Behaviors of Reservoir Computing Models for Textual Documents Classification

Conference Paper · July 2019

DOI: 10.1109/IJCNN.2019.8852304

1 author:

Nils Schaetti
Haute école de gestion de Genève

Behaviors of Reservoir Computing Models for Textual Documents Classification

Nils Schaetti
Computer Science Department (IIUN)
Université de Neuchâtel
Neuchâtel, Switzerland
nils.schaetti@unine.ch

***Abstract*—Reservoir Computing is a paradigm of recurrent neural network (RNN) models, attractive because of its ease of training and new neuromorphic optoelectronic implementations. Applied with success to time series prediction and speech recognition, few works have so far studied the behavior of these networks on natural language processing (NLP) tasks. Therefore, we decided to explore the ability of Echo State Network-based Reservoir Computing (ESN) models with additional embedding layers to classify text documents of the Reuters C50 data set based on authorship. We explored various learned representations such as word and character embedding and deep feature extractors. Our experiments demonstrate that ESN models can achieve state-of-the-art results on this task and are competitive with common models such as Support Vector Machines (SVM). Moreover, we show that these models compute documents as data streams and could then be able to handle other tasks such as event detection and text segmentation. The best performance is obtained by an ESN with a large reservoir of 1,500 neurons based on word vectors. We think that these results demonstrate the possibility of processing massive quantities of textual data in the future using Reservoir Computing-based systems.**

***Index Terms*—Reservoir Computing, Natural Language Processing, Auhorship Attribution, Recurrent Neural Network**

I. Introduction

Recurrent Neural Network (RNN) models are well suited to tasks such as speech recognition or time series forecasting (financial markets) where there is a functional dependency between continuous samples. Natural languages are composed of time series of tokens where the order of appearance is essential. Moreover, representations of textual documents, such as word embedding, can be seen as temporal multi-dimensional signals, RNNs being able to remember previous inputs and take into account token order. Nevertheless, RNNs are known to be very difficult to train as the gradient vanishes through layers.

In late 2000s, as research on recurrent models were slow and difficult, a new approach named Reservoir Computing has been introduced independently in the field of machine learning under the name Echo State Network (ESN) [1] and in neurosciences under the name Liquid State Machine (LSM) [2]. This approach was developed after discovering that training only the output layer of an RNN randomly constructed gives excellent performances in practice. As they focus only on the output layer and use linear methods during the training procedure, ESNs are easier and faster to train in comparison to other neural models. In ESNs, we separate the part where the computing is done and the output layer where the learning is done which makes them more robust against catastrophic inference. Moreover, Reservoir Computing is interesting from a neuromorphic point of view since several physical opto-electronic and fully-optical implementations have been already proposed.

Who is the true author behind a textual document? This question is the main goal of a well-known task in the field of natural language processing (NLP) named authorship attribution, where the objective is to find who wrote a new unseen document based on a corpus of sample documents and a given set of candidate authors. The classical approaches for authorship attribution are based on statistics, and researchers have tried to apply neural network models with promising results but with very long training time due to high computational complexity.

The recent groundbreaking field of deep learning is very efficient on image and video classification tasks [3]. However, deep learning models applied to authorship attribution have faced more troubles and recurrent neural models, such as Vanilla Recurrent Neural Networks (RNN), Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) reached state-of-art results [4]. Since the rise of deep-learning and neural networks, few works have studied the behavior and performances of RNNs on authorship attribution [4], [5] and only one applied ESNs but to cross-domain authorship attribution [6].

As noticed previously, Reservoir Computing is able to deal with real-time problems and neuromorphic implementations are available. Moreover, Reservoir Computing-based models applied to language processing have long been studied in computational neuroscience [7]. *Why then not use this kind of network to solve natural language processing tasks?* In this paper, as a first step, we study the behavior of ESNs on a text document classification task known as authorship attribution with a set of 15 authors of the Reuters C50 data set. The Reuters_50_50 (C50) dataset is composed of 50 authors with 100 news-wire stories per author.

This paper is organized as follows. In the next section, we discuss related work on authorship attribution and Reservoir Computing. In section 3, we present the ESN model and how to train them. In section 4, we describe how to apply this

