Online Learning for Solving Data Availability Problem in Natural Language Processing*

B.W. Kothalawala[1 [0000−0003−3023−2905]], A.R. Weerasinghe[2,3 [0000−0002−1392−7791]], and K.V.D.J.P. Kumarasinghe[3]

University of Colombo School of Computing info@ucsc.cmb.ac.lk
https://ucsc.cmb.ac.lk

Abstract. In Machine Learning(ML) in general, more data is better data. Besides, more data defeats better algorithms for performance improvement in most cases. In practice, access to such data may not be directly forthcoming but become available with time. Existing ML solutions for Sinhala Named Entity Recognition (NER) and Part of Speech (POS) tagging, typically retrain their models from scratch when new data becomes available. The training time required for this purpose increases proportionally to the overall dataset size. This research proposed two online ML models that do not require retraining their models from scratch when data is obtained in batches, namely an Online Conditional Random Fields (CRF) Model, and a Bidirectional Long Short Term Memory-Conditional Random Fields (LSTM-CRF) Model. A Sinhala NER experiment using the Online CRF model improved on previous attempts by an F1-measure of 31.5% to reach 76% while the Bidirectional LSTM-CRF model improved on a previous attempt by an F1-measure of 51.6% to reach 80%. In the Sinhala POS tagging experiment using the Online CRF model improved the accuracy from approximately 71 % to 76% while the Bidirectional LSTM-CRF model improved the accuracy from 70% to 76%. The training time consumed by the proposed online learning model remains constant in each incremental training step since the model is not built from scratch. Further, both the Online CRF model and the Bidirectional LSTM-CRF model achieved improvements over the state-of-the-art Sinhala POS tagging accuracy of 4%. Based on the extrapolation of these results it can be seen that the existing Sinhala NER dataset needs to approximately double in order to obtain state-of-the-art performance reported for English.

Keywords: Natural Language Processing · Online Machine Learning · Sinhala Language.

1 Introduction

Recent approaches to Named Entity Recognition (NER) and Part of Speech (POS) tagging, use Machine Learning (ML) techniques [14,11]. The size of the corpus needed for the training phase is the main influence in obtaining higher accuracies in such ML models. For languages such as English, there already exist large numbers of datasets

* University of Colombo School of Computing

to train models on. However, obtaining such large data sets at the initial stage for most languages is impractical especially low resource natural languages such as Sinhala. In practice, we have to obtain several (mini) batches of data at different time points, to make a large dataset that is capable of producing better results. In general we obtain data in 'mini-batches' $x_1, x_2, x_3, ..., x_n$ at different time steps $t_1, t_2, t_3, ..., t_n$. For this kind of phenomenon, the existing ML techniques for NER and POS tagging, initially train the model using x_1 mini-batch. After obtaining the x_2 mini-batch, we aggregate x_2 and x_1 and then train the model using the aggregated data set $x_1 + x_2$. Similarly, when x_3 becomes available, the model trains using $x_1 + x_2 + x_3$.

This process indicates that the batch learning techniques retrain the ML models using the same dataset multiple times. This retraining process results in a higher training time. The main motivation to carry out this research is to avoid this overhead of retraining, without losing the accuracy of the model. The Natural Language Processing (NLP) models should need the capability to understand the present context of the natural language. The proposed ML models should also adapt to the current natural language context using the most recent data. The state-of-the-art methods for Sinhala NER obtained nearly 92% precision [7]. Another objective of this research is to predict how much data is needed to enhance this accuracy values. The problem of this research causes due to the characteristics of the batch learning techniques. The proposed ML models use online learning techniques because online learning can train an ML model using incrementally collected datasets.

2 Related Works

2.1 NER and POS tagging

The main approaches to NER and POS tagging can be stated as: Rule-based, Data-driven, and Hybrid. In the rule-based approach, linguistic knowledge is used to create a set of rules to identify named entities or part of speech tags. The data-driven approach contrastingly depends on three main ML methods, namely, Supervised, Semi-supervised, and Unsupervised learning. The supervised learning methods use models such as Conditional Random Fields (CRF), Maximum Entropy (MaxEnt), and Hidden Markov Model (HMM) to build models from labeled data.

