Time-evolving Text Classification with Deep Neural Networks

Yu He[1,2], **Jianxin Li**[1,2], **Yangqiu Song**[3], **Mutian He**[1,2], **Hao Peng**[1,2]
[1]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China
[2]State Key Laboratory of Software Development Environment, Beihang University, China
[3]Department of Computer Science and Engineering, HKUST, Hong Kong
{heyu, lijx, mutianhe, penghao}@act.buaa.edu.cn, yqsong@cse.ust.hk

Abstract

Traditional text classification algorithms are based on the assumption that data are independent and identically distributed. However, in most non-stationary scenarios, data may change smoothly due to long-term evolution and short-term fluctuation, which raises new challenges to traditional methods. In this paper, we present the first attempt to explore evolutionary neural network models for time-evolving text classification. We first introduce a simple way to extend arbitrary neural networks to evolutionary learning by using a temporal smoothness framework, and then propose a diachronic propagation framework to incorporate the historical impact into currently learned features through diachronic connections. Experiments on real-world news data demonstrate that our approaches greatly and consistently outperform traditional neural network models in both accuracy and stability.

1 Introduction

Text classification is a fundamental problem in data mining. Traditional classification algorithms are based on the assumption that data are independent and identically distributed. However, in many real applications like news topic classification [Allan *et al.*, 1998; Kim and Hovy, 2006], event detection and tracing [Yang *et al.*, 1998; Atefeh and Khreich, 2015], the data to be learned are not static. For example, topics in online social media and news media are usually observed sequentially from a series of time periods, and their distribution changes smoothly over time. Very often, such change mainly consists of two parts: *long-term evolution* due to concept drift and *short-term fluctuation* due to noise disturbance. For example, the concept of topic "media" commonly meant traditional print media like newspaper before 1960s, and after that it slowly drifted with the emergence of Internet, and today it more refers to the emerging new media including microblogs, podcasts, etc. Meanwhile, it is natural to expect that the overall interests of social media may fluctuate temporarily due to some emergency events during the long-term evolution.

These non-stationary scenarios create what we call time-evolving data or evolutionary data, which raises new challenges to traditional text classification. First, it is undesirable to simply use a single static classifier for all time periods. Otherwise, as the data distribution changes over time, the classifier may only learn some over-generalized features, thus fail to either reflect the nuances between different periods or fit the data of a specific time period. Second, divergences even contradictions lie between data of different time periods due to long-term evolution and short-term fluctuation. Thus, incorporating all the historical and current data to train a current classifier makes little sense in non-stationary scenarios. Even in stationary or non-evolutionary scenarios, training with simple combination of all historical data is also worthless and inapplicable due to the computation cost. Third, it is also unfavorable to obtain a series of completely independent classifiers for each time period, since in this way we may lose extensive valuable information from adjacent time periods. What's worse, when only a slice of all time periods is considered, the classifier may overly fit the slice data of the corresponding time period and not be able to distinguish between long-term data changes and short-term data fluctuations.

In this paper, we propose two neural network frameworks to learn a chain of evolving classifiers for time-evolving data. In the first framework, we introduce the temporal smoothness into learning process to train evolutionary neural network classifiers. In the second framework, we design a diachronic propagation mechanism to incorporate the historical impact into currently learned features. In both frameworks, the classifier can fit the corresponding slice data in each time period as much as possible, thus maintaining high sensitivity to the long-term data changes. In addition, the classifier can also take recent periods into account and not deviate too much to fit the historical data, thus maintaining high robustness to the short-term data fluctuations. Experiments on real-world news data demonstrate that our methods clearly and consistently outperform traditional neural network models in both accuracy and stability. Our main contributions are as follows:

1. We introduce the basic formulation of evolutionary classification with neural networks. To our best knowledge, it is the first time that the evolutionary case of neural network learning is explored to classify time-evolving data.
2. We introduce a basic evolutionary approach based on temporal smoothness framework and propose a diachronic propagation framework to extend arbitrary neural networks to evolutionary learning.

3. We conduct extensive experiments and case studies on two real-world news datasets and three synthetic datasets. The empirical results demonstrate that our approaches clearly and consistently improve the current start-of-the-art neural network models.

The code is available at https://github.com/RingBDStack/Time-evolving-Classification.

