Online Learning for Solving Data Availability Problem in Natural Language Processing

B.W. Kothalawala

Online Learning for Solving Data Availability Problem in Natural Language Processing

B.W. Kothalawala

Index No.: 14000679

Supervised by

Dr. A.R. Weerasinghe
Mr. K.V.D.J.P. Kumarasinghe

January 2019

Submitted in partial fulfillment of the requirements of the B.Sc. in Computer Science (Hons) Final Year Project in Computer Science (SCS4124)

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: B.W. Kothalawala

....

Signature of Candidate Date: January 8, 2019

This is to certify that this dissertation is based on the work of Mr. B.W. Kothalawala under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name: Dr. A.R. Weerasinghe

....

Signature of Supervisor Date: January 8, 2019

Co-Supervisor Name: Mr. K.V.D.J.P. Kumarasinghe

....

Signature of Co-Supervisor Date: January 8, 2019

Abstract

Named Entity Recognition (NER) and Part of Speech (POS) tagging are major prerequisites for various NLP applications. The state-of-the-art performance of NER and POS tagging is obtained using statistical and machine learning (ML) techniques. Machine learning models require a large data corpus to achieve better performance. However, obtaining a large data resource at once is often not practical. In practice data is aquired incrementally, often as a sequence of mini-batches. CRF, HMM, and MaxEnt models are the state-of-the-art ML models for NER and POS tagging. Since these models based on batch learning techniques, these models need to be retrained from scratch each time new data is added to the corpus.

This study proposes to solve this problem using online machine learning techniques because it can deal with incrementally collected data sets. Two online learning models are constructed in this study: 1). Online Conditional Random Fields (CRF) and 2). Bidirectional Long Short Term Memory-Conditional Random Field(LSTM-CRF).

The Sinhala NER experiment using the proposed Online CRF model achieved an F-measure value improvement from 31.4914% to 75.9259%. The F-measure value for Sinhala NER using the proposed Bidirectional LSTM-CRF model increased from 51.6180% to 79.8365%. Sinhala POS tagging using the proposed Online CRF model increased the accuracy from 70.7307% to 75.9147%. The Bidirectional LSTM-CRF model for Sinhala POS tagging increased the accuracy from 69.9681% to 76.0177%. The online learning models performance was able to match the batch learning models while retaining the flexibility of incremental model building. The training time of online learning remains nearly constant over the sequence of mini-batches while the training time for batch learning methods gave linearly increasing training time.

Preface

The existing machine learning (ML) models that applied for Sinhala NLP task have to retrain from scratch when we obtain datasets as a stream. Also, NLP models should have the capability of adapting to the current context of the natural language. Here, we try to solve these problems by using online learning methods. We propose two online learning models to solve these problems: 1). Online CRF and 2). Bidirectional LSTM-CRF.

The architectures of the proposed online machine learning models influenced by the related works which described in chapter 2. The modifications added to those models, to adapt for online learning task are my own investigations. The design of the experiments in an incremental manner and applying those machine learning models into the Sinhala language are my own work. The datasets we use here own and distributed by the Language Technology Research Laboratory (LTRL) of University of Colombo School of Computing (UCSC) under the Lesser General Public License (LGPL) for Linguistic Resources.

Acknowledgement

I would like to express my sincere gratitude to my research supervisor, Dr. A.R. Weerasinghe, senior lecturer of University of Colombo School of Computing and my research co-supervisor, Mr. K.V.D.J.P. Kumarasinghe, lecturer of University of Colombo School of Computing for providing me continuous guidance and supervision throughout the research.

I would also like to extend my sincere gratitude to Dr. D.A.S.Atukorale, senior lecturer of University of Colombo School of Computing and Dr. K.H.E.L.W. Hettiarachchi, senior lecturer of University of Colombo School of Computing for providing feedback on my research proposal and interim evaluation to improve my study. I also take the opportunity to acknowledge the assistance provided by Dr. H.E.M.H.B. Ekanayakeas the final year computer science project coordinator.

I appreciate the feedback and motivation provided by my friends to achieve my research goals. This thesis is also dedicated to my loving family who has been an immense support to me throughout this journey of life. It is a great pleasure for me to acknowledge the assistance and contribution of all the people who helped me to successfully complete my research.

Contents

List of Figures

List of Tables

Listings

List of Algorithms

Acronyms

BoW Bag of Words.

CNN Convolutional Neural Network.

CoNLL Conference on Natural Language Learning.

CPU Central Processing Unit.

CRF Conditional Random Fields.

DAgger Dataset Aggregation.

FIGER Fine-Grained Entity Recognizer.

GB Gigabyte.

GPU Graphics Processing Unit.

HDF5 Hierarchical Data Format 5.

HMM Hidden Markov Model.

ICON International Conference on Natural Language Processing.

IOB Inside, Outside, and Beginning.

LOLS Locally Optimal Learning to Search.

LSTM Long Short-Term Memory.

MaxEnt Maximum Entropy Model.

ML Machine Learning.

NER Named Entity Recognition.

NLP Natural Language Processing.

POS Part of Speech.

RAIL Reduction-based Active Imitation Learning.

RNN Recurrent Neural Network.

SEARN Search-based Structured Prediction.

SMILe Stochastic Mixing Iterative Learning.

SVM Support Vector Machine.

UTF Unicode Transformation Format.

Chapter 1

Introduction

1.1 Background to the Research

Natural Language Processing (NLP) is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do beneficial things. Foundation of NLP based on with computer and information sciences, linguistics, mathematics, electrical and electronic engineering, artificial intelligence and robotics, and psychology [9]. Retrieving the required information more efficiently from large unstructured contexts of native languages is a vital task to achieve in the day to day applications in NLP. In order to retrieve the required information, it is essential to have NLP tasks, namely, Information Extraction, Machine Translation, Automatic Summarization, and Information Retrieval. Named Entity Recognition (NER) and Part of Speech (POS) tagging are prerequisites for the above NLP tasks. The soundness of the above NLP tasks depends on with the accuracies of the Named Entity Recognition (NER) and Part of Speech (POS) tagging.

Named Entity Recognition (NER) is the process of identifying, locate and classification of named entities into predefined classes. Those predefined classes are person names, organizations, locations, expressions of times, quantities, monetary values, and percentages [19].

Part of Speech (POS) tagging is the process of labeling the words in a sentence to its corresponding Part of Speech (POS) tags. Noun, verb, adjective, and adverbs are examples for POS tags [13].

In order to automate the NER and POS tagging tasks, researchers have pursued three main approaches. Those approaches are the rule-based approach, statistical approach, and hybrid approach. In the rule-based approach, linguistics create a set of rules to identify named entities or assign POS tags. Since rule-based approach suffers from the problem of lack of generalizability of the rule sets to other languages, researchers have moved into the statistical approach which based on with corpus training method. The statistical approach totally based on with machine learning (ML) techniques. The hybrid approach is the combination of rule-based and statistical approach. More recent approaches of NER and POS tagging follow statistical approach and hybrid approach because of the practical flaws of the rule-based approach. Further details about these three approaches have described in chapter 2 literature review.

1.2 Research Problem and Research Question

1.2.1 Research Problem

In order to prevent from the lack of generalizability problem in the rule-based approach, researchers more concern about the statistical and hybrid approaches which based on with the machine learning strategy. Usage of a large data corpus[1] in the training phase is the primary influence to obtain higher accuracies in machine learning based models. However, receiving that kind of large dataset at the initial stage is challenging because it consumes more time and human resources. In practical NER and POS tagging models receive data corpora[2] at different time intervals. The traditional training process of machine learning based NER and POS tagging models can be described as follow.

1. Suppose at the initial stage of the project, obtain a 10GB data corpus. Then machine learning (ML) model train using that data corpus.

[1] Data corpus is a large collection of texts

[2] Data corpora means that plural of a data corpus

2. After a few months, another 8GB data corpus arrives. Then aggregate the current 8GB data corpus with initial 10GB data corpus and train the ML model from scratch to the whole 18GB (10GB + 8GB) data corpus.
3. Again obtain another 4GB data corpus. Then aggregate it with previous 10GB and 8GB data corpora. After that, train the ML model from scratch using all the 22GB (10GB + 8GB + 4GB) data corpus.

Likewise, when analyzing the traditional training process, it is clear that the machine learning (ML) model retrains into the same data corpus repetitively. That retraining process causes a higher training time. The primary motivation to carry out this research is to avoid this overhead of retraining. Avoiding retraining when data becomes available in different time steps is not enough for NER and POS tagging. In the meantime, NER and POS tagging models should result in better accuracy values like traditional retraining NER and POS tagging models.

