Abstract—Deep learning approaches now provide state-of-the-art performance in many computer vision tasks such as handwriting recognition. However, the huge number of parameters of these models require big annotated training datasets which are difficult to obtain. Training neural networks with unlabeled data is one of the key problems to achieve significant progress in deep learning. In this article, we explore a new semi-supervised training strategy to train long-short term memory (LSTM) recurrent neural networks for isolated handwritten words recognition. The idea of our self-training strategy relies on the iteration of training Bidirectional LSTM recurrent neural network (BLSTM) using both labeled and unlabeled data. At each iteration the current trained network labels the unlabeled data and submit them to a very efficient "lexicon verification" rule. Verified unlabeled data are added to the labeled dataset at the end of each iteration. This verification stage has very low sensitivity to the lexicon size, and a full word coverage of the dataset is not necessary to make the semi-supervised method efficient. The strategy enables self-training with a single BLSTM and show promising results on the Rimes dataset.

I. INTRODUCTION

Deep learning is a growing technology [1], which can solve complex problems and has been successfully applied in various research fields such as computer vision, pattern recognition, human interaction or biology. However, deep learning requires huge datasets for training the models, while annotated datasets are generally limited in size since the annotation task is complex and costly. Training deep neural networks on unlabeled data is still an open problem, and has been identified as the "next frontier in artificial intelligence" by Y. Lecun1. In this paper we address the problem of training Long Short Term Memory (LSTM) recurrent neural networks with unlabeled data for handwriting recognition. LSTM networks are particular deep learning models dedicated to sequence modeling, and are currently at the state of the art [2] for handwritten words recognition thanks to their reliable character recognition performance. Based on this observation, we investigate how such networks can be trained in a semi-supervised learning scheme, which could then open some new perspectives for training LSTM on very large unlabeled or partially labeled datasets.

Semi-supervised learning [3] is a compromise between supervised and unsupervised learning, that operates using a small labeled dataset and a larger dataset of unlabeled samples. There are only few articles [4]–[6] addressing semi-supervised learning for handwriting recognition. This literature review shows that semi-supervised learning of LSTM recurrent neural networks has been studied using multiple classifiers that are trained and selected for labeling the unlabeled samples using some validation rules based on confidence scores. These approaches require both multiple classifiers and parameters to retrain, which can be a drawback.

In this work we explore an alternative semi-supervised training strategy to train long short term memory recurrent neural networks. This strategy is based solely on a simple lexicon verification stage, and requires only one LSTM network model. Lexicon verification is used as a labeling validation rule to label unlabeled samples, which are added to the labeled dataset, for retraining the LSTM network during the next iteration of the training procedure. Being computationally efficient and at the same time reliable, this semi-supervised learning strategy shows a very low sensitivity to the lexicon size and the lexicon word coverage. This retraining achieves interesting results compared to a standard supervised training strategy. The LSTM network character error rate (CER) is reduced from 20.34% to 14.30%.

This article is organized as follows: section II provides a brief overview of the literature concerning semi-supervised learning applied to handwriting recognition. Section III describes our proposition, and section IV presents the results.

II. SEMI-SUPERVISED LEARNING FOR HANDWRITING RECOGNITION

A. Handwriting recognition

Handwriting recognition is the process of transcribing images of handwritten words into a string of characters. Handwriting recognition is mainly based on two steps: first the optical character recognition then the linguistic processing to correct errors. Optical character models can be classified according to the character segmentation approach which can be either explicit or implicit [7]. The optical character recognition has motivated many surveys and books [8], [9]. Optical models with explicit character segmentation have been the state of the art during a long period of time, however recently a great progress has been made with deep learning,
and especially with long short term memory (LSTM) recurrent neural networks. The LSTM recurrent neural networks enable to recognize characters without any explicit segmentation process and achieves state of the art since 2009 [2], [10]–[12]. This impressive performance of LSTM networks raise the question whether they can be self trained by exploiting only their outputs to label unlabeled data. We present briefly their principles in the next section.

B. Recurrent Neural Networks

Standards recurrent neural networks (RNN) have the ability to process sequences and bring the context of the previous frames when processing the current frame. However recurrent neural networks can not learn long term dependencies, due to the vanishing gradient problem. To overcome this problem, Long Short Term Memory (LSTM) cells have been introduced by Hochreiter et al. [13]. In LSTM cells long term memory dependencies can be learned thanks to the introduction of a memory cell controlled by the input, forget and output gates.