2 Related Work

As traditional static machine learning algorithms are incapable for the goal to reflect long-term data evolution and maintain robustness against short-term data fluctuations, a new topic of evolutionary learning is introduced to deal with time-evolving data [Chakrabarti *et al.*, 2006] and has attracted significant attention in recent years. For the unsupervised case of evolutionary learning problem, multiple classical clustering algorithms have been extended to evolutionary versions, such as k-means clustering [Chakrabarti *et al.*, 2006], spectral clustering [Chi *et al.*, 2007; Ning *et al.*, 2007; Tang *et al.*, 2008], Gaussian mixture model [Zhang *et al.*, 2009], etc. Moreover, Wang et al. [2012] proposed a general model for clustering large-scale evolutionary data based on low-rank kernel matrix factorization. Xu et al. [2014] proposed an adaptive evolutionary clustering method by tracking the time-varying proximities between objects followed by static clustering. Furthermore, Jia et al. [2009] first considered the semi-supervised evolutionary learning problem, and proposed a general framework based on the Reproducing Kernel Hilbert Space. However, up to now, the work on the supervised case of evolutionary learning is still limited, especially on the time-evolving text classification.[1]

Relevantly, deep learning models have proven to be state-of-the-art methods in various applications including text classification [Liu *et al.*, 2017]. Particularly, recurrent neural networks (RNN, LSTM, GRU, etc.), which use feedback loops to exhibit dynamic temporal behavior, have shown promising results on sequential data classification issues such as speech recognition [Graves *et al.*, 2013; Graves and Jaitly, 2014] and document classification [Tang *et al.*, 2015; Yang *et al.*, 2016]. However, they are still traditional methods dealing with static data with identical distribution, with the difference that each sample is a sequence of data (e.g., each document consists of a sequence of words). To our best knowledge, deep learning models have not been explored for evolutionary learning scenario to reflect the long-term changes of dynamically distributed data, which is our focus in this article.

Online learning is also related to our work, which is used to update the decision model continuously as new data arrives [Gama *et al.*, 2014] and has been extensively studied recently [Crammer *et al.*, 2006; Masnadi-Shirazi and Vasconcelos, 2010; Duchi *et al.*, 2011]. Although online learning targets the similar problem setting as evolutionary learning, they are essentially different. On the one hand, online learning is mostly used in areas where it is computationally infeasible to train over the entire dataset and is under the same assumption that data are independent and identically distributed as traditional static machine learning. On the other hand, although there are also some studies on online learning in non-stationary scenarios such as concept drift [Gama *et al.*, 2014; Ghazikhani *et al.*, 2014], they mainly focus on the snapshot changes between old data and new data to improve adaptability to new data and to reduce learning costs, while do not concern about the long-term data trends and interactions, which is one of the core requirements of evolutionary learning.

[1]Here we distinguish the concept of evolutionary classification on time-evolving data from evolutionary algorithms (EAs) motivated by biological evolution. Up to now, EAs are still traditional optimization algorithms to perform evolutionary computation on static data, which is unrelated to the problem discussed in this paper.

3 Time-evolving Text Classification

In this section, we first introduce the basic formulation of supervised evolutionary classification problem and then propose two evolutionary neural network (NN) frameworks.

3.1 Problem Formulation

Unlike traditional static classification, in which the data are assumed independent and identically distributed, the goal for evolutionary classification is to train a chain of evolving classifiers on time-evolving data to reflect the long-term evolution and to maintain robustness against the short-term fluctuation.

In evolutionary classification applications, time is regularly sliced into a sequence of steps according to certain periods, which presents a sequence of sliced data, and the distribution of these data usually changes over time. Assuming there are T consecutive time steps, we are given a set of sequential data sets $\mathbb{X} = \{X_1, X_2, ..., X_T\}$, and each subset X_t contains n_t data points generated in time step t with an unknown data distribution $P(x;t)$: $X_t = \{x_1^t, x_2^t, ..., x_{n_t}^t\} (t = 1, 2, ..., T)$, along with a corresponding set of class labels $Y_t = \{y_1^t, y_2^t, ..., y_{n_t}^t\}$ where y_i^t represents the class label of data point x_i^t. As the posterior distribution $P(y|x;t)$ usually changes smoothly over time, the classification functions of different time steps are time-evolving. That is, the classifier f_t is not only reflected from the mapping relationship between X_t and Y_t which sampled in time step t, but also evolved from previous classifiers $\{f_i\}_{i=1}^{t-1}$. More generally, the task of time-evolving data classification is to find T consecutive time-evolving classifiers $\{f_1, f_2, ..., f_T\}$ to reflect the mapping relationships $\{f_t : X_t \rightarrow Y_t\}_{t=1}^{T}$ for all time steps. As for neural networks, it is to train the corresponding T sets of evolving model parameters $\{\theta_1, \theta_2, ..., \theta_T\}$ on time-evolving data sets $\mathbb{X}$.