In natural languages, the meaning of some words become vary with respect to the time period. Hence, NER and POS tagging models should understand the present context of the natural language. It is worthless to use NER and POS tagging models which trained using old data sources. From time to time NER and POS tagging models should train for present data resources. Then, the results of NER and POS tagging models adapt to the present domain of the natural language. Providing this capability of adapting NER and POS tagging based models into the current context is a partial consideration of our research.

1.2.2 Research Question

How to ***adapt and enhance*** *an* ***NER model*** *to learn the* ***most recent data*** *without* ***explicitly retraining to the entire dataset*** *for the* ***Sinhala language****?*

The research question contains the following key points:

- Learn the model using most recent data

- No explicit retraining to the entire dataset
- Adapt and enhance the model
- Build an NER model for Sinhala language

The data arrives at different time steps into this NLP task (NER). If the first data corpus arrived to the model for training at the (*t*) time step, then the second data corpus will be available at the (*t+1*) time step. Likewise, NLP task receives data corpora at *(t+2), (t+3), ...* time steps. When data corpora are available in different time steps, the NER model does not explicitly retrain to the whole data corpora. Instead, the NER model adapts for the most recent data corpus. The enhancement of the model means that, the overall performance of the model should increase in the each training step. Further, the training process of the proposed solution should not decrease the overall performance of the model.

The research findings will be investigated by implementing an NER model. Thus the scope is limited to implement an NER model. The NER model that present in this research based on with the Sinhala language.

1.3 Research Aim and Objectives

1.3.1 Research Aim

The aim of the project is to find out NER model which can adapt and enhance the NER model using the data sources that comes at different time intervals.

1.3.2 Research Objectives

In order to accomplish the goal of the research, several objectives have to be fulfilled as below:

- **Online learning based algorithm to avoid retraining**

  The primary objective is to find out online learning based NER algorithm that avoids the overhead of retraining. Whenever data become available in sequential order according to the time dimension, the online learning based NER model should be capable of the train only on the current data corpus and enhance the model.

- **Increase the overall accuracy**

  Avoid retraining into whole data corpora is not enough for practical NER models. In the meantime, the NER model should have higher accuracy. Moreover, when ML models train using distinct data corpora in an incremental manner, the accuracy values should also improve incrementally.

- **Training time optimizations**

  When machine learning models train to a large data corpus at once, those models consume more training time. Hence the online learning based NER model should consume less training time in the training phase.

- **Proof the application of the purposed model into other NLP tasks**

  Solving the retraining overhead only on NER does not enough to accelerate the efficiency of NLP solutions. In the meantime, this study examines another significant preprocessing task in NLP, which is POS tagging. Final investigations have to provide a formal argument that, proposed online learning based model has the capability of applying to POS tagging too.

- **Find out dataset size that needs to obtain maximum performance in Sinhala NER and POS tagging**

  The proposed approach of the research should have the capability to train the model using mini batches which available at different time intervals. After train on enough amount of mini-batches, the results depict a variation of the accuracies. From that, this study attempts to predict what dataset size gives much better performance near to human performance from our models.

1.4 Justification for the Research

Chapter 2 discussed the existing NER approaches which related to online and incremental learning. Most of the online learning approaches applied to English and German languages which have large annotated data corpora for training [7, 12]. However, the research problem of this study occurred for the languages, which already have small data corpora for training. That kind of low resources languages obtains a large amount of data as a stream. Online and incremental learning approach is essential to train on that kind of stream data in an incremental manner. There are no online and incremental learning models which employed for the Sinhala NLP tasks. Hence observing this kind of approach is an essential thing for low resource languages like Sinhala. The concluding solution of our research should have the capacity of generalizing it to several languages easily. The proposed solution can be used for other natural languages.

The characteristics of batch learning technique are the dominant causes to the research problem. The batch learning technique not only applied for NLP tasks but also applied to other real-world applications like digital signal processing and DNA analysis. This research problem can occur in those applications. The proposed solution of this research could be used for that kind of real-world applications as well.

1.5 Methodology

The problem in this research occurred due to the flaws in batch learning techniques. This research study tries to find out a solution to the research problem using online learning method because online learning can train ML models incrementally. This research attempts to test the hypothesis of, is it possible to apply online learning methods to accomplish adaptation and avoid retraining for NLP problems. This research follows the quantitative approach to test the hypothesis. There is a data corpus which has ten million words. This research attempts to test our hypothesis using that dataset. Hence, the investigation follows a deductive approach to test this hypothesis.

In the conceptual level of the research, it has a clear problem statement of retraining overhead when data comes as a stream to the model. In the meantime, this research follows the clear pathway of online learning to solve the research problem. Even though it is not clear which online learning algorithm would solve the problem, the research has a clear perspective to find out a solution. Thus, this research has a clear theoretical framework. This research follows the empirical approach to solve the problem using an existing data set. This research project already has a dataset and a particular approach to apply. The online learning algorithms train using that data corpus. These online algorithms try to find out patterns from the data set. This research follows the positivism approach in the research methodology. Since the study follows an empirical research approach, the research methodology has to follow the main steps in the empirical research cycle like observation, induction, deduction, testing, and evaluation as shown in Figure 1.1. Each step of the research methodology depicted in Figure 1.1 can describe as follow:

1. **Observation**

   First, the process starts with the observation step. This research perceives the phenomena of the batch learning into the NLP tasks in this step. This step of the methodology contains the observations about the online learning and batch learning techniques.

2. **Induction**

   This step formulates the hypothesis that online learning can avoid this retraining and give adaptation to NER when data comes at different time intervals.

3. **Deduction**

   This step of the research test the hypothesis which built on the previous step. The experiments formulated using the following two cases:

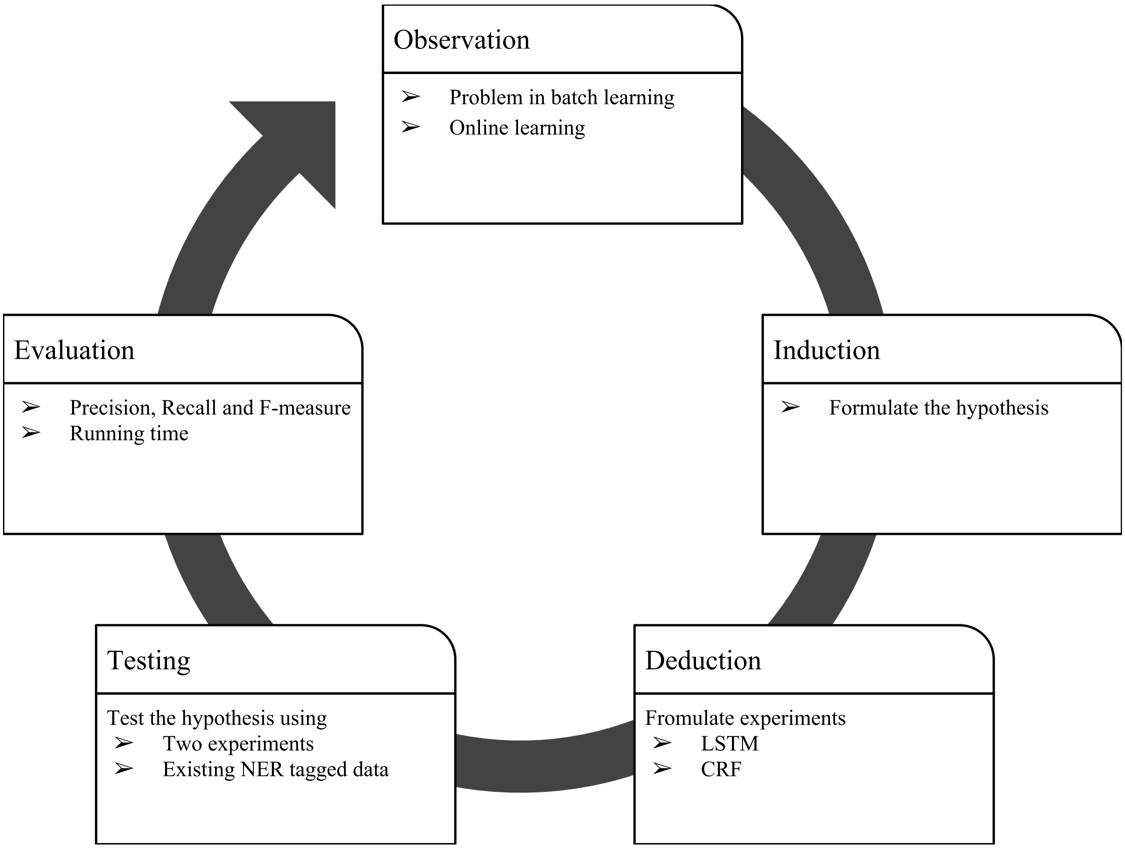

Observation
➢ Problem in batch learning
➢ Online learning
Induction
➢ Formulate the hypothesis
Deduction
Fromulate experiments
➢ LSTM
➢ CRF
Testing
Test the hypothesis using
➢ Two experiments
➢ Existing NER tagged data
Evaluation
➢ Precision, Recall and F-measure
➢ Running time

Figure 1.1 – Research Methodology - Steps of the Empirical Research Cycle

(a) Convert existing statistical NER approach into online learning(CRF)

(b) Using the Long Short-Term Memory(LSTM) model for Sinhala NER

4. **Testing**

   This step tests the hypothesis using the formulated two cases in the previous step and the data corpus.