Bidirectionality has been introduced in RNN [14] to process sequences in both direction and then they have been extended to LSTM networks [15] creating bidirectional LSTM recurrent neural networks : BLSTM. However LSTM networks only became popular with the introduction of the Connectionist Temporal Classification (CTC) [16]. This loss function is a variant of the Forward Backward algorithm that enables to train LSTM networks by inferring the ground truth at the frame level from the word transcription level. The insertion of the "joker" class between characters, has shown to improve the character recognition performance.

A typical LSTM network output has a particular profile that exhibits peaks, as the classes posterior probabilities are always nearly 0 or 1, as one can observe on Figure 1. This output profile allows to apply "Best path decoding" [17], which is a simple decoding. Best path decoding is simply the maximum class’s posterior probability at each frame and successive repetitions of classes are removed.

By achieving state of the art results, nowadays LSTM networks are the reference model for optical character recognition. This makes us wonder if a LSTM network can be further trained by itself in a semi-supervised learning and enhanced to further improves its performances.

C. Semi-supervised learning

Semi-supervised learning has been a topic for a long time [18] as the interest of using unlabeled data has always interested researchers. The aim of semi-supervised learning is to improve a classifier performance by training it on both labeled and unlabeled data. Among the many semi-supervised methods [3], co-training and self-training emerge. Co-training has been first introduced in 1998 [19], and is based on the training of two classifiers. Unlabeled data recognized with a high confidence are added to the training set of the other classifier, and vice versa. Self-training is also an old approach [18] which consists in first training a classifier with labeled data, and then retrain it multiple times using its own predictions on unlabeled data.

To our knowledge, only very few papers addressed handwriting recognition with semi-supervised learning with both co-training [5] and self-training [4], [6], and only in [4], [5] the authors retrain LSTM recurrent neural networks. We restrained this study to self-training as we only want to train a single network for obvious reasons of classifier design cost, and computational cost. In self-training, the retraining rule is important as it must provide a sufficient amount of auto-labeled data to make the training dataset grow over iterations, without introducing too much label errors at the same time. Finding an efficient retraining rule is difficult. In [4], the authors use 10 BLSTM networks to estimate a confidence score used as the retraining rule. This retraining rule relies on a threshold of the confidence score that must be determined on a labeled validation dataset. Moreover this strategy requires training multiple classifiers which perverts the original concept of self-training.

In this paper we propose a new self-training strategy that relies on an alternative retraining rule based on lexicon verification. This strategy can operate using a single LSTM network, and is free of any parameters.

III. LEXICON VERIFICATION BASED SEMI-SUPERVISED LEARNING

As shown above, self-training can be tedious and may require some parameter optimization. It may also require many different classifiers, with different features set, to be trained. In this paper we take a different approach, where the self-training strategy only requires a lexicon to validate the labels predicted on unlabeled data.

A. Lexicon verification

Lexicon verification has been proposed in a previous paper [20] so as to replace lexicon-driven decoding. It simply considers the sequence of characters predicted by the LSTM networks, and applies the following simple rule: "accept the
sequence hypothesis if it belongs to the lexicon, and reject it otherwise”. This rule obviously provides a rather high reject rate, but has the big advantage of generating a very low error rate, since it is very unlikely that an erroneous predicted sequence belongs to the lexicon. This probability \( P_{\text{wrong}} \) can be computed for a word of length \( n \) characters thanks to equation 1, where \( CER \) is the Character Error Rate of the classifier, \( m_n \) is the number of words of length \( n \) in the lexicon and \( d \) is number of character labels. This probability is very low for word length of more than 2 characters.

\[
P_{\text{wrong}} = (1 - (1 - CER)^n) \times \frac{m_n}{d^n} \tag{1}
\]

Besides its small error rate, lexicon verification can be easily implemented with a hash table (e.g. dictionary in python) achieving nearly instant result for lexicon of gigantic size [2]. Therefore, lexicon verification has strong advantages that can benefit to semi-supervised learning. These capacities combined with the high performance LSTM network let us think that it can be used as a self-training rule.