3.2 NN with Temporal Smoothness Framework

Temporal smoothness framework was first proposed by Chakrabarti et al. [2006] in evolutionary clustering problem, which is also applicable for the general learning problem on time-evolving data. Based on the smoothness assumption of time-evolving data, this framework aims to smooth the learning results over time as much as possible while remaining faithful to the current data. Following this framework, classifiers in evolutionary classification task should not change dramatically from one time step to the next. In other words, the classifiers f_{t_1} and f_{t_2} are expected to be more similar when time step t_1 and t_2 are closer. As for neural network

models, since these models at different time steps only differ in model parameters and share the same model structure, we could ensure the similarity by preventing large deviation of model parameters during the training process.

Therefore, for evolutionary classification problem, we propose to seek the optimal classifier f_t at each time step $t(2 \leq t \leq T)$ by minimizing the following objective function:

$$\mathcal{J}(f_t) = C_t + H_t, \tag{1}$$

where the first term C_t is the current cost function defined on current data, and the second term H_t is the historical distance penalty derived from the distance between current and historical classifiers. Specifically, we use negative log-likelihood losses to measure the cost on each data point (x, y):

$$\mathcal{L}(x, y, \theta) = -\sum_{l \in L} \ell(l, y) \log f(l \mid x, \theta), \tag{2}$$

where L is the class labels set of the whole data, $\ell(l, y)$ refers to the true probability that the class label of data point x is l, and $f(l \mid x, \theta)$ is the predicted likelihood produced by the neural network classifier f with parameters θ. Then the whole cost function in time step t is like following:

$$C_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}(x_i^t, y_i^t, \theta_t). \tag{3}$$

Using θ_i to denote the model parameters of classifier f_i at time step i, H_t could be expressed as follows:

$$H_t = \sum_{i=1}^{t-1} \alpha_t^i \|\theta_t - \theta_i\|, \tag{4}$$

where the historical parameters $\theta_i (1 \leq i \leq t-1)$ are known and fixed, which have been well-trained in an earlier time; α_t^i is the weight to reflect user's emphasis on current cost and historical distance penalty. Generally, α_t^i is exponentially decaying over time as follows:

$$\alpha_t^i = a \cdot \mathrm{e}^{-(t-i-1)}, \tag{5}$$

where a is the initial decay factor to control the scale of historical distance penalty.

Following this way, we finally obtain a basic evolutionary approach to extend an arbitrary neural network model to seek a chain of evolving classifiers on time-evolving data, and we call it NN with temporal smoothness (NN-TS).

3.3 NN with Diachronic Propagation Framework

In general, a neural network classifier contains two components [Ganin *et al.*, 2016], feature extractor to extract the high-level representation of the samples and label predictor to predict the probability of each sample belonging to a label. Specially, for evolutionary classification problem, as classifiers are usually trained successively for each time step, there are a chain of evolving extractors and predictors, and these components only differ in model parameters and share the same network structure at different time steps. In this case, for a data point x, there is a chain of features extracted from those time-evolving extractors, and these features are usually not independent of each other. Thus, we propose a diachronic propagation framework to incorporate these sequential features produced by evolving extractors.

A neural network with diachronic propagation framework usually contains more than one row (as shown in Figure 1), corresponding to successive time steps on time-evolving data. It starts with a traditional row without any specialties: a feature extractor E_1 to extract sample's feature representation and a label predictor P_1 to predict the probability based on the feature extracted. When we shift to a second time step, E_1 and P_1 have been trained well and their model parameters are frozen and immutable, and a new row of feature extractor E_2 and label predictor P_2 will be trained. But different from the first row, a diachronic connection layer is added and trained, it combines the features extracted by E_1 and E_2 and generates a new feature with the same dimension. We use h_i to denote the feature extracted by extractor E_i and its dimension is k, then the diachronic propagation process via the diachronic connections could be expressed as follows:

$$h_2^* = g(W_2[h_2, h_1]), \tag{6}$$

where $W_2 \in \mathbb{R}^{k \times 2k}$ is the weight matrix of diachronic connections in second row, $[h_2, h_1]$ is the concatenation of h_2 and h_1 and is as the input of the diachronic connections, h_2^* is the output of the diachronic connections, g is an element-wise activation function such as RELU [Nair and Hinton, 2010].