CRF is an undirected graphical model and matches up with the conditionally trained probabilistic finite-state automata. CRF is capable of including arbitrary features easily because it trained conditionally. CRF model has been used over the HMM model because CRF solves the label bias problem [7]. Suppose (*X*) is a set of feature vectors corresponding to a particular data corpus and *Z* is the set of corresponding NER labels of each word in *X*. The CRF model is a graph *G(V, E)* such that the vertices(*V*) represent the NER tags($z_i \in Z$). In a CRF model, $z_i \in Z$ is the random variable and it adheres to the Markov property. The Z is conditioned on X such that $p(z_i|X, z_i, i \sim j)$ where $i \sim j$ means *i* and *j* vertices are neighbors in *G* [13]. Since the CRF model provides better accuracies for the Sinhala language, we use the CRF model for our experiments in an incremental manner [7].

2.2 Online and Incremental Learning

Online learning algorithms are those which execute the training process on the data as it becomes available and not all at once. Incremental learning is an online learning strategy that works with limited memory resources and relies on the compact representation of the already observed data [8]. The key challenges related to online and incremental learning are as follow [8]:

1. *Concept Drift*: When data become available in different time steps then there exist several changes in the data distribution which relevant to the time dimension.
2. *Catastrophic Forgetting*: Online learning models keep learning as long as the data comes to the model. When it learns new information, there is a chance to forget previously learn things. The forgetting speed will determine how fast the online learning model learns new information. The process of forgetting previously information called catastrophic forgetting.
3. *Stability Plasticity Dilemma*: If an online learning model learns new information quickly, then it will forget past information immediately. On the other hand, if an online learning model decreases the leaning speed, it will drop some of the crucial information from the learning process. This challenge of handling both ends called Stability Plasticity Dilemma.

Carreras et. al. [5] introduced a new approach for NER referred to as the voted perceptron model which is based on an online perceptron strategy. The online algorithm they propose was a mistake-driven online algorithm. The execution of the online algorithm could be categorized into two phases. First, the algorithm is applied to learn at the word level to identify named entity candidates utilizing a Begin-Inside (BIO) classification. Then the algorithm makes use of functions learned at the phrase level. Finally they apply the online learning strategy at a sentence level. In our research, we are trying to apply online learning using this mistake-driven online strategy. For the English language, they obtained overall precision, recall and F-measure values of 85.81%, 82.84%, and 84.30% respectively [5].

Recurrent Neural Network (RNN) architecture has been designed to learn in an online manner [4] and for handling structured prediction tasks making it a good fit for the problem at hand. However, the RNN model performs poorly when there are long-term dependencies in the sequence prediction task. To handle these long-term dependencies, researchers enhanced the internal structure of the RNN cells into Long Short-Term Memory (LSTM) cells [15].

Athavale et. al. [1] applied a deep neural network approach for Hindi NER using different kinds of RNN layers, namely, Vanilla RNN, LSTM, and Bi-directional LSTM. From these three types, the bi-directional LSTM layered model outperforms the other two. As the final output, they obtained 90.32% accuracy for Conference on Natural Language Learning for the CoNLL-2003 dataset without using any Gazetteer information. Chiu and Nichols [6] implemented a hybrid bidirectional LSTM and a Convolutional Neural Network (CNN) architecture for their classification. The experiments they carried out used word level and character level features for NER classification. Finally, they obtained 91.62 F1 score on the CoNLL-2003 dataset. Huang et. al. [12] combined an LSTM model with a CRF model for sequence tagging tasks. The bi-directional LSTM

layer has the capability of using past and future features to make predictions, while the CRF layer has the capability of using sentence-level features. They obtained 97.55% accuracy from their bi-directional LSTM-CRF model.

