, 39, 234-265*;
[17] **Serre, D. (2001),** ***Matrices: Theory & Applications Additional Exercises***. *L'Ecole Normale Supérieure de Lyon;*
[18] **Waheeb, S. A., Ahmed Khan, N., Chen, B., Shang, X. (2020),** ***Machine Learning Based Sentiment Text Classification for Evaluating Treatment Quality of Discharge Summary***. *Information,* 11(5), 281;
[19] **Wang, H., Wang, L., Yi, L. (2010),** ***Maximum Entropy Framework Used in Text Classification***. *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2, 828-833;
[20] **Wang, Z.-Q., Sun, X., Zhang, D.-X., Li, X. (2006),** ***An Optimal SVM-based Text Classification Algorithm***. *International Conference on Machine Learning and Cybernetics, IEEE*, 1378-1381;
[21] **Wold, S., Esbensen, K., Geladi, P. (1987),** ***Principal Component Analysis***. *Chemometrics and intelligent laboratory systems,* 2(1-3), 37-52;
[22] **Zeiler, M. D. (2012),** ***Adadelta: An Adaptive Learning Rate Method***. *arXiv preprint arXiv:1212.5701*;
[23] **Zhang, H. et al. (2020),** ***ELM-MC: Multi-label Classification Framework Based on Extreme Learning Machine***. *International Journal of Machine Learning and Cybernetics,* 11(10), 2261-2274;
[24] **Zheng, W., Qian, Y., Lu, H. (2013),** ***Text Categorization Based on Regularization Extreme Learning Machine***. *Neural Computing and Applications,* 22(3), 447-456.

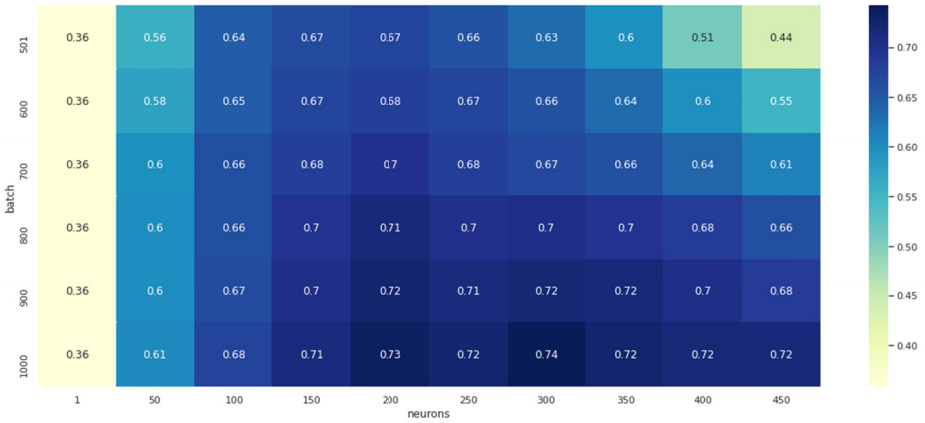
Numele de tari cu majuscule, cred

Annexes



Annex 1. Accuracy reported for different hyperparameter sets using the sum of word embeddings

Source: our computations - Python



Annex 2. Accuracy reported for different hyperparameter sets using the average of word embeddings

Source: our computations - Python