Similarly, when switching to a deeper time step, a deeper row shall be appended and only that row is trainable. Then a deeper diachronic propagation occurs by recursively combining the extracted features via deep diachronic connection layers. A deep diachronic neural network with t $(t \geq 3)$ rows is shown in Figure 1, and the deep diachronic propagation could be recursively expressed as follows:

$$h_i^* = g(W_i[h_i, h_{i-1}^*]), \quad i = 3, ..., t, \tag{7}$$

where $W_i \in \mathbb{R}^{k \times 2k}$ is the weight matrix of diachronic connections in layer i. The last h_t^* is the composited feature produced by the deep diachronic connection layers, which is propagated to the last predictor P_t. Finally, P_t produces a probability distribution to perform classification similar to traditional neural networks.

In this diachronic propagation framework, we can find that features will propagate diachronically along successive time steps while forward propagating along the current time step. By combining currently and previously learned features in this manner, this framework achieves a richer compositionality, which is beneficial to learn the dynamic characteristic in non-stationary scenarios and to reflect long-term changes on time-evolving data. Moreover, the diachronic connections could naturally transfer prior knowledge across different time steps, making it more capable of keeping robust to temporary noise by incorporating historical information.

Following this framework, the output probability distribution predicted by neural networks not only represents the current data feature by forward propagation, but reflects the impact of historical information. By incorporating the current features and historical impact, the final evolving classifier function could be expressed as follows:

$$f_t(y|x, \theta) = P_t\left(g(W_t[h_t, h_{t-1}^*])\right), \tag{8}$$

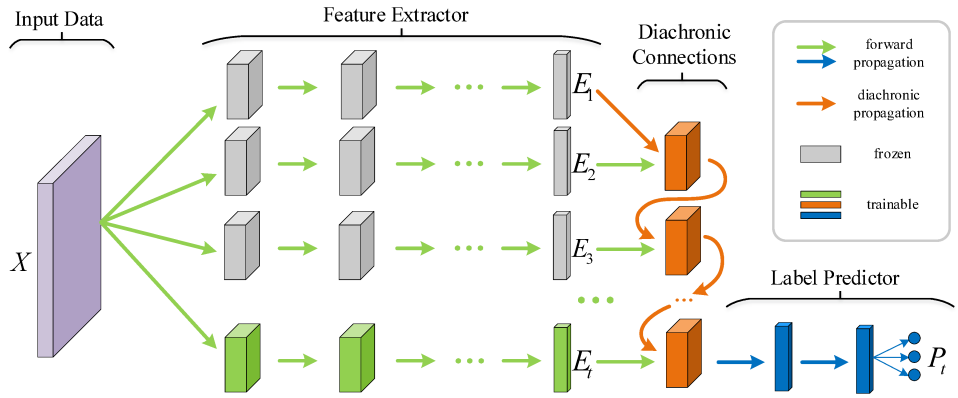


Figure 1: The proposed framework with deep diachronic connections. In this architecture, for input data x, there is a series of features extracted by time-evolving extractors at previous time steps, noted by the subscript $1, 2, ..., t$. Then, all features are fed to the deep diachronic connections. By combining diachronic propagated features and forward propagated features, the diachronic connections generate an output with the same dimension at the last time step, which is used as the final composited feature to perform classification. Note that only the current extractor E_t, predictor P_t and the diachronic connections are trainable, the historical extractors $E_i (1 \leq i \leq t-1)$ which have been trained well in an earlier time are frozen and immutable.

where h_t is the current feature propagated via extractor E_t; h^*_{t-1} is the diachronic feature propagated via diachronic connections. Following this function, the diachronic propagation framework trains the model parameters by minimizing the cost function completely identical with traditional neural networks (similar as Eq.(3)). Finally, we denote this evolving framework as NN with diachronic propagation (NN-DP).

4 Experiments

In this section, we report experimental results to show effectiveness and efficiency of our approaches.

4.1 Datasets

We conduct text classification experiments on two real-world news datasets and three derived datasets.

• **NYTimes:** NYTimes is a large-scale corpus contains nearly every article published in the New York Times between January 01, 1987 and June 19th, 2007 [Sandhaus, 2008]. We select subtags of NEWS according to the taxonomy of News Desk, and finally we have 26 reasonable categories in total.

• **RCV1:** RCV1 is a manually labeled newswire collection of Reuters News from 1996 to 1997 [Lewis *et al.*, 2004]. The news documents are categorized with respect to three controlled vocabularies: industries, topics, and regions. We traverse the topic hierarchy tree from the root to find disjoint subtrees (i.e., topics) with at least two branches (i.e., subtopics). Finally, we obtain a sub-corpus with 12 subtrees (as shown in Table 1) and each subtree has at least two leaves whose concepts are inclusive in their parent node.