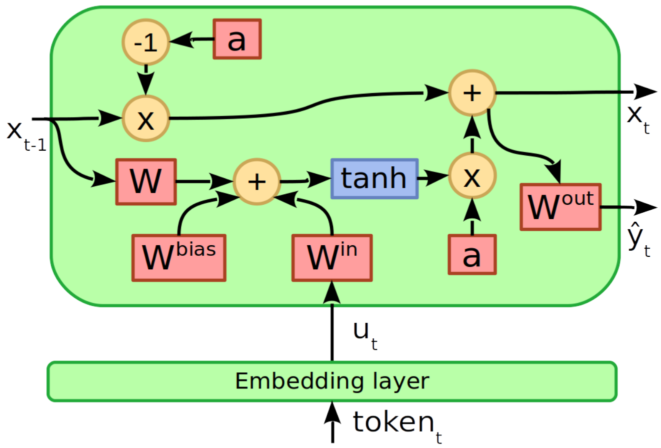

Fig. 1: The complete diagram of the operations carried out at each timestep t by the *Echo State Network* with a pre-trained embedding layer. Matrices and scalar operations are marked in red, functions in blue and basic operations in yellow.

model to natural language processing tasks. In section 5, we propose to apply ESNs to the restrained Reuters C50 data set with various parameters and analyze these results. Finally, in section 6, we discuss our proposal and compare it to other approaches such as Support Vector Machines (SVM) and give some suggestion for future work.

II. Related work

The first work on neural networks applied to authorship attribution has analyzed the work of Shakespeare and Marlowe [8], [9] and the Federalist Papers [10]. Neural networks have since been used to identify the authors of Tamil articles [11] and Dutch poets using character-based features [12].

ESN has been applied to various fields of scientific domains, from astrophysics to robotic motor control and interaction [13]–[16], temporal series forecasting and classification in finance and weather forecasting [17]–[20], to author profiling on social network data [21] and to cross-domain authorship attribution [6]. Moreover, it has been applied to image classification on the MNIST dataset [22], [23] with stacked architectures inspired by deep-learning [24]. Multiple recurrent neural network architectures such as RNNs, LSTMs, and GRUs have been applied to authorship attribution in [4] and [5] respectively with interesting results.

Authorship attribution is a long-studied domain of applied statistics and NLP. Stylometric methods were first applied to the authorship of the disputed "Federalist Paper" [25], a series of 85 political texts written by John Jay, Alexander Hamilton and James Madison, and to the work of Shakespeare [26]. At the beginning of the 20th century, methods based on Bayesian statistical analysis of word frequencies have been applied by Yule and Zipf. Previous works have studied the attribution of long texts such as books [27]–[29] and, recently, short texts such as emails or social network data [30]–[32].

In authorship attribution, different features are found in the scientific literature, from lexical (words and character n-grams) to syntactic ones such as Part-of-Speech tags (PoS) [33]. In this paper, we decided to explore the behaviors of ESNs on various kinds of lexical features.

III. Echo State Network for Text Classification

A. Echo State Networks

Equation 1 [34] describes the main kind of network used in this study. The reservoir state is defined by a non-linear vector x_t,

$$x_{t+1} = (1-a)x_t + af(W^{in}u_{t+1} + Wx_t + W^{bias}) \quad (1)$$

where $x_t \in \mathbb{R}^{N_x}$ is its activation vector at time t, with N_x the number of units in the recurrent layer. The matrix $W^{in} \in \mathbb{R}^{N_x \times N_u}$ represents the connection weights between the inputs u_t and the reservoir. N_u is the dimension of the input signal. $W \in \mathbb{R}^{N_x \times N_x}$ is the matrix of connection weights between the recurrent units. Figure 1 shows the complete computation diagram of an ESN with an embedding layer. We start usually with a null state $x_t = 0$ for the initial vector. The parameter a is the leak rate which allows us to adapt the network's dynamic and W^{bias} is the biases to the reservoir's units. The network's outputs $\hat{y}$ is then defined by,

$$\hat{y}_t = g(W^{out}x_t) \quad (2)$$

where $W^{out} \in \mathbb{R}^{N_y \times N_x}$, with N_y the number of outputs, and W^{out} the output weights matrix. The function g is usually the identity function. The learning phase consists to solve a system of linear equations to minimise the error $E(Y, W^{out}X)$ between the target to be learned and the network's output. The matrix W^{out} can be computed with linear methods, $W^{out}X = Y$, where $X \in \mathbb{R}^{N_x \times T}$ is the matrix containing the reservoir states resulting of the training phase, and $Y \in \mathbb{R}^{N_y \times T}$ is the matrix containing each target outputs, with T the length of the training set. To find W^{out}, one can use the Ridge Regression [35], which minimise the magnitude of the output weights,

$$W^{out} = YX^T(XX^T + \lambda I)^{-1} \quad (3)$$

where λ is the regularisation factor which must be fined tuned for each specific task.

Some parameters should be taken into account at the creation of the reservoir weights. More precisely, it is necessary to ensure the presence of the *Echo State Property* which guarantees that inputs will vanish with time and will not be amplified. As the ESN is a dynamical system, if this property is not present, it could display chaotic behaviors which could lead to a drop in performance. The most commonly used method to ensure that a reservoir has the *Echo State Property* is to set its spectral radius below 1. The spectral radius of a matrix W, noted $\rho(W)$, is its highest absolute Eigenvalue of W.