3 Implementation

The implementation of the research followed the common machine learning pipeline as below:

1. Data Preprocessing
2. Apply Online Learning Algorithm
3. Post Processing

This research followed the supervised learning approaches because supervised learning methods obtained superior performance compared to other learning methods [14,11]. The vocabulary was ordered according to the word frequency.

Algorithm 1 Train and Predict

```
1:  procedure TRAIN_AND_PREDICT(data, tokens)
2:      train_data,test_data,validation_data ← SPLIT(data)
3:      train_tokens,test_tokens,validation_tokens ← SPLIT(tokens)
4:      X_1, X_2, X_3, X_4 ← CREATE_MINIBATCHES(train_data)
5:      Y_1, Y_2, Y_3, Y_4 ← CREATE_MINIBATCHES(train_tokens)
6:      ML ← INITIALIZE(params)
7:      for i in (1, 2, 3, 4) do
8:          TRAIN(ML, X_i, Y_i)
9:          VALIDATE(ML, validation_data, validation_tokens)
10:         results ← PREDICT(ML, test_data)
11:     end for
12:     ANALYZE(results, test_tokens)
13: end procedure
```

Algorithm 1, first splits the data using a SPLIT function and divides the data set into train, test, and validation sets. Subsequently, the CREATE MINIBATCHES function creates four (in this case) mini-batches to train the model using the training dataset. The ML models are then initialized. The for loop in the algorithm iterates through each mini-batch. The training, validating, and testing procedures are applied to each mini-batch in that loop. At the end of the for loop, predicted results are stored in the variable $results$. The $results$ are used to analyze the performance of the proposed ML models.

The research proposes two online ML models: an Online Conditional Random Fields (CRF) Model and a Bidirectional Long Short Term Memory-Conditional Random Fields (LSTM-CRF) Model.

The architectures of the two proposed models are shown in Figure 1. *Model A* depicts the Online CRF model while *Model B* depicts the bidirectional LSTM-CRF model with a dropout layer. Both models start with an embedding layer which converts words

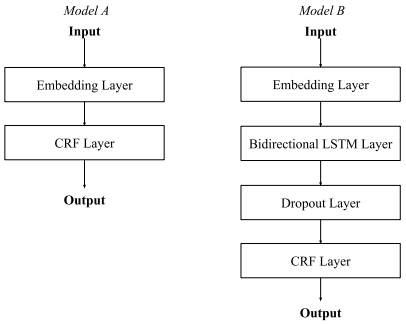

Fig. 1. Network Architectures: Online CRF Model and Bidirectional LSTM CRF Model

into vector representations. The previously unseen words will not be represented in the BoW vector model because it generates a sparse vector representation. This problem of unseen data does not occur in the embedding technique because it creates a dense vector space [3].

The proposed two models have a final CRF layer that can use sentence-level features in the predictions. *Model B* also uses a bidirectional LSTM layer which can use past and future contextual features for making predictions. The architecture of *Model B* is influenced by a previously proposed solution [12]. However, we modify their model by adding a dropout layer. The primary duty of this dropout layer was to randomly ignore several neurons of the network, during the training phase. This random dropping of neurons makes other neurons contribute to the prediction which would normally be done by the dropped neurons. Thus over specializing certain neurons to specific predictions is removed from the neural network. This allows the model to avoid overfitting [2]. The previously discussed challenges of online learning (Catastrophic Forgetting, Concept Drift, and Stability-Plasticity Dilemma) become critical if the model applied quick updating. The ML models can handle these challenges if they can regulate these quick updates. Since the dropout layer regulates these quick updations, the dropout layer can handle these challenges.

4 Experiments

Two tasks were carried out in this research: 1). Sinhala NER and 2). Sinhala POS tagging. For these experiments we used Sinhala NER and POS tagging datasets from the Language Technology Research Laboratory (LTRL) of the University of Colombo School of Computing (UCSC) under the Lesser General Public License (LGPL) for Linguistic Resources. The datasets applied to the proposed models adhere to the CoNLL-2003 format.