B. Self-training rule

The self-training rule is defined by adding self labeled samples (previously unlabeled) to the labeled dataset in order to iteratively perform supervised training on a larger labeled dataset. In this work, we propose to benefit from the reliability of the lexicon verification strategy and use it as a retraining rule. At each iteration, the current LSTM network process the unlabeled dataset, and its predictions are submitted to the lexicon verification stage. A LSTM network is first trained using a small labeled dataset, then this network will initialize self-training iterations with its own results (see Fig. 2). This self-training strategy is very simple to use and is free of any parameter to tune. It only requires a lexicon and some labeled data to be initialized. Let us emphasize that any lexicon can be used: a small lexicon with a poor coverage of the unlabeled dataset will result in a small increase of the number of unlabeled data at each iteration (with a very small error rate), while a very large lexicon will result in a fast labeled dataset growth (at the expense of a possible higher level of erroneous labeled words in the labeled dataset).

In this article, an alternative retraining rule has been experimented, which is more related to co-training strategy: two networks are trained on the labeled dataset, and both process the unlabeled dataset. This co-training strategy adopts the following retraining rule: unlabeled data are added to the labeled dataset if the results hypotheses are identical for both networks, we call this rule “agreement”. This “agreement” retraining rule is only a way to compare and is not part of our claims.

We now describe the LSTM networks used for evaluating our semi-supervised strategy.

C. Network architecture

To test our lexicon verification retraining strategy we use a single BLSTM network. It is composed of two layers of respectively 70 and 100 LSTM cells. The network use Histogram of Gradients as input features [21]. Histogram of Gradients have shown to be an efficient feature to feed a BLSTM in [22]. The features are extracted on a sliding widow of width 8 pixels with a step of 1 pixel.

As evoked earlier we have another strategy for comparison purposes that requires a second classifier. Therefore for this strategy we also trained a MDLSTM. The MDLSTM network is composed of 3 layers of 2, 10 and 40 neurons respectively, and is directly applied on the pixels of the image.

For both networks, training is performed with RNNLIB [23]. We uses the standard stochastic gradient descent with a fixed learn rate of \( 10^{-4} \) and a momentum of 0.9.

IV. EXPERIMENTATION AND RESULTS

We now describe our experimentation setup and the results obtained using our new semi-supervised training strategy. This section details how we use the Rimes dataset, and the results we obtain for this given network.

A. Dataset

For the evaluation of our approach we use the ICDAR 2011 Rimes dataset of isolated words. We kept the same split between training, validation and test as it is provided. The dataset is composed of 51 738 training images, 7464 validation images and 7716 test images. The lexicon of the Rimes dataset contains every words from the whole dataset (validation and test included) that is to say 5744 words. In order to evaluate the influence of the lexicon size, we have created a large lexicon based on the union of the Rimes lexicon and of the French dictionary Gutemberg, which contains 342 275 words. In order to demonstrate the insensitivity of our approach to very large lexicons, we have even made a lexicon of more than 3 millions words extracted from the French Wikipedia and Wiktionary pages and the French dictionary Gutemberg. This latest gigantic lexicon has a low word coverage of the Rimes dataset even with its huge size. This article presents results for these 3 lexicons, referenced by their size 5k, 340k and 3M prefixed with ”verification” in tables and figures.
We randomly sampled 10,000 images\(^2\) from the training set as the initialization images of semi-supervised learning. The other images are considered unlabeled and used at every iteration of the self-training process. With the dataset settled, we can train and iterate with our lexicon verification strategy. We present the results in two parts, the first part presents the results and compares to other methods, the second is an analysis of the self-labeled data.

\section*{B. Results}

We use for our results multiple metrics:

- Character error rate (CER) on the test dataset (edit distance);
- Accuracy is the character recognition, that is to say 100$-$CER;
- Word error rate (WER) on the test dataset: proportion of mis-recognized words\(^3\);
- Improvement: relative accuracy progress between classifier trained only on labeled data, and trained on labeled and unlabeled with our semi-supervised strategy;
- $\%$ to supervised: relative percentage separating classifier trained on labeled and unlabeled with our semi-supervised strategy, classifier trained on the whole labeled dataset (51,738 training images).

We report in table I CER, improvement and $\%$ to supervised for the verification strategy, for the best iteration. Supervised full is the performance of our network on the whole (51k words) training dataset. We can observe that our semi-supervised strategy performs well, and significant improvement is observed for every lexicons with respect to a supervised training on 10K words. As also shown on figure 3, our strategy works very well for the 3 lexicons, and as a comparison the simple agreement strategy works significantly less.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
Strategy & CER & Improve-ment($\%$) & $\%$ to su-
\multicolumn{1}{c|}{\text{ervised}} \\
\hline
Supervised full & 11.45 & - & - \\
Supervised 10k & 20.34 & - & - \\
Verification 5k (ite5) & 14.20 & 7.6 & 3.2 \\
Verification 340k (ite6) & 14.96 & 6.7 & 3.9 \\
Verification 3M (ite5) & 15.89 & 3.6 & 5.0 \\
\hline
\end{tabular}
\caption{Comparison of the character error rates obtained with: a supervised training on the fully labeled dataset, a supervised training on only 10K words, and our semi-supervised strategy with 3 different lexicons.}
\end{table}