Furthermore, we construct three synthetic datasets derived from **RCV1** to better evaluate the effectiveness and stability of our proposed methods on different evolutionary scenarios.

• **RCV1-org:** Based on the 12 subtrees in **RCV1**, we choose half of the leaf nodes to represent the top category for each subtree, as a result RCV1-org dataset is produced with 12 categories and each of which contains half of the concepts in the corresponding subtree.

• **RCV1-noise:** We perturb the **RCV1-org** dataset by adding pseudo-random noise in each time step to evaluate the robustness and stability of proposed methods. For different levels of comparison, we divide the time steps into 4 groups with noise ratios σ of 0, 0.15, 0.3, 0.45, respectively.

• **RCV1-drift:** In RCV1-org dataset, we have half of concepts of each category in each time step, noted as old concepts. In this dataset, we use the different half of concepts noted as new concepts to simulate a drift process of topic concepts. For each category, we resample the concepts using a $\mathcal{S}$-type probability $s_t = \frac{1}{1+e^{-(t-6)}}$ to make the new concepts growing with the speed of s_t over time, and the old concepts oppositely decaying with the speed of $1 - s_t$. By speeding up a man-made evolution process in this way, we obtain a new time-evolving dataset with a stronger drift.

Finally, we treat each year as a time step for NYTimes dataset and get 10 time steps from 1997-2006, treat each month as a time step for RCV1 datasets and have 12 time steps from 1996.09-1997.08. For each time step, we use half of the data for training and the other half for testing. The dataset statistics are summarized in Table 2.

4.2 Baseline Methods and Experimental Settings

We apply the following three state-of-the-art neural models to evaluate the efficacy and adaptability of proposed methods.

• **TextCNN:** TextCNN [Kim, 2014] is a popular CNN-based model and has achieved the current state-of-the-art in text classification [Liu *et al.*, 2017].

• **RCNN:** RCNN [Lai *et al.*, 2015] is a CNN and RNN based model with a recurrent convolutional structure.

• **HAN:** HAN [Yang *et al.*, 2016] is a RNN-based model with two-level GRU-based sequence encoder and hierarchical attention mechanism.

For each baseline, we implement its evolutionary versions by applying temporal smoothness framework (**NN-TS**) and diachronic propagation framework (**NN-DP**), and evaluate their classification performances in each time step.

For the input data representation, we use public release

Sub-root	C15	C17	C18	C31	E13	E14	E21	E31	E51	G15	M13	M14
Leaf Number	2	4	3	3	2	3	2	3	3	9	2	3
Total Samples	138,754	28,118	40,442	23,457	5,656	1,708	34,074	2,174	14,481	11,055	48,590	74,932

Table 1: Statistics of the 12 subtrees branched from the topic hierarchy in RCV1.

Dataset	Time-steps	#(Doc)	#(Len)	#(Labels)
NYTimes	1997.01-2006.12	627,915	629	26
RCV1	1996.09-1997.08	403,143	240	12

Table 2: Dataset statistics: #(Doc) is the total number of documents; #(Len) is the average tokens number per document; #(label) is the number of class labels.

of word2vec [Mikolov *et al.*, 2013] to train 100-dimensional word embeddings from Wikipedia corpus based on CBOW model. For training neural network models, we use mini-batch stochastic gradient descent (SGD) optimizer to minimize the corresponding objective functions, and the common parameters are empirically set, such as $batch\ size$ as 128 and $moving\ average$ as 0.999, etc.

To reduce error, we repeat each experiment five times independently and use the mean value as the final experiment result. We also try our best to tune the models for both our methods and baseline methods. As a result, all the experiments of each method produce best results to our best efforts.

4.3 Results and Analysis

We compared state-of-the-art neural models (TextCNN, RCNN, HAN) and their evolutionary versions based on temporal smoothness framework (suffixed by TS) and diachronic propagation framework (suffixed by DP) proposed in this paper, and Table 3 and Table 4 are their experimental results for multi-class classification in different time steps on NYTimes and RCV1 datasets. We evaluate each dataset with two metrics, accuracy (higher is better) including a single value in each time step and an average value in all time steps, and Std (Standard Deviation, lower is better) among all time steps.

From the experimental results, it is observed that the evolutionary methods clearly and consistently outperform the standard neural network models on different datasets and different time steps, especially for RCNN-DP, which achieves the best performance compared with other methods (averagely 3% improvement on NYTimes and 2% improvement on RCV1). Generally, by applying evolutionary frameworks on different neural network models, we improve the classification accuracies in different time steps by 1%-6% on NYTimes and by 1%-3% on RCV1. The results prove the effectiveness of the proposed evolutionary frameworks.