The research proposed two online learning models, namely, an Online CRF and a Bidirectional LSTM-CRF. We carried out two experiments for each task mentioned above. Hence the overall research consisted of four experiments using the models. For

each experiment in the research, we simulate the batch learning technique and the online learning technique to compare and contrast the performance of the learning techniques. Each experiment used four mini-batches of data. Suppose the four mini-batches are denoted as $m_1, m_2, m_3,$ and m_4. A separate dataset T is used to test the performance of the models. In each training step, the batch learning model retrains from scratch while the online learning model has been saved and used it for further training. The training phase of the batch learning approach and the online learning approach consisted of four steps. These four steps of each batch learning and online learning experiment can be described as follows:

- Batch Learning Experiment

Step 1: Initially trained using m_1.
Step 2: When m_2 mini-batch becomes available, we train the model from scratch using whole aggregated $(m_1 + m_2)$ dataset.
Step 3: When m_3 mini-batch becomes available, we train the model from scratch using whole aggregated $(m_1 + m_2 + m_3)$ dataset.
Step 4: When m_4 mini-batch becomes available, we train the model from scratch using whole aggregated $(m_1 + m_2 + m_3 + m_4)$ dataset.

- Online Learning Experiment

Step 1: Initially trained using m_1 mini-batch and test on dataset T.
Step 2: When m_2 becomes available, the model trained only using m_2.
Step 3: When m_3 becomes available, the model trained only using m_3.
Step 4: When m_4 becomes available, the model trained only using m_4.

5 Results

5.1 Online CRF Model

Sinhala NER Experiment For the Sinhala NER Experiment, Figure 2 shows the variation of precision, recall, and F1-measure of the online and batch learning techniques in each step of the experiment. The precision values of the batch learning were higher in the first three steps, but in the fourth step, the online learning method performed better. The Recall values of batch learning techniques have higher values in all the steps, except for the third step of the experiment. The F1-measure values of batch learning were higher throughout the experiment. However, the online learning F1-measure value was closely related to the batch learning value in the last mini-batch.

Note that batch learning and online learning gives distinct accuracies in the initial step of this experiment. Since we have used the same dataset for the learning process, most of the time these accuracies are close. However, we train these models separately. Each ML model (batch and online) starts with a set of random values and then are trained on the data. These random initializations may have affected in changing the initial accuracies. More importantly, even though we use the same dataset for initial training, the training procedures are quite different. The online learning algorithm performs quick updates to the ML model while the batch learning algorithm does not perform such quick updates. These updates also affect the accuracies of online and batch learning.

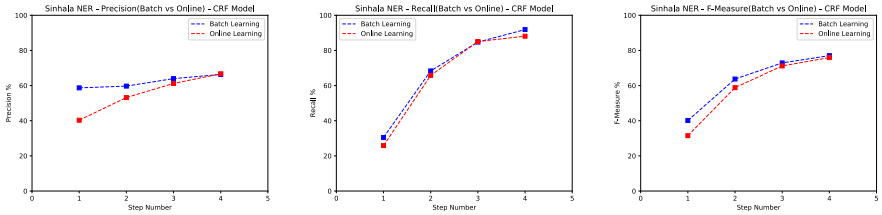

Fig. 2. Precision, Recall, and F1-Measure value variation of four steps using the CRF model - Sinhala NER

Sinhala POS Tagging Experiment We checked the applicability of the proposed Onlin CRF model for the Sinhala POS Tagging task. Figure 3 depicts the accuracy variation of the POS tagging experiment for both the batch learning and online learning approaches. The accuracy values of the online learning model were closely related to the batch learning technique. Further, in the first step of the experiment, the online learning model obtained slightly higher accuracies than the batch learning approach. However, in the other steps of the experiment, batch learning techniques demonstrate slightly higher accuracies.