A lot of work has shown that the ESN has optimal computational performance in dynamics near the border of chaos [36]. The border of chaos is a region of parameters of a dynamical system where it operates at the limit between a non-chaotic and chaotic behavior. This lead in the Reservoir Computing

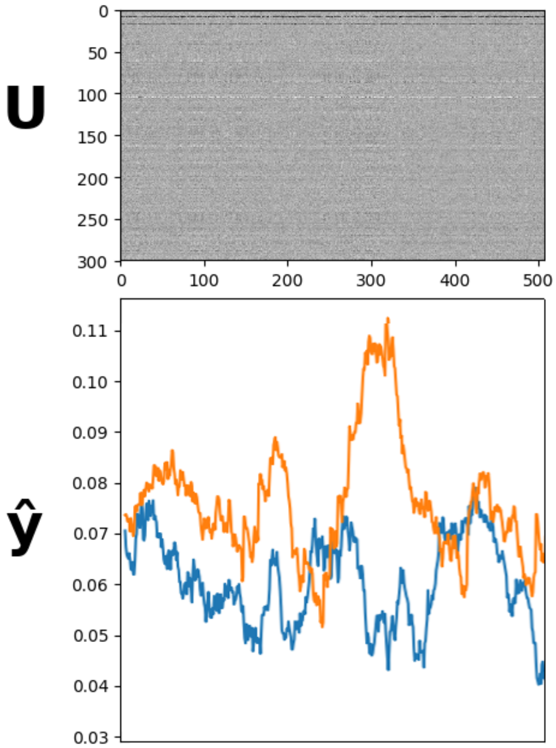

Fig. 2: Inputs and outputs of an ESN for text classification with word embedding. *Top* The input (u_t) time series with 300 dimensions (GloVe). *Bottom* The predicted outputs ($\hat{y}_t$) of two authors representing the probability that the tokens in the ESN's memory have been written by each author.

community to a widely used method consisting to set the spectral radius of W near the unity, but just below, to put the dynamical system near the border of chaos [37].

IV. Application on the Reuters C50 problem

A. From texts to temporal signals

To use ESNs to classify texts, we have to transform texts into temporal representations. Here, we will test a set of three lexical features composed of pre-trained word embedding (WV), pre-trained embedding for character trigrams (C3), and a character encoder (CE).

As a word-based representation, we used vectors pre-trained with Glove, of dimension 300, with a vocabulary of 1.1 million words. These vectors have an average out-of-vocabulary ratio (OOV) on the whole dataset of 4.3%. To process text, we feed the ESN with each token (word, tag or character) in temporal order. The figure 2 shows an example of textual data transformed to a highly-dimensional time series with word embedding (WV).

B. Pre-trained character embedding

To evaluate the performances of these models on character-based representations, a character embedding with a bag-of-words-based model was pre-trained. Here, a neural network has to predict a token based on its context. In our case, we want

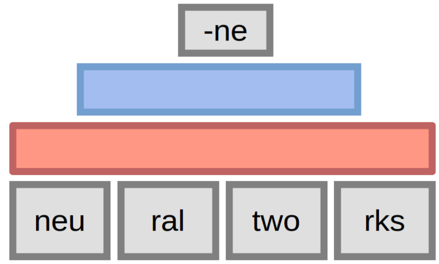

Fig. 3: The full reservoir architecture of the three layers feed-forward neural network used to pre-train character embedding. The bottom grey blocks are the four character trigrams. In red and blue are respectively the embedding layer (size 50) and the fully-connected layer. The final block is the character trigram to predict computed by the softmax function.

to predict the character trigrams (C3) from six surrounding characters on both sides of the target token.

To this end, a four layers feed-forward neural network (FFNN) was trained with the following architecture :

1) An input layer of size 4, one input for every two tokens on each side ;
2) An embedding layer of dimension 60 and vocabulary size $|V|$ of size 167,026 ;
3) A fully connected layer with a ReLU non-linear activity function ;
4) A softmax function as output, of size 167,026. One output for each possible predicted token ;

This model outputs a probability distribution over possible tokens. We trained our model on 230 million examples extracted from Wikipedia for 20 iterations with stochastic gradient descent. The embedding layer was then used as fixed inputs of the ESN model. Figure 3 shows the architecture of the feed-forward model used to pre-train our character trigram embedding.

C. Character-based deep feature extractor

To evaluate if adding layers to the ESN inputs can improve the performances, we trained a feature extractor on data with no temporal dimension and then used it as an input layer. Here a character-based deep feed-forward model was trained first with the classical stochastic gradient descent algorithm. The ESN is then trained with standard linear methods and the feature extractor as input. We will refer to this deep architecture as CE for character encoder.