The experimental results demonstrate the advantage of the evolutionary methods in two aspects. First, by considering the long-term data changes and interactions within consecutive time steps, evolutionary models can effectively capture the evolutionary information of features compared with traditional neural network models, and clearly help to increase the classification accuracies on time-evolving data. Second, over the whole time steps of each dataset, we can observe that it can maintain a more stable and smooth result by applying evolutionary frameworks, which suggests that the evolutionary methods are more robust against noise and more suitable for handling the data fluctuations in non-stationary scenarios. In addition, we can find the diachronic propagation framework achieves better performances than temporal smoothness framework in most cases, which indicates that the diachronic connection is more effective to adaptively learn the evolution characteristic than a simple regularization term of parameter distance. We also find that the evolutionary methods bring about more improvements on the NYTimes dataset than the RCV1 dataset. The reason could be that the NYTimes dataset has a much longer time period, and hence has a stronger long-term evolution, making it more conducive to evolutionary learning. Overall, the proposed evolutionary frameworks have good adaptability and efficacy, and can be easily extended to different neural network models and achieve much better classification performances on both long-term and short-term time-evolving data.

4.4 Case Study

To better understand the effectiveness of proposed methods on different evolutionary scenarios, we conduct more experiments on three synthetic datasets introduced in Section 4.1, and the results are shown in Figure 2-(a)(b)(c).

From the classification results of RCNN and its evolutionary versions on RCV1-org, RCV1-noise, and RCV1-drift, we can consistently conclude the efficacy and stability of proposed methods as analyzed in section 4.3. Moreover, compared with the results on RCV1-org and RCV1-noise, we can see that the standard RCNN is very sensitive to noise, and the accuracy declines sharply as the noise ratio σ increases. In contrast, by incorporating historical information to current decision, the evolutionary versions RCNN-TS and RCNN-DP are quite robust against noise. Compared with the results on RCV1-org and RCV1-drift, it is easy to find that data drift greatly affects the performance of text classification. When there is a stronger drift between different time steps, the standard RCNN performs much worse, especially near the inflection point of the data drift curve with maximum drift (time step = 6). Whereas, RCNN-TS and RCNN-DP can still achieve meaningful results, which proves that the proposed methods are quite effective to reduce the impact of drift and to reflect the long-term data changes.

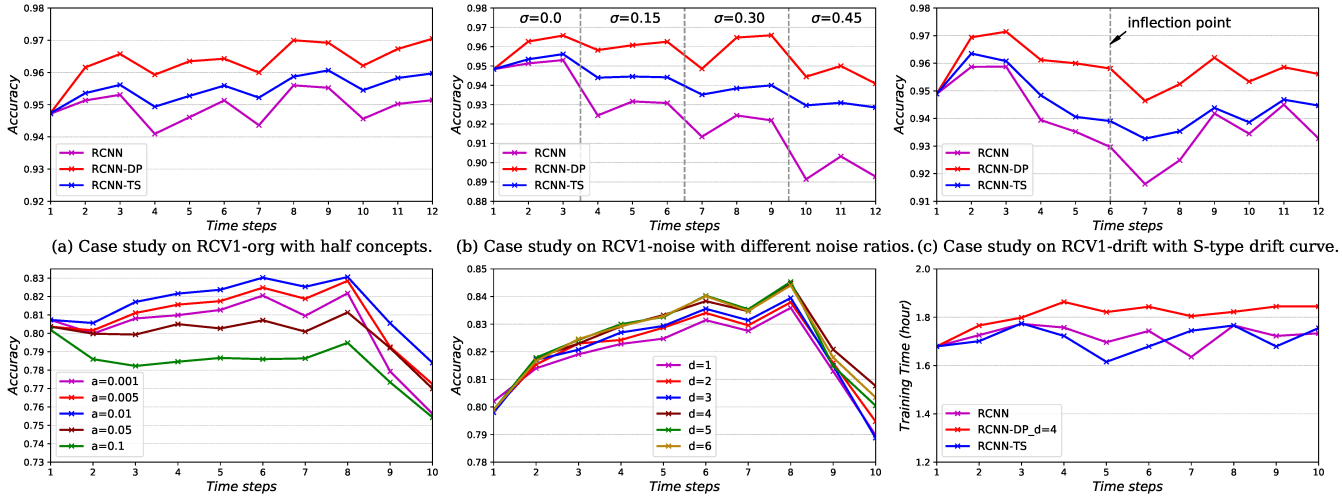
It is worth mentioning that RCNN-TS does not perform as well on the RCV1-drift dataset as it does on the RCV1-noise dataset. We believe the main cause is that the temporal smoothness framework focuses on smoothing the classifiers by preventing large deviation of model parameters, thus it is more robust against the data fluctuations. However, the downside is it may also become insensitive with the data drift. In contrast, the diachronic propagation framework achieves a richer compositionality by incorporating the historical impact into currently learned features through diachronic connections, which is more beneficial to adaptively reflecting the long-term changes while maintaining robustness to noise.