In table II, WER results are presented with a simple Viterbi\(^4\) decoding to compare our method to others with the same lexicon as verification that is to say the 5k words lexicon. The semi-supervised WER for verification approach also improves comparing to iteration 0 and is not very far from the upper limit of fully supervised learning. For comparison purpose, if we were competing in handwriting recognition ICDAR 2011 competition, we would rank 3rd just behind the second and greatly a head of third place, while only a fifth of the whole training set is labeled.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Strategy & WER \\
\hline
Supervised full & 11.10 \\
Supervised 10k & 21.90 \\
Verification 5k (ite5) & 15.41 \\
ICDAR 2011 second place \cite{11} & 12.53 \\
\hline
\end{tabular}
\caption{Comparison of our strategy WER between iteration 0 and the upper limit of fully supervised.}
\end{table}

In table III we compare our semi-supervised method to both approaches proposed in \cite{4}, \cite{5}. As no network architecture nor dataset sampling was given in \cite{4}, \cite{5}, the exact comparison on the same data was impossible to conduct. However, we report our improvement and $\%$ to supervised criterion for all the approaches in table III. It appears that for both criteria we perform better than compared methods.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
System & Improvement($\%$) & $\%$ to supervised \\
\hline
Our work \cite{24} & 7.6 & 3.2 \\
Frinken et al \cite{4} & 3.7 & 3.6* \\
Frinken et al \cite{5} & 7.24 & - \\
\hline
\end{tabular}
\caption{Comparison of the relative improvement to CER of our method to other state of the art methods.}
\end{table}

This interesting results show that our strategy, without any parameters to tune, is efficient and allows to further train LSTM RNN. In addition we analyze self labeled data as it is an important factor.

\section*{C. Self labeled images analysis}

Self labeled images are an important part of the success of our retraining rule. Indeed there must be a sufficient amount

\(^2\)Random sampling available on request.

\(^3\)In a case-insensitive way, as it is usually done on the Rimes dataset.

\(^4\)Result computed from the curves given in the article.
of new data injected in the training iterations, while keeping
an error rate as low as possible. Lexicon verification allow us
to have both.

Pie charts of percentages of the length of words are pictured
on figure 4. Figure 4a has a wide variety of length, words
of length 4 and less account for around 51% of the dataset
whereas at iteration 0 (figure 4b) self-labeled words are
at 73% words of length 4 and less. This shows that our
lexicon verification rule first validates short words that are
less complex to train on. Where our strategy shines is that
after some iterations, more complex words are introduced as
only 61% are short words, as shown on figure 4c. These pie
charts show that lexicon verification allows to gradually feed
more complex examples and then improves performances.

As pictured in figure 5, the percentage of self-labeled ex-
ample is increasing with iterations, providing new informa-
tions for the network. On another side in figure 6, the CER of
the self-labeled images is almost stable, unless for the gigantic
lexicon for which there is a clear decrease. However CER
is also very low for the agreement and lexicon of size 5k
and 340k, whereas it is higher for the lexicons of more than
3 millions words. Quality of self-labeled data is constant or
improved, and quantity is always increasing. Both figures show
how our retraining rule influence the self-labeled dataset over
iterations. There is still room for improvements, especially for the gigantic lexicon, but the percentage of quantity of new data is getting significantly lower and better results are not expected to overcome a progress of 1% in accuracy.

We have demonstrated that our retraining rule with lexicon verification both performs very well and ensures data quality and quantity.

V. CONCLUSION

In this work, we exploit the idea that unlabeled data can be used for training a recognizer. There are many semi-supervised techniques, but few for handwriting recognitions. Current state of the art methods are often complex and involved many classifiers. In this article, we have proposed a new retraining strategy for self training. This strategy is based on lexicon verification and only requires a single network to retrain and a lexicon. Our method doesn’t depend on the lexicon size and is parameter free. We achieve very good results and our improvement with the retraining is higher to other methods. One way to improve our strategy could be to mix agreements between several networks and lexicon verification in order to improve both data quantity and quality.

REFERENCES