It takes a series of 20 raw characters as input and proceeds as follow :

1) An embedding layer of dimension 50 ;
2) A fully connected layer composed of 300 features with a ReLU non-linearity ;
3) A softmax function as an output of dimension 15 (one per class);

The model is trained with a learning rate of 0.001 for 300 iterations and cross-entropy as loss function. Once trained,

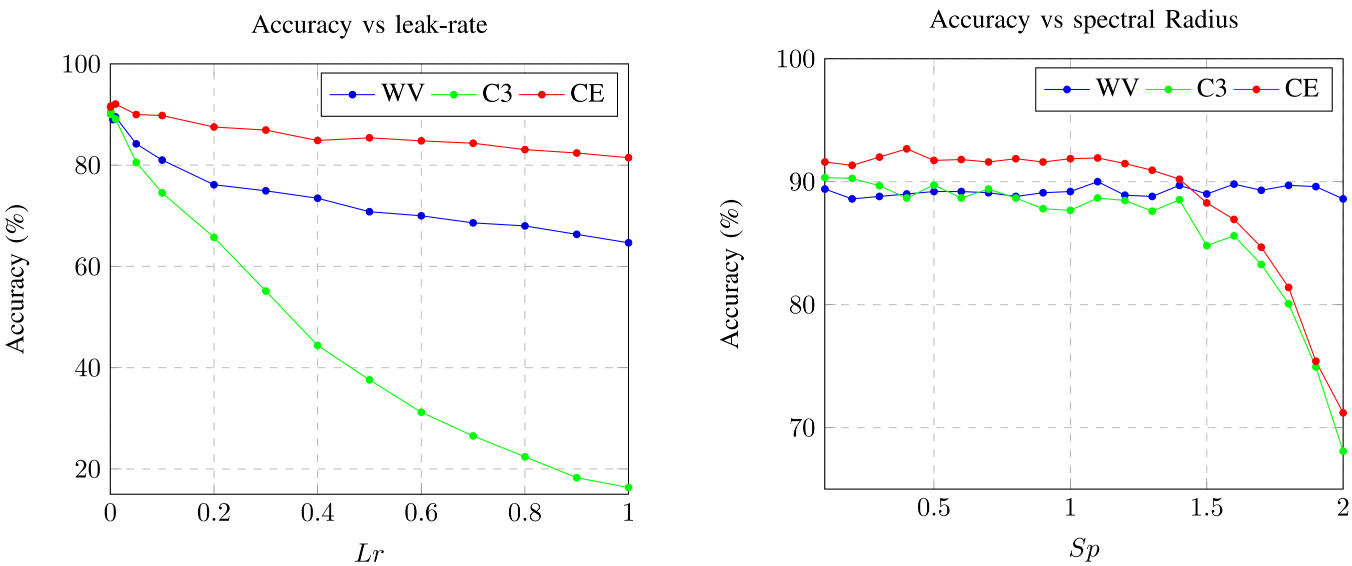

Fig. 4: 10-Fold cross-validation accuracy of the ESN model ($N_x = 1000$) on 15 authors of the Reuters C50 dataset with varying values for the leaky rate and spectral radius parameters.

the output layer is removed, and the linear layer is used as a feature extractor for the ESN inputs.

D. Output layer

The outputs of the ESN are defined by $\hat{y}_{t,a}$, the probability that tokens in the ESN's memory at time t have been written by author a. The result is an output temporal signal of authorship probabilities and we then choose the predicted author $\hat{a}$ of a document as the one with the maximum average over time,

$$\hat{a} = \arg\max_{a} \langle \hat{y}_{t,a} \rangle_t \tag{4}$$

The angular brackets, $\langle \cdot \rangle_t$, are defined as the averaging over time t. Figure 2 shows an example of output stream representing the authors probability at each point of the text. We used here this stream to classify the whole document, but it is interesting to note that these models can give a probability for any parts or sub-parts of the text without redoing the training phase. In addition, with this kind of output, it could be possible to detect interesting events in the text, such as authorship changes, or to segment documents based on classes learned by the model.

V. Results

To apply the model introduced in the previous section and analyze its behaviors, we looked for right parameters for the leak rate and the spectral radius. The leak rate is of particular interest for our study as it defines the dynamical properties of highly-dimensional and temporal representations of text. We want to analyze how the different features behave under different dynamics and if the border of chaos also applies to textual data.

A. Leak-rate and spectral radius

To analyze the dynamical properties of each textual representation, we varied the leak rate parameter from 0.005 to 1.0 and the spectral radius between 0.1 and 2.0. For each parameter value, matrices W, $\overset{in}{W}$ and $\overset{bias}{W}$ were kept constant.