5. **Evaluation**

   In this step, the research evaluates results according to the evaluation plan described in chapter 5 Results and Evaluation.

1.6 Outline of the Dissertation

This dissertation consists of six chapters as described below.

1. **Introduction**

   This chapter comprises the specific introduction to the research. Mainly presented the introduction to this research, research problem, research question, research aims & objectives, a justification for this research, and the methodology of this research.

2. **Literature**

   This chapter explains the related works and the background theories related to this research. Mainly focused on to find out existing works related to NER & POS tagging approaches and online & incremental learning based approaches for NLP.

3. **Design**

   This chapter presented about the dataset and the design of this research with relevant theoretical details.

4. **Implementation**

   The details of research implementation included in this chapter. Furthermore, this chapter contains the details of the software tools used in the research and the algorithmic level details of the proposed solution. This chapter also comprises the online learning algorithm which used to solve the problem and code level implementation of the proposed online learning model.

5. **Results and Evaluation**

   This chapter first presented the evaluation model of this research and then describe the results using graphs and tables.

6. **Conclusion**

   Finally, in this last chapter present our conclusions to the research problem and the question. Then discuss the limitations of the research and the future works of this research.

1.7 Delimitation of Scope

The research mainly focuses on building online learning based NER model which avoid retraining overhead and provide adaptation for the current context of the language. The primary influence to solve this problem in this research is to convert existing statistical NER approaches into online learning. This research mainly considers transforming a CRF model into online learning methods.

The proposed solutions demonstrated using an NER model. This research does not directly consider NLP tasks other than NER. Sinhala is the natural language that these NLP tasks apply. This research does not examine natural languages other than Sinhala.

The research problem in this research not only apply for NER but also these problems appear in other NLP task like POS tagging. The research scope does not consider to build machine learning models other than NER. However, the research should provide a particular argument which shows the applicability of the proposed online learning model to other NLP tasks like POS tagging.

1.8 Summary

In this chapter discussed the background area of this research that incorporated with the theoretical aspects and the practical evidence. After that, the research problem precisely defined and explained the research question. Research question followed by the main aim and the objectives of this research. The justification section discussed the knowledge gap in this research and how to fulfill those gaps in this research. Methodological details of this research process illustrated in the methodology section. Hereafter, this chapter stated brief outline details about each chapter of this dissertation. Delimitations of scope section described the things that in scope and out scope details in this research.

Chapter 2

Literature Review

The research tries to solve the training time overhead problem in NLP tasks. The research problem arises due to the characteristics of batch learning techniques [6]. Hence, our solution strategy based on online learning techniques. The batch learning techniques use the entire data corpus at once to generate the output hypothesis, which is a function F that maps instances of an input set X to a label set Y. On the other hand, online learning is a training algorithm in which a learner operates on a sequence of data entries which available on different time steps [6]. As described in the research question in chapter 1 this research has focused on to develop an NER model to proof the work. Hence, the overall literature can be described under two main categories:

1. NER and POS tagging
2. Online and Incremental Learning

In section 2.1 of literature describes the existing approaches to the NER and POS tagging. On the other hand, section 2.2 mainly focuses on to find out online and incremental learning based approaches which applied to NER and POS tagging.

2.1 NER and POS tagging

The existing NER and POS tagging based research approaches can be categorized into three main categories as follow:

1. Rule-based approach

2. Statistical approach
3. Hybrid approach

2.1.1 Rule-based Approach

The rule-based approach is the most classical approach for NER and POS tagging. In this methods, several linguists create a set of rules to identify named entities or part of speech tags. The rule set parsed the given text and generate an intermediate representation of the text to identify named entities and POS tags. The intermediate representation could be a parse tree or some abstract representation. Even though the rule-based methods could obtain better results compared to other approaches, it needs more human effort to build such kind of rule sets. The rule-based approach is extremely language dependent one. For example, the rule set developed for the English language cannot be used for the Sinhala language [14, 19].

2.1.2 Statistical Approach

The researchers move on to the statistical approaches because of the shortcomings and the inefficiency of the rule-based method. The support that needs from linguists is really low in the statistical approach. The statistical models have the capability of generalizing for different languages. According to the survey [19], the statistical models can categorize as follow.

1. Supervised Learning
2. Semi-supervised Learning
3. Unsupervised Learning

2.1.2.1 Supervised Learning

Supervised learning has several models for NER and POS tagging like Hidden Markov Model (HMM), Maximum Entropy Model (MaxEnt), Conditional Random Fields (CRF), Decision tree and Support Vector Machine (SVM). Supervised learning needs large annotated data corpus for the training phase. Supervised learning approach typically consists of a system that reads the large annotated corpus and train according to that corpus. Then the NER or POS tagging were performed by the previously trained model. The supervised learning models cannot achieve better results if it has not large annotated data corpus for training. Usually, the size of the annotated data corpus for training is proportional to the final accuracy of the model [14]. Out of all the supervised learning methods, this study more focus on the CRF model because it obtains better results in Sinhala NER [10]. Further paragraphs describe abstract theoretical concepts of the HMM, MaxEnt and CRF models because the research try to change the underlying mechanism of these models to achieve adaptation in the learning process.

HMM, is one of the earliest models that applied to NER problem [10]. Markov chain is a mathematical system that undergoes a transition from one state to another in a chain-like manner. The number of states in the Markov chain should be finite or countable. Markov chain has the Markov property which means that the next state depends only on the current state and not on the past. Let $x_1, x_2, x_3, ...x_k, ...x_n$ is the input that took by HMM model where $x_i \in X$. X is the set of all words that in the data set. In an NER model, these are the words of the sentences that input to the HMM model. The named entities that predict can be represented as $z_1, z_2, z_3, ...z_k, ...z_n$ corresponding to the input where $z_i \in Z$. Z is the set of all named entity tags. HMM, considers two probability values. First one is the emission probability that can be denoted as $P(x_k|z_k)$. Second one is the transition probability that can be denoted as $P(z_k|z_{k\text{-}1})$. The HMM algorithm for prediction based on with two parts: Forward algorithm and Backward algorithm. The forward and backward algorithms are recursions which based on with emission and transition probabilities [22]. However, the HMM model suffers from long distance dependency problem [10]. Long distance dependency problem means that these kinds of models unable to handle the dependency between two entities that have long distance.

Maximum Entropy model (MaxEnt) is another statistical model that follows the maximum entropy principle. The maximum entropy principle state that the correct distribution $P(X, Z)$ which maximize the entropy or uncertainty [21]. In the maximum entropy theory *X* denotes the set of all contexts and the *Z* denotes the set of all classes to be predicted. The *p* should maximize the entropy

$$H(p) = -\sum_{k \in \xi} p(k)log(p(k)) \tag{2.1}$$

where $k = (x, z), x \in X, z \in Z$ *and* $\xi = X \times Z$. The maximum entropy model is about all things that are known. It assumes nothing about which is unknown. In other words, unknown things are distributed uniformly [21]. The maximum entropy model solves the long-distance dependency problem. However, the maximum entropy model suffers from the label bias problem [21].

Machine learning models which independently trained next state classifiers are potential victims of the label bias problem. Suppose a simple finite state machine which was developed for named entity recognition. In those kinds of machines the states with a single outgoing transition effectively ignore their observation. In other words, the states with a single transition simply have to move to the next state without considering their current observation [17].