Time-step	1	2	3	4	5	6	7	8	9	10	All	Std
TextCNN	80.29	79.62	80.15	80.60	80.62	81.82	80.83	82.01	77.61	75.09	79.82	2.07
TextCNN-TS	–	81.16	82.01	82.46	82.67	83.62	83.32	83.76	81.75	80.00	82.30	1.16
TextCNN-DP	–	81.33	81.93	82.75	82.91	**84.27**	83.33	84.39	81.03	79.22	82.35	1.56
RCNN	79.72	79.78	80.20	80.82	80.91	81.58	80.60	81.32	76.94	74.65	79.64	2.19
RCNN-TS	–	80.56	81.71	82.16	82.37	83.02	82.53	83.06	80.55	78.40	81.59	1.43
RCNN-DP	–	**81.77**	**82.30**	**82.92**	**83.33**	83.83	**83.47**	**84.47**	**82.08**	**80.76**	**82.77**	1.08
HAN	68.09	67.90	68.06	67.97	67.58	67.50	66.86	68.26	66.18	64.97	67.25	1.01
HAN-TS	–	68.04	68.40	68.30	67.96	67.93	67.64	68.45	67.36	66.36	67.83	**0.62**
HAN-DP	–	68.49	69.01	68.60	68.76	68.50	68.14	69.13	67.43	66.24	68.25	0.86

Table 3: Experimental results (%) on NYTimes dataset with 10 time steps from 1997-2006 (each year as a step).

Time-step	1	2	3	4	5	6	7	8	9	10	11	12	All	Std
TextCNN	92.60	93.46	93.02	91.44	92.39	92.93	91.73	93.00	93.61	91.89	93.62	92.61	92.70	0.72
TextCNN-TS	–	93.87	93.58	92.99	93.49	93.90	93.52	94.09	94.40	93.83	94.52	94.30	93.86	0.43
TextCNN-DP	–	94.27	93.98	93.35	94.15	94.64	94.01	94.71	95.36	94.37	94.80	94.50	94.38	0.50
RCNN	93.33	94.06	93.50	92.26	93.02	93.49	92.36	94.01	94.52	92.44	94.09	93.18	93.36	0.74
RCNN-TS	–	94.50	94.20	93.60	94.10	94.23	93.82	94.38	94.80	93.95	94.40	94.06	94.19	**0.32**
RCNN-DP	–	**95.60**	**95.58**	**95.20**	**95.61**	**95.94**	**95.50**	**96.11**	**96.35**	**95.50**	**95.99**	**95.98**	**95.76**	**0.32**
HAN	82.06	84.62	83.60	79.74	83.13	82.48	82.20	85.62	86.80	81.95	84.85	84.40	83.58	1.87
HAN-TS	–	84.89	84.25	82.29	83.78	83.48	83.20	85.15	86.07	84.39	85.51	85.45	84.41	1.09
HAN-DP	–	85.42	84.90	82.99	84.81	84.60	84.63	86.63	87.63	85.03	86.72	86.63	85.45	1.26

Table 4: Experimental results (%) on RCV1 dataset with 12 time steps from 1996.09-1997.08 (each month as a step).



(a) Case study on RCV1-org with half concepts. (b) Case study on RCV1-noise with different noise ratios. (c) Case study on RCV1-drift with S-type drift curve.

(d) Parameter sensitivity of RCNN-TS on NYTimes. (e) Parameter sensitivity of RCNN-DP on NYTimes. (f) Training time of RCNN models on NYTimes.

Figure 2: Case study, parameter sensitivity, and time consumption.

4.5 Parameter Sensitivity and Time Consumption

We illustrate the parameter sensitivity and time consumption of the proposed frameworks on the NYTimes dataset.