Figure 4 shows the 10-fold cross-validation accuracy for the three features and the various values of the leak-rate and spectral radius parameters. For the leak rate, the best results (92.07%, 90.53%, and 89.53%) were obtained respectively for CE, character trigrams (C3) and word embedding (WV) respectively, with leak rate values of 0.01, 0.001 and 0.01. The accuracy for WV slowly decreases as the dynamic accelerates to reach 64.67% for a leak-rate value of 1.0. On the other hand, the accuracy decreases less for CE when dynamics accelerates going from 92.07% to 81.47%. It is an interesting question for future research to determine if additional pre-trained inputs can make ESN performances more stable over a large range of leak rate values.

The character encoder (CE) is the least affected by the different dynamics and shows much more stability, and word embedding (WV) is less stable, but stay far above a random classifier (6.67%). However, the character-based embedding shows high instability with various leak rate value losing 73.73 points of accuracy going from 90.06% to 16.33%, not so far from a random prediction.

For the spectral radius, it has been shown in the RC literature that the highest ratio between computational power and memory is reached just below the border of chaos [36]. Here we tested different spectral radius for each features, with the best leak rate values found above (0.01, 0.001 and 0.01, for respectively CE, C3 and WV). With word vectors (WV), the highest accuracy is reached with a spectral radius of 1.1 with an accuracy of 90%, but with no significant differences

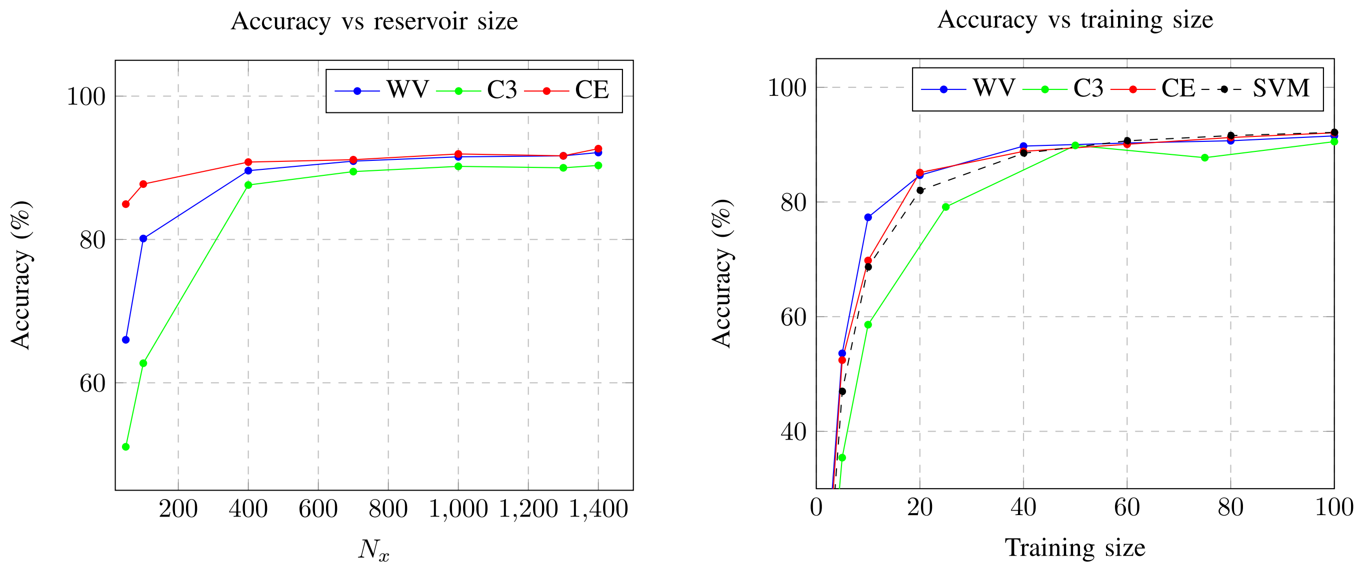

Fig. 5: *Left.* 10-fold cross validation accuracy on 15 authors of the C50 Reuters data set for reservoir sizes between 50 and 1,400 (N_x) on the whole data set. *Right.* 10-fold cross validation accuracy for different dataset sizes between 5% and 100% of the whole dataset.

on the whole spectrum of value, and shows no drop and strong stability up to 2.0.

For the character trigrams (C3), this feature has very stable results for spectral radius up to 1.4 with a maximum of 90.33% for a value of 0.1. Above 1.4, the accuracy drops quickly going from 88.53% to 68.08%. The character encoder (CE), shows a similar pattern as the best accuracy is reached with a spectral radius of 0.5 with 92.67%. As for C3, the accuracy drops quickly above 1.4 going from 90.20% to 71.20% for a spectral radius of 2.0.