CRF is an undirected graphical model and matches up with the conditionally trained probabilistic finite state automata. One of the main advantages of CRF is capable of including arbitrary features easily because it trained conditionally. CRF model reduces the overhead of independence assumption that required in HMM model. Let suppose *(*X) as a set of feature vectors corresponding to particular data coupus and *Z* as the set of corresponding NER labels of each word in *X*. CRF model can be considered as a graph *G = (V,E)*. So that each element $z_i \in Z$ represent by the vertices of *G*. The graphical model *G* is a conditional random field when the random variable $z_i \in Z$ is conditioned on X and obey the Markov property according to the graph G as $p(z_i|X, z_i, i \sim j)$ where $i \sim j$ means i and j vertices are neighbours in G [17].

2.1.2.2 Semi-supervised Learning

The semi-supervised learning techniques used both labeled and unlabeled data corpus to train the model. Hence it falls in between supervised and unsupervised learning methods. The initial entities called seeds input to the semi-supervised learning model and trigger the learning procedure. Then the system searches for similar seeds and identifies them. After that, the system tries to identify other entities that are in a similar context. Then the other entities added to the overall seeds set. Using that overall set of seeds the semi-supervised model train again. Likewise, the semi-supervised learning model trains iteratively. The bootstrapping is the main semi-supervised learning method [14].

2.1.2.3 Unsupervised Learning

The unsupervised learning methods learn without using annotated data corpus. Also, the unsupervised learning method learns without any feedback or past data. The unsupervised learning methods more suitable for natural languages which have not large annotated data corpus. The goal of unsupervised learning is to give representations from data and those representations can be used for data comparison, classification and decision making. Clustering is the main unsupervised learning technique used in NER and POS tagging. Even though unsupervised methods can apply to the vast range of natural languages, it cannot be obtained better accuracies compared to supervised and semi-supervised methods [14, 19].

2.1.3 Hybrid Approach

The rule-based approach can achieve better accuracies compared to the statistical approach. On the other hand, the statistical approach has the ability to generalize among several languages. Therefore the researchers have tried the hybrid approach which is a combination of rule-based methods and the statistical methods. If the statistical model fails in a certain context, then the rule set can be used for that kind of contexts for predictions[14]. Gunasekara *et al.* [13] have used a hybrid approach for POS tagging in the Sinhala language. As final results, Gunasekara *et al.* [13] have obtained the overall accuracy of 72% when the average unknown word percentage is 20%.

2.2 Online and Incremental Learning

This section observes the existing online and incremental learning methodologies which applied for NLP tasks. The online learning algorithms execute the training process on the data that available at different time intervals. Incremental learning is an online learning strategy which works on with limited memory resources. Also, incremental learning has to depend on with the signals which related to already observed data instances [11]. Gepperth and Hammer [11] have emphasized the key challenges which related to the online and incremental learning as follow.

- **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. These changes refer as the *concept drift*. The implementation of the online learning algorithm should have the capability to cope with these changes.

- **Catastrophic Forgetting**

  Catastrophic forgetting means that the tendency of an artificial neural network to forget certain things which learned previously upon newly learn things. Catastrophic forgetting seems like disadvantageous for a machine learning model. Though, for some scenarios, it may be useful like, when someone needs more adaptation for the current context rather than keeping more previously learned information. As an example, some words in a natural languages change time to time. In order to learn the current meaning of that kind of word precisely the catastrophic forgetting would be much useful. Since the catastrophic forgetting has both positive and negative things online learning algorithm should have the capability to manage it into a certain extent.

- **Stability-Plasticity Dilemma**

  Stability-plasticity dilemma refers to that when the online learning algorithm executes quick updates it has a high tendency to forget things proportionally. On the other hand, if the online learning algorithm decelerates the updates then the model will keep previously learned information for a long time, but the reactivity of the system becomes lower.

The literature related to online and incremental learning section further categorizes into three main algorithm as follow:

1. Initial Online and Incremental Learning Algorithms
2. Imitation Learning based Algorithms
3. Deep Learning based Algorithms

2.2.1 Initial Online and Incremental Learning Algorithms

Carreras *et al.* [7] have observed the applicability of the online learning algorithms into the NER task. In this research, Carreras *et al.* introduced a new approach for NER called *voted perceptron* which based on with online perceptron strategy. The online algorithm that Carreras *et al.* have proposed was a mistake driven online algorithm. The execution of the online algorithm could be categorized into two phases. First, the algorithm applied to learn at word level in order to identify named entity candidates by means of a Begin-Inside classification. Then the algorithm makes use of functions learned at the phrase level. Carreras *et al.* have applied the online learning strategy at a sentence level. In our research problem, we are trying to apply online learning strategy at a mini batch level. Carreras *et al.* have performed their experiments for the English and German languages. For the English language, they have obtained overall precision, recall and F-measure values of 85.81%, 82.84%, and 84.30% respectively. For the German language, experiments have obtained overall precision, recall and F-measure values of 77.83%, 58.02%, and 66.48% respectively.

Gimpel *et al.* [12], have tried to find out the applicability of online strategy in the distributed manner in order to achieve asynchronous learning. Then they have applied this online learning strategy for NER and POS tagging. Gimpel *et al.* mainly understand the problems that they had in synchronous approach in which wasting CPU time of machines when waiting until others complete. In order to solve this problem, they have proposed an asynchronous algorithm which executes under the mini batch learning techniques.

2.2.2 Imitation Leaning based Algorithms

The NER and POS tagging problems can be considered as a structured prediction task because the NER and POS tagging are predicting structured objects that corresponding to a particular sentence rather than predicting a single prediction value. Imitation learning is one of the major methods that apply to structured prediction problems and results in better accuracies. Since imitation learning has a close relationship with online learning and reinforcement learning, we observe the imitation learning techniques in the literature. Imitation Learning refers to a sequential task where the learning algorithm tries to mimic the behavior of an expert in the context [1].

Attia and Dayan [1] have provided the overall idea of the imitation learning algorithms and how those algorithms could be applied to the structured predictions tasks. Attia and Dayan have stated the evolution of the imitation learning algorithms with respect to the overall accuracy values obtained from those algorithms. Attia and Dayan have explained the execution process of the several imitation learning algorithms such as SEARN, SMILe, RAIL and Dataset Aggregation (DAgger). However, the DAgger is the most common algorithm on nowadays, because it generally outperforms the other imitation learning models.

Augenstein *et al.* [16] have used the DAgger algorithm to extract the relations between non-standard entities. Augenstein *et al.* have used a combination of distant supervision and imitation learning to extract relations between non-standard entities. They have obtained precision values than 10 points and 19 points higher in the imitation learning methods compared to the FIGER and the Stanford NER models.

Chang [9] has explained the theoretical aspects and the practical application of Locally Optimal Learning to Search(LOLS) algorithm in imitation learning. The previous imitation learning algorithms like DAgger performs better only if the reference policy performs better. If the reference policy was suboptimal then the overall accuracy of the previous models(DAgger) could not obtain better results. LOLS is the solution for that kind of scenarios. Even if the reference policy was suboptimal the LOLS based model could reach the higher accuracies. The LOLS algorithm consists of two main parts: roll-in and roll-out. The roll-in policy comprises the learned policy, and the roll-out policy consists of the composite method of the reference and the learned policy.

The main objective of finding the information about the imitation learning algorithms such as DAgger and the LOLS is to observe how to handle reference policy(Oracle) and obtained better results even if reference policy was suboptimal. Handling the loss function and updating the current model in imitation learning are major things that apply to the proposed NER model.

2.2.3 Deep Leanirng based Algorithms

The main reason to observe the literature about RNN and LSTM, because those models have the online learning capability within the network architecture [5]. As discussed previously, the NER and POS tagging problem can be considered as a structured prediction problem. Hence the researchers have applied the recurrent neural network(RNN) approach to these NER and POS tagging tasks [2, 8, 15]. RNN has the capability of handling sequence and structured prediction tasks because of its internal chain-like structure. However, the RNN model performs poorly, when there exist long-term dependencies in the sequence prediction task. The NER and POS tagging tasks have consisted with those long-term dependencies. Hence the RNN layers perform poorly on those tasks [2].

In order to handle these long-term dependencies researchers have enhanced the internal structure of the RNN cells into LSTM cells. The LSTM is also a type of RNN which has the capability of handling long-term dependencies. The LSTM cells consist of three main gates as follow. These gates are used to keep track of the previously learned information [20].

1. **Forget Gate**

   This forget gate decides which information need to throw away from the cell state. It has used a *sigmoid function* to make this decision.

2. **Input Gate**

   Input layer decides which information need to store within the cell. The input layer has used a *sigmoid layer* and a *tanh layer* to make this decision.

3. **Output Gate**

   Using another *sigmoid layer,* the output gate decides what kind of information to be output by the cell.