Figure 2-(d) shows the experiment results of RCNN-TS with different initial decay factors to weight historical distance penalty. As we observe, the accuracy reaches its optimum value when $a = 0.01$, while it becomes lower with a larger or smaller value of a. Figure 2-(e) examines the effects of increasing the depth of diachronic connections to the RCNN-DP model. We find that the accuracy improves as the depth increases, and the improvement becomes negligible when the depth exceeds four. Figure 2-(f) reports the training time based on $K80$ GPUs. Compared with the standard RCNN, the temporal smoothness framework does not bring additional time consumption and the diachronic propagation framework can achieve a great performance with only a minimal increase in the computational time.

5 Conclusion and Future Work

In this paper, we mainly focus on the evolutionary supervised learning problem, and present two evolutionary neural network frameworks for time-evolving text classification. We conduct extensive experiments on two real-world news datasets and three synthetic datasets. Empirical results demonstrate that our approaches consistently and significantly outperform standard neural networks on all datasets. We believe evolutionary neural networks are very beneficial to adaptively learn the long-term changes and interactions of data in non-stationary scenarios, and can greatly improve the current start-of-the-art performances in evolutionary learning tasks, such as news topic classification, event classification and tracing, etc. In the future, we aim to explore more about the dynamic nature of data evolution like what is the convergence of long-term data evolution, how to illustrate and visualize the trend and feature of evolution, etc.

Acknowledgments

The corresponding author is Jianxin Li. This work is supported by China 973 Fundamental R&D Program (No.2014CB340300) and NSFC program (No.61472022, 61421003). The co-author Yangqiu Song is supported by China 973 Fundamental R&D Program (No.2014CB340304) and the Early Career Scheme (ECS, No.26206717) from Research Grants Council in Hong Kong.

References

[Allan *et al.*, 1998] James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

[Atefeh and Khreich, 2015] Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

[Chakrabarti *et al.*, 2006] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *KDD*, pages 554–560, 2006.

[Chi *et al.*, 2007] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162, 2007.

[Crammer *et al.*, 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[Gama *et al.*, 2014] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44, 2014.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[Ghazikhani *et al.*, 2014] Adel Ghazikhani, Reza Monsefi, and Hadi Sadoghi Yazdi. Online neural network model for non-stationary and imbalanced data stream classification. *International Journal of Machine Learning and Cybernetics*, 5(1):51–62, 2014.

[Graves and Jaitly, 2014] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *ICML*, pages 1764–1772, 2014.

[Graves *et al.*, 2013] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, pages 6645–6649, 2013.

[Jia *et al.*, 2009] Yangqing Jia, Shuicheng Yan, Changshui Zhang, et al. Semi-supervised classification on evolutionary data. In *IJCAI*, pages 1083–1088, 2009.

[Kim and Hovy, 2006] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, pages 1–8. Association for Computational Linguistics, 2006.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, 2014.

[Lai *et al.*, 2015] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *AAAI*, pages 2267–2273, 2015.

[Lewis *et al.*, 2004] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[Liu *et al.*, 2017] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In *SIGIR*, pages 115–124, 2017.

[Masnadi-Shirazi and Vasconcelos, 2010] Hamed Masnadi-Shirazi and Nuno Vasconcelos. Risk minimization, probability elicitation, and cost-sensitive svms. In *ICML*, pages 759–766, 2010.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[Nair and Hinton, 2010] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[Ning *et al.*, 2007] Huazhong Ning, Wei Xu, Yun Chi, Yihong Gong, and Thomas Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. In *SDM*, pages 261–272, 2007.

[Sandhaus, 2008] Evan Sandhaus. The new york times annotated corpus ldc2008t19. In *Linguistic Data Consortium*. 2008.

[Tang *et al.*, 2008] Lei Tang, Huan Liu, Jianping Zhang, and Zohreh Nazeri. Community evolution in dynamic multi-mode networks. In *KDD*, pages 677–685, 2008.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432, 2015.

[Wang *et al.*, 2012] Lijun Wang, Manjeet Rege, Ming Dong, and Yongsheng Ding. Low-rank kernel matrix factorization for large-scale evolutionary clustering. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1036–1050, 2012.

[Xu *et al.*, 2014] Kevin S Xu, Mark Kliger, and Alfred O Hero Iii. Adaptive evolutionary clustering. *Data Mining and Knowledge Discovery*, 28(2):304–336, 2014.

[Yang *et al.*, 1998] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and on-line event detection. In *SIGIR*, pages 28–36, 1998.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL–HLT*, pages 1480–1489, 2016.

[Zhang *et al.*, 2009] Jianwen Zhang, Yangqiu Song, Gang Chen, and Changshui Zhang. On-line evolutionary exponential family mixture. In *IJCAI*, pages 1610–1615, 2009.