These results allow us to make three important observations, first that the dynamics for this task and this kind of data is very slow, secondly, that the word vector representation (WV) show an immunity against chaotic behaviors up to a spectral radius of 2.0. A point which is not present with character-based representations. And finally, no rise of performance is shown just below the border of chaos.

B. Reservoir size and training set

In order to see the influence of increasing the number of units and the possible impact of over-fitting, the accuracy of ESNs with various reservoir sizes (N_x) was evaluated. The left-side of figure 5 show the 10-fold cross-validation accuracy obtained with increasing reservoir sizes from 50 to 1,400 neurons.

For small reservoir sizes ($N_x = 50, N_x = 100$), the character encoder (CE) largely outperforms the other two features with an accuracy of 84.93% and 87.73% respectively, for 50 and 100 units, against 66% and 80.13%, and 51.07% and 62.73% for words and character trigrams. This can be explained by the fact that CE has more units on the inputs and then show more computational power independently of the reservoir size. A second observation is the very poor performances of the character trigrams with small reservoir sizes. It is reasonable to make the assumption that this feature needs more memory as longer sequences of characters are needed to separate classes. For reservoir size above 500 units, the results for WV and CE stay stable around 91%, and the C3 remains significantly lower at 90%.

These models also show an impressive capacity for fast learning from a small dataset. Figure 5 shows the accuracy for each representation and for the additional SVM baseline for a dataset size varying from 5 files per author to the whole dataset composed of 100 files per author. For each evaluation, we proceeded by doing 10-fold cross-validation on a subset of the dataset of the desired size, and this evaluation was done for each possible non-overlapping subsets. The final evaluation is the average accuracy.

For a data set size above 20 files, all classifiers get very close results, but for a dataset of 5 to 20 files the word embedding representation (WV) outperforms the other approaches, especially with a dataset composed of 10 files. The feature encoder (CE) gets results very close the SVM classifier with however a higher accuracy with 20 files. The character-based representation is performing less well with results below other models.

Finally, we wanted to know which textual representation can achieve the best accuracy. To this end, the average performance of each representation on 20 randomly generated reservoirs was evaluated. Figure 6 shows the result of this experience. The average accuracy is respectively of 91.5%, 90.5%, 91.8% and 92.1% for the word embedding (WV), the character trigrams (C3), the character encoder (CE) and the SVM baseline. WV and CE-based ESNs got results significantly better than the C3 representation (t-test, 5%). However, our results show

Classifier	Accuracy	Params
ESN-C3 ($N_x = 1000$)	90.50 %	15,000
ESN-WV ($N_x = 1000$)	90.60 %	15,000
Linear SVM + Word 1-2 grams	92.13 %	317,940
ESN-CE ($N_x = 1400$)	92.46 %	325,800
Best of 50 ESN-WV ($N_x = 1000$)	92.80 %	15,000
Best of 50 ESN-WV ($N_x = 1500$)	93.40 %	22,500

TABLE I: Comparison of 10-fold cross-validation accuracy of baselines and ESN models on 15 authors of the C50 Reuters data set along with the number of parameters to be learned of each model.

no sign of significant differences between the WV and CE. The SVM baseline got an accuracy significantly higher (92.1%) than WV (91.5%) and CE (91.8%).

C. Comparison

Table I shows the comparison of our different models and representation with the Support Vector Machines (SVM) baseline based on word and word bigrams. These baseline models were implemented using the Sklearn framework. The third column of table I shows the number of parameters to be learned for each model. The number of parameters to learn of an ESN model is the size of the output matrix ($\overset{out}{W}$). In comparison to baseline models, ESN models, except ESN based on CE, have a lot fewer parameters to learn. In addition to the fact that they can be learned with linear methods, this shows that ESNs are much easier to train compared to other applications of neural models to natural language processing.

Our best model is an Echo State Network (ESN) with a reservoir of 1,500 units ($N_x = 1500$) using word-based representation. To find this model, we generated 50 ESNs with random internal weights matrix (W) and selected the one with the best 10-CV accuracy. This model obtained an accuracy of 93.40%, a model with 1,000 units ($N_x = 1000$) found with the same principle got the second best result (92.80%). The three other ESN models, namely average reservoirs based respectively on CE, word embedding and character trigrams, obtained the third, fifth and sixth places with 92.46%, 90.60%, and 90.50%. In comparison, the baseline model obtained an accuracy of 92.13%, just below our ESN-CE model, and significantly lower than our top two models.