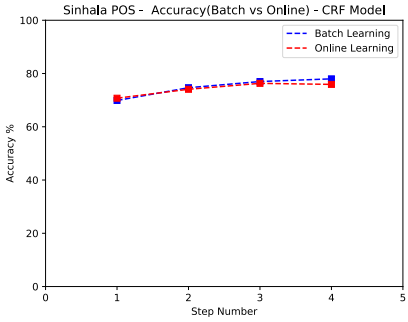

Fig. 3. figure
Accuracy value variation of four steps using the CRF model - Sinhala POS

5.2 Bidirectional LSTM CRF model

Sinhala-NER Experiment For the Sinhala-NER Experiment, Figure 4 shows the variation of precision, recall, and F1-measure of the batch learning technique with the online learning technique, for each step of the experiment. The precision values of the online learning model were closely related to the batch learning model except for the second step. When examining the recall values, the second and third steps of the batch

learning technique gave higher values and in other steps, the online learning technique closely related to the batch learning technique. Also when it comes to F1-measure values, the online learning model was closely related to the batch learning model except for the second step.

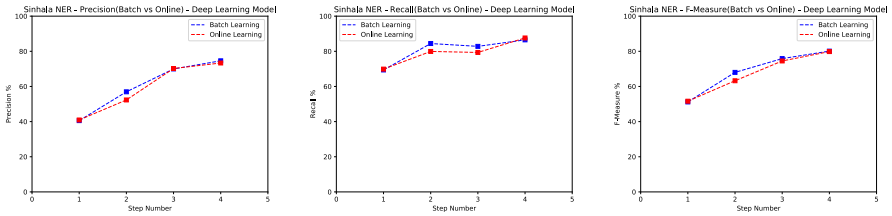

Fig. 4. Precision, Recall, and F1-Measure value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala NER

Sinhala-POS Tagging Experiment We also checked the applicability of the bidirectional LSTM-CRF model for the Sinhala-POS Tagging task. The variation in the accuracy of online and batch learning strategies are shown in Figure 5. The accuracy values of the online learning and batch learning techniques are closely related to each other as apparent from Figure 5.

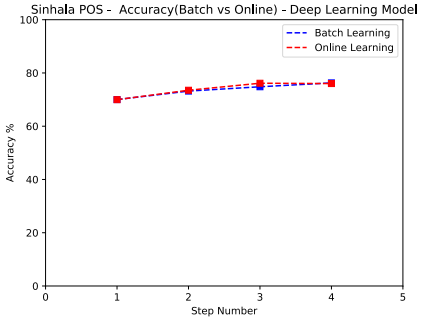

Fig. 5. Accuracy value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala POS

5.3 Training Time Comparison

As shown in Figure 6, the training time consumed by online learning models remains almost constant in each step. However, the training time of batch learning models has increased linearly in each step.