Athavale *et al.* [2] have applied the deep neural network approach for Hindi NER using several RNN layers. Athavale *et al.* have used a neural network which has four layers: embedding layer, two recurrent layers, and a final softmax layer. For their experiments, Athavale *et al.* have used three recurrent layers: Vanilla RNN, LSTM, and bi-directional LSTM. From these three recurrent layers, the bi-directional LSTM layer outperforms other layers. As the final output, Athavale *et al.* have obtained 90.32% accuracy for CoNLL-2003 dataset without using any Gazetteer information and achieve 77.48% accuracy for ICON NLP Tools 2013 dataset without using any Gazetteer information or chunking information from the bi-directional LSTM model.

Jason *et al.* [8] have implemented hybrid bidirectional LSTM and CNN architecture for their classification. Jason *et al.* have used word level and character level features, for the NER classification. The word embeddings and the character embeddings were the main features used in this research. As a language dependent feature, Jason *et al.* have used capitalization feature in their experiments. Finally, Jason *et al.* obtained 91.62 F1 score value on the CoNLL-2003 dataset.

Huang *et al.* [15] have combined an LSTM model and a CRF model and executed their experiments. For their experiments, Huang *et al.* have used four main network architectures: LSTM network, bi-directional LSTM network, LSTM network with CRF layer and bi-directional LSTM network with a CRF layer. Compared to other neural network architectures that Huang *et al.* have used here, the bi-directional LSTM network with a CRF layer architecture performs well. The reasons for the success of the bi-directional LSTM-CRF are, bi-directional LSTM layer has the capability of using past and future features to make the predictions and the CRF layer has the capability of handling sentence level features. Finally, Huang *et al.* have obtained 97.55% accuracy from the bi-directional LSTM-CRF model.

2.3 Summary

This chapter mainly considered the existing research works that related to our research area. First of all, this chapter elaborated two main areas which used to literature. Those two areas were NER & POS tagging, and online & incremental learning. Then observed the existing NER and POS tagging based researches according to three main approaches: rule-based approach, statistical approach, and hybrid approach. The second section observed about the online and incremental learning based researches and discussed it using three main approaches: 1). Initial online and incremental learning algorithms, 2). Imitation learning-based algorithms, and 3). Deep learning-based algorithms.

Chapter 3

Design

This chapter covered the detailed description of the research design. The overall research design of this chapter presents three main concerns:

1. Dataset
2. Design Overview
3. Abstract Design

Section 3.1 explains the details about the datasets which used in this research. Section 3.2 describes the detailed description of the research design step by step. The abstract view of the final model of this research describes in section 3.3.

3.1 Dataset

The first dataset used in this research was an annotated data corpus for the Sinhala NER tasks. The dataset owns and distributed by the Language Technology Research Laboratory (LTRL) of University of Colombo School of Computing (UCSC) under the Lesser General Public License (LGPL) for Linguistic Resources. The dataset was related to the Sinhala language. Also, the dataset followed the data format which used in CoNLL-2003 data corpora. The dataset mainly contains the Sinhala text in UTF-16 format. Each word in the dataset recorded in a single line along with its corresponding named entity tag. The empty lines in the data file represent the sentence boundaries. The named entity tag of each word followed the IOB format [23]. In IOB format the *I* represents the inside of a named entity, *O* represents the outside of a named entity and *B* indicate the beginning of a named entity. Here only the person names, organization names and locations consider as named entities. The annotated data corpus for this research consists about 80,400 Sinhala words and 3268 sentences.

The experiments observed the applicability of the proposed models into POS tagging. That experiment used an annotated Sinhala POS tagged dataset owns and distributed by the Language Technology Research Laboratory (LTRL) of University of Colombo School of Computing (UCSC) under the Lesser General Public License (LGPL) for Linguistic Resources. This dataset mainly contained 22 POS tags. All text data for this dataset retrieved from Sinhala newspapers, which consists of 10 million words. All text in this dataset exists as paragraphs. The POS tag and the word separated by an underscore in the text in the dataset.

3.2 Design Overview

Since this study follows machine learning techniques to solve the research problem, the research design has to follow the common machine learning pipeline. Thus research design can be described supporting three major steps:

1. Data preprocessing
2. Applying online learning algorithm
3. Postprocessing

Every three steps represent a submodule of the final NER model. The final model should isolate these submodules from each other as much as possible. Each submodule should be loosely coupled and highly cohesive. The main intention of doing that kind of thing is that makes the final model more easy to change for different languages, different domains and even the model can be portable to various features.

3.2.1 Data preprocessing

As discussed in chapter 2, the supervised learning methods result in better accuracy values compared to other approaches. Hence, apply online learning strategy into the supervised learning models. The supervised learning algorithms need an annotated data corpus in the training process. The *step 01* of Figure 3.1 shows the annotated data corpus.

The initial step of the typical machine learning based research design is to split the dataset in a way that assigns partial datasets for training, testing, and validation. The data splitting process shows in *step 02* of Figure 3.1. There exists two experiments for Sinhala NER and the Sinhala POS tagging. The Sinhala NER dataset and POS tagging datasets used the 90:10 ratio for training and testing. Both experiments used 10% of training data for validation. The experiments need several mini-batches of data for training. Therefore if the dataset was not large enough, the size of the mini batch becomes small and the accuracy variations may not be shown up. Since the Sinhala NER and POS tagging datasets were not large enough, the splitting ratio have to be 90:10 proportion. That makes the mini-batch size large. The training, testing, and validation data sets after the splitting process shows in *step 03* of Figure 3.1.

As explained in previous chapters, the solution of the research entirely relies on online and incremental learning techniques. In order to demonstrate the incremental learning analogy of the machine learning algorithm, this study needs multiple distinct data mini-batches. Each mini batch can use in each incremental step of the learning algorithm. Figure 3.1 shows the mini batch creation process in *step 04*. The resulting mini-batches show in *step 05*.

After that the research design going through an iterative process according to the online learning algorithm. One mini batch at a time each mini batch will input into the feature extraction process. Here, the feature extraction process has shown in the *step 06*. The creation of word embedding happens through an

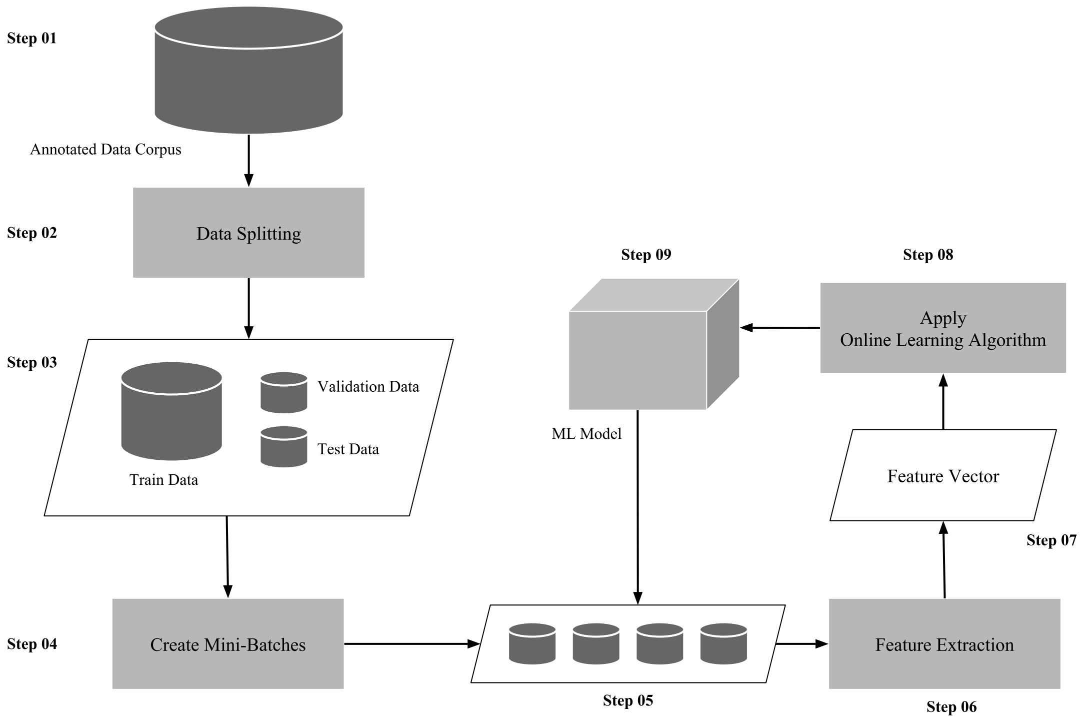


Figure 3.1 – Design Steps of the Research

embedding layer which bypasses into the machine learning model.