VI. Discussions

In this paper, we investigated the performance of Echo State Network-based Reservoir Computing (ESN) on a text document classification task with 15 authors of the C50 Reuters dataset and demonstrated that ESNs are able to compete with Support Vector Machines (SVM). Our results also show that the deep character encoder and the word-based features are the most efficient representations, followed by the character

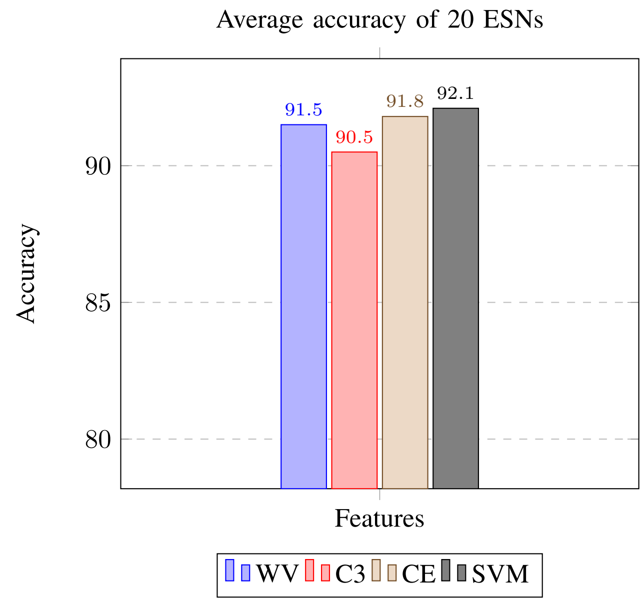

Fig. 6: The average 10 fold cross-validation accuracy of 20 ESNs model with 1,000 neurons on 15 authors of the Reuters C50 dataset.

trigrams. We tested the possibility of using deep feature extractor as ESN inputs and our results show that it has not significantly improved the accuracy compared to word-based embedding layer, and it also strongly increased the training time, going from a few minutes to several hours.

The useful conclusion of this study is the impact of the leak rate parameter which shows that ESN's dynamics is very slow with textual data. A piece of important information for any member of the research community who would like to apply ESN to natural language processing. In contrary to other features, words embedding show much more stability against chaotic behaviors over a large range of spectral radius.

Beyond these preliminary results, a lot of applications to physical implementations of Reservoir Computing are possible in fields such as optoelectronic computers and neuromorphic nonlinear transient computing (NTC). The application of our work on this kind of new computing approaches could lead to new possibilities in the field of real-time speech and text processing.

Lately, a new field of research named *Deep Reservoir Computing* has tried to apply deep learning methods to Reservoir Computing with stacked architectures such as *Stacked-ESN* and *Deep-ESN* in order to create hierarchical temporal representations of data. In the future, we want to compare these new approaches with the results presented in this paper. But we also want to test more complex architectures trained with stochastic gradient descent such as Long Short-Term Memory (LSTM) and GRU (Gated Recurrent Units) with the same variety of features.

In the future, we also want to test ESN on other datasets and apply these models to collaborative work and to social media data. As we demonstrated in section 4, these models see texts as streams and give as output the class probabilities for every point of the given text. Consequently, we want to explore the possibility to apply these models to detect particular points of interest in a text such as changes in authorship (author identification and profiling), emotional (sentiment analysis) or semantic (topic modelling) content of the text. ESN applied to these tasks could be useful for application such as document segmentation, plagiarism detection and computer forensics. We think that the work presented in this study shows the interest of ESN models applied to natural language processing tasks.

References

[1] Herbert Jaeger. *Short term memory in echo state networks*, volume 5. GMD-Forschungszentrum Informationstechnik, 2001.

[2] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[3] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[4] LZ Wang. News authorship identification with deep learning, 2017.

[5] Chen Qian, Tianchang He, and Rao Zhang. Deep learning based authorship identification.

[6] Nils Schaetti. Bidirectional echo state network-based reservoir computing for cross-domain authorship attribution : Notebook for pan at clef 2018. *CLEF2018*, 2018.

[7] Peter Ford Dominey. Recurrent temporal networks and language acquisition—from corticostriatal neurophysiology to reservoir computing. *Frontiers in psychology*, 4:500, 2013.

[8] Robert AJ Matthews and Thomas VN Merriam. Neural computation in stylometry i: An application to the works of shakespeare and fletcher. *Literary and Linguistic Computing*, 8(4):203–209, 1993.

[9] Thomas VN Merriam and Robert AJ Matthews. Neural computation in stylometry ii: An application to the works of shakespeare and marlowe. *Literary and Linguistic Computing*, 9(1):1–6, 1994.

[10] Fiona J Tweedie, Sameer Singh, and David I Holmes. Neural network applications in stylometry: The federalist papers. *Computers and the Humanities*, 30(1):1–10, 1996.