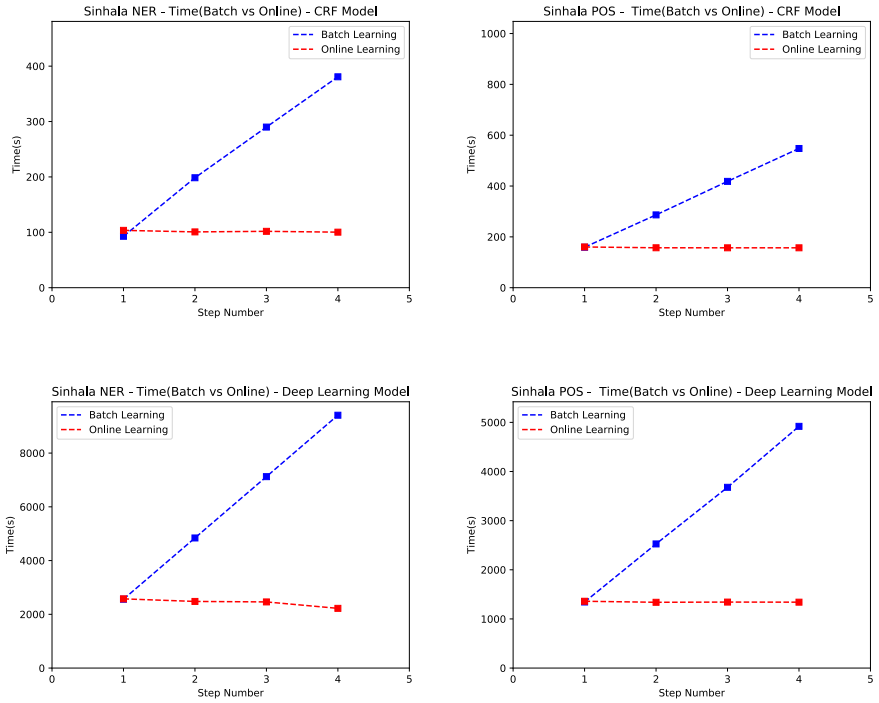

Fig. 6. Training time variation of the four experiments

Table 1. Predictions of the Sinhala NER Experiment

Experiment	Logarithmic Function	Number of Data Sources with 662 Sentences	Sentences
CRF	$f(x) = 33.3 + 32.8ln(x)$	8	5296
Deep Learning	$f(x) = 50.8 + 20.7ln(x)$	11	7282

5.4 Dataset Predictions

One of the main objectives of this research was to find out how much data would be needed for the proposed models to reach acceptable accuracy. The research analyzed

the performance variation of these experiments over a given mini-batch. For each experiment, we generate a function to approximate the performance variation. That approximated function is then used to predict the size of data that the proposed models require to obtain acceptable performance. For the NER experiment, we have considered the F1-measure variation since it is the combination of precision and recall. Since the POS tagging experiment has more than 22 labels to be predicted, it is convenient to use accuracy as the evaluation metric instead of precision, recall, and F1-measure. Hence, the accuracy variation is used for the dataset size prediction in the Sinhala POS tagging experiment.

According to Figure 2 and 4, we observed that the performance variation of NER follows a logarithmic scale. Hence, the F1-measure variation of Sinhala NER was approximated using a logarithmic regression. The results of the NER prediction are shown in Table 1. The first column of Table 1 contains the experiment name. The second column contains the approximated function using logarithmic regression. These logarithmic functions are defined in the range [0-100) exclusively. The independent variable of these functions is the number of steps in the experiment. Each step in the Sinhala NER experiment is trained using a separate data source (mini-batch) of 662 sentences. Thus the third column contains the number of data sources (the data source size is 662 sentences) that is needed to obtain near optimal performance. The final column shows how many sentences are needed to obtain such performance based on the fitted model.

We currently have a Sinhala NER dataset consisting of 3268 sentences. To obtain acceptable performance from the CRF model, the Sinhala NER data set needs to be at least 5296 sentences long, while the bidirectional LSTM-CRF model requires a total of 7282 sentences. Both experiments point to the need to approximately double the Sinhala NER dataset in order to obtain acceptable performance.

Sinhala POS tagging did not have a large enough dataset in the training phase, to observe the incremental improvement of accuracy in the POS tagging experiments. Hence, the dataset size prediction for this task were not possible to calculate.

6 Conclusion

We proposed two online learning algorithms: an Online Conditional Random Fields (CRF) and a Bidirectional Long Short Term Memory-Conditional Random Fields (LSTM-CRF). Both models can be used across various NLP tasks such as NER and POS tagging. Since the models do not use any language-dependent features, these models can also be used across any natural language. The proposed models increase the performance in each incremental training step. The training time consumed by the proposed models remains constant over each incremental training step. The dataset size needed to reach acceptable performance for Bidirectional LSTM-CRF models is higher compared to that needed for the Online CRF model. However, the performance of the Bidirectional LSTM-CRF model is higher than the Online CRF model. The training time of the Online CRF model is low compared to the deep learning model. The analysis of four experiments showed that online learning techniques can reach batch learning performances. The training time for online learning methods remains nearly constant in each training step. However, the training time for batch learning increases linearly. The inclu-

sion of the dropout layer for the proposed online learning model solved the stated key challenges (Catastrophic Forgetting, Concept Drift, and Stability-Plasticity Dilemma) of online learning. Most importantly, the dropout layer gives consistent growth to the online ML model.