3.2.2 Applying online learning algorithm

The feature extraction process results in the corresponding feature vectors and those vectors then input into the training phase of the online learning algorithm as shown in *step 07* and *step 08* of Figure 3.1. After that, the online learning algorithm trains the first iteration of the machine learning algorithm. After completing the first iteration, the online learning algorithm chooses another mini batch from *step 05* and iteratively continue the algorithm. The continuation of each iteration of the algorithm enhances the final machine learning model from the current state. Likewise, these iterative steps (*step 05-09*) execute for every mini-batches. For each mini batch, the validation happens at mini batch level using the allocated dataset for validation.

As discussed in chapter 2, the CRF model performs better for the Sinhala NER task [10]. The NER for other languages like English, the deep learning models perform much better than other models [15]. Also the LSTM model and CRF model have that capability of training as an online algorithm. There are two strategies of online learning algorithms for *step 08* of Figure 3.1.

1. Train CRF model in an incremental way
2. Use Long Short-Term Memory(LSTM) based deep learning model

The main reason for using a deep learning model is the better performance of those models which found out in the literature. Proposed Deep learning-based model had a problem of finding large annotated data corpus of the Sinhala language. Lack of large annotated data corpus is the main reason to use another online learning strategy apart from the deep learning techniques. This study tries to train a CRF model incrementally in this research. The network architectures of the two machine learning models shown in Figure 3.2.

Model A of Figure 3.2 shows the CRF model which learn incrementally. *Model B* of Figure 3.2 shows the network architecture of the deep learning model. The two networks start with an embedding layer which converts words of the input into word embeddings. After that, the *Model A* has a CRF layer. Since deep learning based models hungry for data, this research used a simple model like *Model A* to train on a little amount of data. On the other hand *Model B* depict the deep learning based model. Here in the *Model B*, the embedding layer followed by a bidirectional LSTM layer. The bidirectional LSTM layer has the capability of using past and future context to the prediction. The bidirectional LSTM layer of *Model B* followed by a dropout layer. Since this research follows online learning approach, these models have a high tendency to overfit on a single data corpus. The final model needs to regulate this overfitting and provide more generalizability. That task achieve using a dropout layer [3]. The dropout layer in *Model B* followed by a CRF layer. The CRF layer has the ability to use the sentence level features for the prediction. The architecture of the *Model B* is similar to the network architecture of the Huang *et al.* [15]. In addition to the network architecture of Huang *et al.* [15], this research used a dropout layer to prevent from the overfitting.

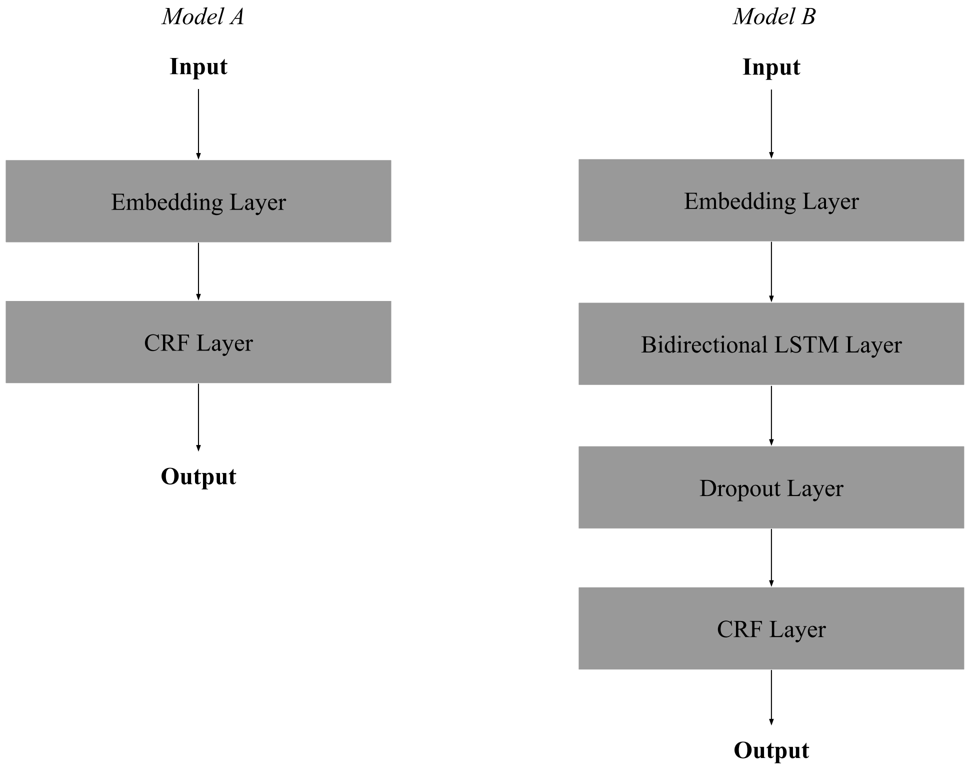


Figure 3.2 – Network Architectures: Online CRF Model and Bidirectional LSTM CRF Model

3.2.3 Postprocessing

The final machine learning model will predict the relevant class for a given word. Though the prediction value is a numerical value. Therefore the numerical value should be mapped to the associated class name. This mapping process happens in the postprocessing step.

3.3 Abstract Design

The final model of the research should have two main functionalities.

1. Training
2. Predicting

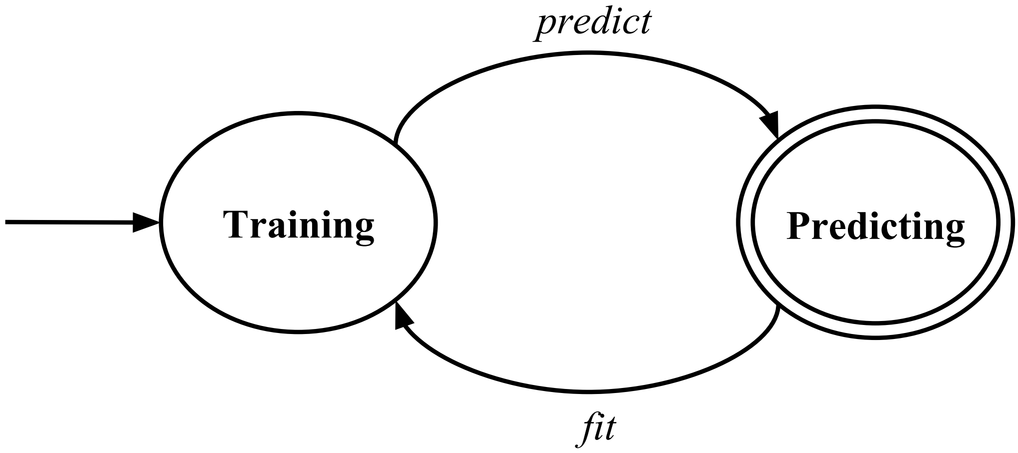


Figure 3.3 – Abstract Transition Diagram of the Proposed Approach

The online learning algorithm initially trains for the first data corpus. After that, the resulting machine learning model can use for real-world predictions. Whenever another data corpus become available for training, the online learning algorithm performs the training phase into that data corpus. Likewise, the machine learning model does prediction and training from time to time. Conceptually, the behavior of the final online learning algorithm can be visualized as Figure 3.3. Mainly the model has two states, *Training* and *Predicting* which corresponding to the training phase and the predicting phase. Here the model uses function call *predict* to switch from training state to predicting state and uses function call *fit* to switch back from predicting state to the training state. In practical, the model iteratively shifts states to train and make predictions.

3.4 Summary

This chapter mainly explained the design aspects of the research. First, this chapter described the datasets that used in the research. After that chapter comprised a detailed description of the research design. The design of the research explained under three main steps: Data preprocessing, Applying online learning algorithm and Postprocessing. After the detailed description of the research design, this chapter discussed the abstract view of the final model.

Chapter 4

Implementation

This chapter mainly focuses on the aspects of the implementation at the algorithmic level. Section 4.1 illustrates the software tools and the libraries used during the implementation. Section 4.2 includes a comprehensive explanation of the implementation.

4.1 Software Tools

The proposed online learning based model was implemented using python 3.5 version. Python has numerous libraries which convenient to implement machine learning models. That was the main reason to choose Python as the programming language to implement the intended solution.