[11] Urszula Stańczyk and Krzysztof A Cyran. Machine learning approach to authorship attribution of literary texts. *International journal of applied mathematics and informatics*, 1(4):151–158, 2007.

[12] Johan F Hoorn, Stefan L Frank, Wojtek Kowalczyk, and Floor van Der Ham. Neural network identification of poets using letter sequences. *Literary and Linguistic Computing*, 14(3):311–338, 1999.

[13] Junichi Kuwabara, Kohei Nakajima, Rongjie Kang, David T Branson, Emanuele Guglielmino, Darwin G Caldwell, and Rolf Pfeifer. Timing-based control via echo state network for soft robotic arm. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.

[14] Paul G Plöger, Adriana Arghir, Tobias Günther, and Ramin Hosseiny. Echo state networks for mobile robot modeling and control. In *RoboCup 2003: Robot Soccer World Cup VII*, pages 157–168. Springer, 2004.

[15] Eric A Antonelo, Benjamin Schrauwen, Xavier Dutoit, Dirk Stroobandt, and Marnix Nuttin. Event detection and localization in mobile robot navigation using reservoir computing. In *Artificial Neural Networks–ICANN 2007*, pages 660–669. Springer, 2007.

[16] Eric Antonelo, Benjamin Schrauwen, and Dirk Stroobandt. Mobile robot control in the road sign problem using reservoir computing networks. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 911–916. IEEE, 2008.

[17] Francis Wyffels and Benjamin Schrauwen. A comparative study of reservoir computing strategies for monthly time series prediction. *Neurocomputing*, 73(10):1958–1964, 2010.

[18] Paulin Coulibaly. Reservoir computing approach to great lakes water level forecasting. *Journal of hydrology*, 381(1):76–88, 2010.

[19] A.A. Ferreira, T.B. Ludermir, R.R.B. de Aquino, M.M.S. Lira, and O.N. Neto. Investigating the use of reservoir computing for forecasting the hourly wind speed in short -term. In *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 1649–1656, June 2008.

[20] Xiaowei Lin, Zehong Yang, and Yixu Song. Short-term stock price prediction based on echo state networks. *Expert Systems with Applications*, 36(3, Part 2):7313 – 7317, 2009.

[21] Nils Schaetti and Jacques Savoy. Comparison of neural models for gender profiling.

[22] Nils Schaetti, Raphaël Couturier, and Michel Salomon. Reservoir computing: Étude théorique et pratique en reconnaissance de chiffres manuscrits, mémoire de master. 2015.

[23] Nils Schaetti, Michel Salomon, and Raphaël Couturier. Echo state networks-based reservoir computing for mnist handwritten digits recognition. *19th IEEE International Conference on Computational Science and Engineering (CSE 2016)*, 2016.

[24] Azarakhsh Jalalvand, Glenn Van Wallendael, and Rik Van de Walle. Real-time reservoir computing network-based systems for detection tasks on visual contents. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2015 7th International Conference on*, pages 146–151. IEEE, 2015.

[25] Jacques Savoy. The federalist papers revisited: A collaborative attribution scheme. *Proceedings of the Association for Information Science and Technology*, 50(1):1–8, 2013.

[26] Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, 9(214):237–249, 1887.

[27] Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification using functional lexical features. *Journal of the Association for Information Science and Technology*, 58(6):802–822, 2007.

[28] Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the Association for Information Science and Technology*, 60(1):9–26, 2009.

[29] Efstathios Stamatatos, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. Overview of the pan/clef 2015 evaluation lab. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 518–538. Springer, 2015.

[30] Roy Schwartz, Oren Tsur, Ari Rappoport, and Moshe Koppel. Authorship attribution of micro-messages. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1880–1891, 2013.

[31] Farkhund Iqbal, Rachid Hadjidj, Benjamin CM Fung, and Mourad Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. *digital investigation*, 5:S42–S51, 2008.

[32] Robert Layton, Paul Watters, and Richard Dazeley. Authorship attribution for twitter in 140 characters or less. In *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*, pages 1–8. IEEE, 2010.

[33] Moshe Koppel and Jonathan Schler. Exploiting stylistic idiosyncrasies for authorship attribution. In *Proceedings of IJCAI'03 Workshop on Computational Approaches to Style Analysis and Synthesis*, volume 69, pages 72–80, 2003.

[34] Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[35] Tikhonov AN. Solution of incorrectly formulated problems and the regularization method. *Soviet Math Dokl 4, English translation of Dokl Akad Nauk SSSR*, 151:501–504, 1963.

[36] Joschka Boedecker, Oliver Obst, Joseph T Lizier, N Michael Mayer, and Minoru Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 2012.

[37] Izzet B Yildiz, Herbert Jaeger, and Stefan J Kiebel. Re-visiting the echo state property. *Neural networks*, 35:1–9, 2012.