The state-of-the-art Sinhala POS tagging experiment by Gunasekara et. al. [10] obtained 72% accuracy from their hybrid approach. The final accuracies of our Online CRF model have improved the state-of-the-art accuracy by nearly **4%**. The Bidirectional LSTM-CRF model has also improved the state-of-the-art accuracy by **4** percentage points. The dataset needed for Sinhala NER to perform at acceptable levels is approximately double the current size. The Sinhala POS tagging task requires a large heterogeneous dataset to learn new information from each incremental training step and is currently too small to be used to estimate from.

7 Future Work

Integrating the online learning based NER and POS tagging models with other NLP tasks can be implemented as future work. The actual usage of these online learning techniques become more worthwhile after converting the major NLP tasks such as Information Extraction, Machine Translation, Automatic Summarization, and Information Retrieval into the online learning strategy.

References

1. Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., Shrivastava, M.: Towards deep learning in hindi NER: an approach to tackle the labelled data sparsity. CoRR **abs/1610.09756** (2016), http://arxiv.org/abs/1610.09756
2. Brownlee, J.: Dropout regularization in deep learning models with keras, machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/
3. Brownlee, J.: How to use word embedding layers for deep learning with keras, machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
4. Brownlee, J.: Instability of online learning for stateful lstm for time series forecasting, machinelearningmastery.com/instability-online-learning-stateful-lstm-time-series-forecasting/
5. Carreras, X., Màrquez, L., Padró, L.: Learning a perceptron-based named entity chunker via online recognition feedback. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4. pp. 156–159. CONLL '03, Association for Computational Linguistics, Stroudsburg, PA, USA (2003). https://doi.org/10.3115/1119176.1119198, https://doi.org/10.3115/1119176.1119198
6. Chiu, J.P.C., Nichols, E.: Named entity recognition with bidirectional lstm-cnns (2015), http://arxiv.org/abs/1511.08308, cite arxiv:1511.08308
7. Dahanayaka, J.K., Weerasinghe, A.R.: Named entity recognition for sinhala language. In: 2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer). pp. 215–220 (Dec 2014). https://doi.org/10.1109/ICTER.2014.7083904
8. Gepperth, A., Hammer, B.: Incremental learning algorithms and applications (2016)

9. Gimpel, K., Das, D., Smith, N.A.: Distributed asynchronous online learning for natural language processing. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning. pp. 213–222. CoNLL '10, Association for Computational Linguistics, Stroudsburg, PA, USA (2010), http://dl.acm.org/citation.cfm?id=1870568.1870593
10. Gunasekara, D., Welgama, W.V., Weerasinghe, A.R.: Hybrid part of speech tagger for sinhala language. In: 2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer). pp. 41–48 (Sept 2016). https://doi.org/10.1109/ICTER.2016.7829897
11. H.Shah, P.Bhandari, K.Mistry, S.Thakor, M.Patel, K.Ahir: Study of named entity recognition for indian languages. International Journal of Information Sciences and Techniques (IJIST) (2016)
12. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. CoRR **abs/1508.01991** (2015), http://dblp.uni-trier.de/db/journals/corr/corr1508.html#HuangXY15
13. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001), http://dl.acm.org/citation.cfm?id=645530.655813
14. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Linguisticae Investigationes **30**(1), 3–26 (January 2007), www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002, publisher: John Benjamins Publishing Company
15. Olah, C.: Understanding lstm networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/