Keras and *keras_contrib* were the main two libraries which used in this implementation. *Keras* library used to implement a sequential machine learning model and import LSTM layers and embedding layers as shown in Figure 3.2. The CRF layer of the machine learning model was developed using the *keras_contrib* library. As described in section 3.3, the proposed model should have the capability of switching into training and predicting states. *Keras* models achieve this using HDF5 technology. Here the *keras* models save the weights of the models using HDF5 data format. HDF5 is a technology which has the capability of managing complex data formats.

Google Colaboratory platform with the GPU support used to train the all machine learning models in this research.

4.2 Implementation Details

Overall implementation of the research can be categorized into three main steps which influenced by the chapter 3 research design.

1. Data Preprocessing
2. Apply Online Learning Algorithm
3. Analyze Results

4.2.1 Data Preprocessing

The preprocessing functionality concisely explained in algorithm 1. The annotated data corpus is stored in a text file as described in section 3.1. That text file inputs into the $PREPROCESS$ function of algorithm 1. First, the $PREPROCESS$ function read data from a file and assign those data into the variable called $text$. Hereafter, using that text, the algorithm initiates a vocabulary. The $PREPROCESS$ function uses two arrays called $data$ and $tokens$ to store the sentences and the corresponding tokens respectively. Hereafter there is a for loop which iterates through each sentence. Each word in a sentence represented using the index value of the vocabulary which corresponds to that word. Index values of each word of a sentence stored in an array called s_words. The corresponding named entity token values of words in a sentence, stored in the s_tokens array. After that, the s_words appends to the final $data$ array and s_tokens array appends to the final $tokens$ array. At the end of the algorithm 1, it returns the $data$ array and the $tokens$ array.

4.2.2 Apply Online Learning Algorithm

Applying the online learning algorithm explains in algorithm 2. The return results$(data, tokens)$ of algorithm 1 input into algorithm 2. First, algorithm 2

Algorithm 1 Data Preprocessing

```
procedure PREPROCESS(file)
    text $\leftarrow$ read(file)
    vocabulary $\leftarrow$ CREATE_VOCABULARY(text)
    data $\leftarrow$ [ ]
    tokens $\leftarrow$ [ ]
    for sentence in text do
        s_words $\leftarrow$ VOCABULARY_INDEX(sentence)
        s_tokens $\leftarrow$ TOKENS(sentence)
        data.add(s_words)
        tokens.add(s_tokens)
    end for
    return (data, tokens)
end procedure
```

split the data and tokens to generate training, testing and validation data. After that, algorithm 2 creates four mini-batches to use in each iteration of the online learning algorithm. Hereafter, algorithm 2 initializes the particular online learning based machine learning model ML. After that, there exists a for loop which iterates through each mini batch. For each mini batch, algorithm 2 does training, validation, and testing. The testing results stored in a variable called $results$

There exist two configurations of ML which corresponding to the two networks architecture that used in this research as shown in Figure 3.2. The code 4.1 shows the implementation of the online learning based CRF model. That CRF sequential model starts with an embedding layer. Word embedding is a technique which used to convert words into vector formats. Earlier methods like the bag of words (BoW) convert words into vector representations. However, these methods like BoW produces a sparse vector space for words. Hence, there can be unseen words which do not fit into that vector space. The vectors of that kind of unseen words inadequate to represent features of that words well. Though, the embedding layer produces a dense representation of words. The embedding layer resolves that problem and provides better representation to the given word [4]. The embedding layer followed by a CRF layer. The CRF layer uses $adam$ optimizer and the CRF loss function.

```python
model = Sequential()
model.add(
    Embedding(len(vocabulary), EMBEDDING_DIMENSION, mask_zero=True)
)
crf = CRF(len(tokens), sparse_target=True)
model.add(crf)
model.compile('adam', loss=crf.loss_function, metrics=[crf.accuracy])

```

Listing 4.1 – Python Implementaion of the Online CRF Model

On the other hand, code 4.2 shows the python implementation of the bidirectional LSTM model with a CRF layer. This implementation of the network architecture influenced by the work of Huang et al.[15] and the implementation details of the [18]. This research study changes the previous model by adding a dropout layer. This model also begins with an embedding layer. The embedding layer followed by a bidirectional LSTM layer. The $return\_sequence = True$ in the bidirectional LSTM layer indicates that the output format to be a sequence instead of a single vector. The *kernel_initializer=’RandomNormal’* in the LSTM layer indicates the initial weights of the model follow the *normal* distribution. After the bidirectional LSTM layer, the network has a dropout layer with the dropout value of 0.20. Dropout is a regularization technique used to minimize the overfitting of a neural network. 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 to contribute to the prediction which has done by dropped neurons. Thus specializing certain neurons to specific predictions has been removed from the neural network. Therefore model prevents from overfitting. The *dropout value* indicates the probability of removing randomly selected neurons from training and weight updation [3]. The dropout layer followed by a CRF layer which uses $adam$ optimizer and the CRF loss function.

```python
model = Sequential()
model.add(
    Embedding(len(vocabulary), EMBEDDING_DIMENSION, mask_zero=True)
)
model.add(
    Bidirectional(
        LSTM(
            BIDIR_UNITS//2,
            return_sequences=True,
            kernel_initializer='RandomNormal'
        )
    )
)
model.add(Dropout(0.20))
crf = CRF(len(tokens), sparse_target=True)
model.add(crf)
model.compile('adam',loss=crf.loss_function ,metrics=[crf.accuracy])

```

Listing 4.2 – Python Implementaion of th Bidirectional-LSTM-CRF Model

Algorithm 2 Train and Predict

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

4.2.3 Analyze Results

The analyzing step needs the variable $results$ which is the output of the function $PREDICT$ and actual tokes($test\_tokens$). Using that $results$ and $test\_tokens$ the algorithm 2 executes the analyzing step as shown in *line 12*. In NER, the $ANALYZE$ function calculates the precision, recall and F-measure values for the prediction. In POS tagging the $ANALYZE$ function calculates the accuracy of the model.

4.3 Summary

This chapter elaborated the main concepts of the implementation under two main factors: Software tools and Implementation details. The software tools subtopic described the language and the libraries that used in the implementation phase. Implementation details subtopic discussed the algorithmic approach using pseudo codes. Also, the implementation details provided the explanation about the machine learning models using the implementation code.

Chapter 5

Results and Evaluation

This chapter mainly describes the details related to the results and the evaluation process of the research. Section 5.1 describes the evaluation model of the research. The following section 5.2 discovers the final performance results of the two online learning models of the research and the evaluation of the results according to the evaluation model. After that, section 5.3 describes the efficiency measurements of the two models. Here mainly consider the training time efficiency of the models. Finally, section 5.4 examines the predictions of how much of data need to get closer to the human performance in Sinhala NER and POS tagging.

5.1 Evaluation Model

There are multiple types of attributes that are evaluated in this research. According to the research problem, there are three main characteristics to be estimated.

1. Performance of the NER predictions
2. Performance of the POS tagging
3. The efficiency of the model

The performance of the NER model can be evaluated using the following measures[10, 23].

1. Precision(P)

- Precision is the percentage of named entities found by the learning system that is correct

2. Recall(R)

   - Recall is the percentage of named entities present in the corpus that are found by the system

3. F-measure

   - $F1 - measure = \frac{2.P.R}{R+P}$

The final output of the research will give us a model that learn and enhance the performance using the data available as a stream. There are two accuracy values, that correspond to the batch learning model that use currently and the online learning model that this research focuses on to build. In the first step of the evaluation compare and contrast precision, recall and F-measure values of the batch learning and the online learning models.

Secondly, evaluate the POS tagging experiment. POS tagging experiment used the *accuracy* metric to evaluate the model. *Accuracy* is the ratio of the number of correct predictions to the total observations. Since POS tagging have more than 22 labels to be predicted in this task, it is convenient to use accuracy as the evaluation metric instead of precision, recall, and F-measure.

The third step of the evaluation measures the efficiency of the final model. The research mainly considers the training time comparison of the batch learning approach and the online learning approach since testing times are negligible for both.

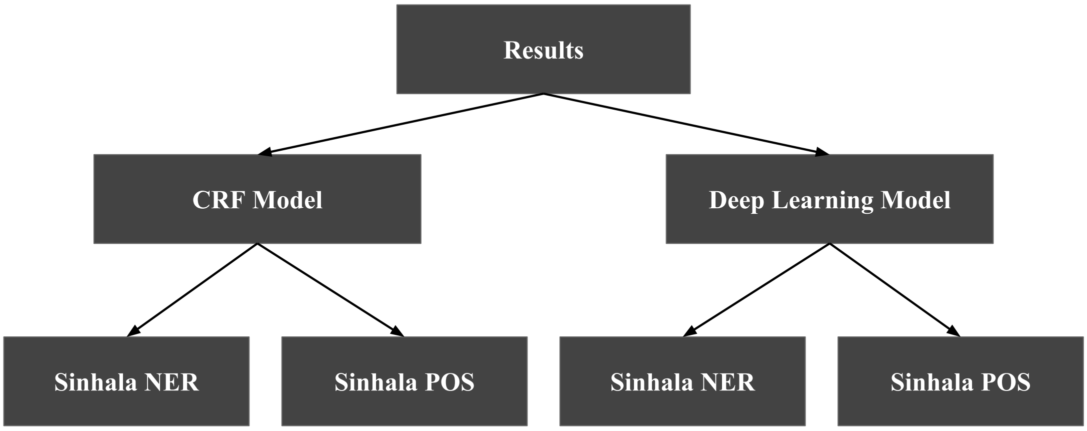


Figure 5.1 – Hierarchical Representation of each Experiment

5.2 Performance

Figure 5.1 shows the final experiments of the two online learning based models proposed in this research.

1. CRF Model
2. Deep Learning Model (Bidirectional LSTM-CRF model)

The proposed models have experimented the Sinhala NER task and the Sinhala POS tagging task as shown in Figure 5.1. Overall research has to analyze the results of four experiments as follow.

1. CRF Model
    - Sinhala-NER Experiment
    - Sinhala-POS Tagging Experiment
2. Deep Learning Model (Bidirectional LSTM-CRF model)
    - Sinhala-NER Experiment
    - Sinhala-POS Tagging Experiment

Each experiment used four mini-batches to simulate online learning and batch learning procedures. Suppose the four mini-batches as $m_1, m_2, m_3,$ and m_4. A separate dataset T used to test the performance of the models. The training phase of the batch learning approach and the online learning approach contained four steps. These four steps of each batch learning and online learning experiments can be described as follow:

- Batch Learning Experiment

Step 1: The batch learning model initially trained using m_1 mini-batch and test on dataset T.

Step 2: When m_2 mini-batch becomes available, aggregate m_1 and m_2 mini-batches and train the model from scratch using whole aggregated $(m_1 + m_2)$ dataset. Then test the model using T.

Step 3: When m_3 mini-batch becomes available, aggregate m_1, m_2 and m_3 mini-batches and train the model from scratch using whole aggregated $(m_1 + m_2 + m_3)$ dataset. Then test the model using T.

Step 4: When m_4 mini-batch becomes available, aggregate m_1, m_2, m_3 and m_4 mini-batches and train the model from scratch using whole aggregated $(m_1 + m_2 + m_3 + m_4)$ dataset. Then test the model using T.

- Online Learning Experiment

Step 1: The online learning model also 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. Then test the model using T.

Step 3: When m_3 becomes available, the model trained only using m_3. Then test the model using T.

Step 4: When m_4 becomes available, the model trained only using m_4. Then test the model using T.

The batch learning experiment consisted of the retraining procedure as demonstrated previously. However, the online learning experiment prevented from this retraining. Each Sinhala NER and Sinhala POS tagging experiments used, four equal sized mini batches of annotated data. For Sinhala NER experiment, the size of a mini-batch was 662 sentences. In the Sinhala POS experiment, a mini-batch size was 740 sentences. Every experiment executed all the previously described steps in batch learning technique and online learning technique. Then compared the performance of corresponding steps of batch learning and online learning.

5.2.1 CRF Model

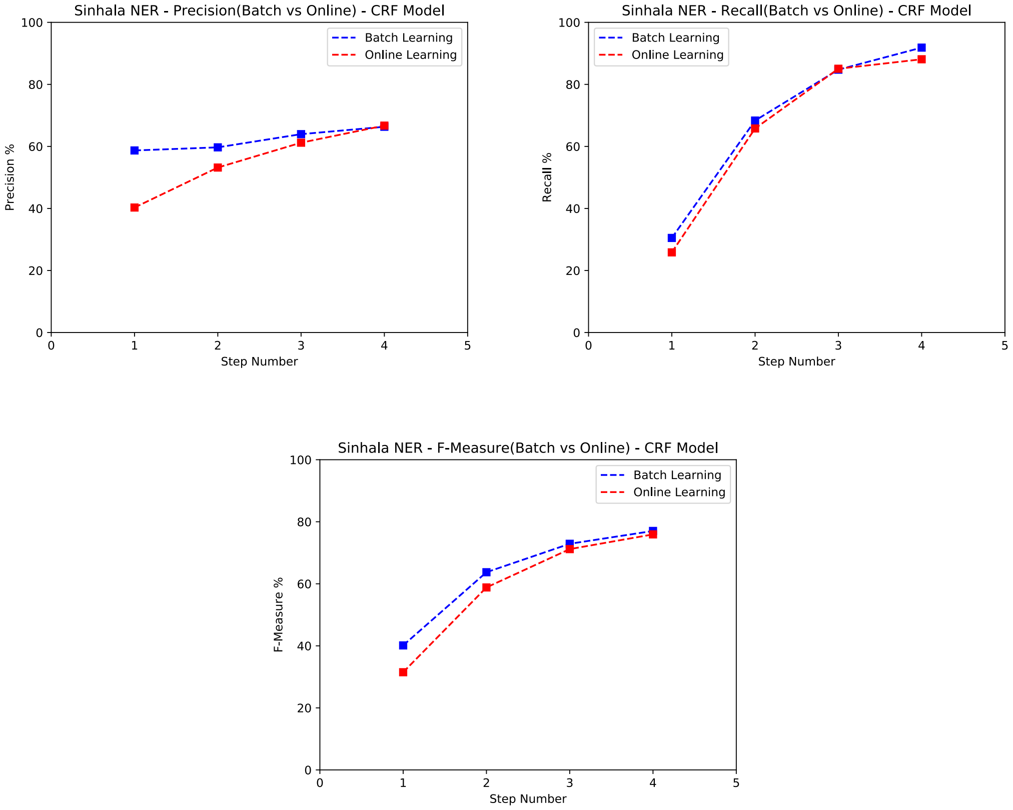
5.2.1.1 Sinhala-NER Experiment

This experiment trained the CRF model for Sinhala NER. Then compared the results of batch learning and proposed online learning model. Figure 5.2 shows the variation of precision, recall, and F-measure of the online and batch learning techniques in each step of the experiment. Moreover, Table 5.1 included the numerical precision, recall, and F-measure value variations of the experiment.

Precision values of the batch learning were higher in the first three steps, but in the fourth step, the online learning method was higher. Recall values of batch learning techniques have bit higher values in all the steps, except the third step of the experiment. F-measure values of batch learning were higher throughout the experiment. However, the online learning F-measure value was closely related to the batch learning in the last mini-batch.

5.2.1.2 Sinhala-POS Tagging Experiment

This experiment observed the applicability of the proposed Online CRF model into the Sinhala POS tagging task. Figure 5.3 depicts the accuracy variation of POS tagging experiment of the batch learning and online learning approaches. The numerical accuracies of the experiment have shown in Table 5.2. The accu-


Sinhala NER - Precision(Batch vs Online) - CRF Model

Sinhala NER - Recall(Batch vs Online) - CRF Model

Sinhala NER - F-Measure(Batch vs Online) - CRF Model

Figure 5.2 – Precision, Recall, and F-Measure value variation of four steps using the CRF model - Sinhala NER

racy values of the online learning model were nearly closed to the batch learning technique. Further, in the first step of the experiment, the online learning model obtained little higher accuracies than batch learning approach.

5.2.2 Bidirectional LSTM CRF model

5.2.2.1 Sinhala-NER Experiment

The Bidirectional LSTM-CRF model for Sinhala NER trained in this experiment. Figure 5.4 shows the variation of precision, recall, and F-measure of the batch learning technique with the online learning technique, in each step of the experiment. The numerical precision, recall, and F-measure values included in Table 5.3.

Table 5.1 – Precision, Recall, and F-Measure value variation of four steps using the CRF model - Sinhala NER

Mini Batch	Precision %		Recall %		F-Measure %	
	Online	Batch	Online	Batch	Online	Batch
1	40.3004	58.6984	25.8427	30.4941	31.4914	40.1369
2	53.1915	59.6996	65.7895	68.3381	58.8235	63.7275
3	61.2015	63.9549	85.0435	84.743	71.179	72.8959
4	66.7084	66.3329	88.0992	91.8544	75.9259	77.0349

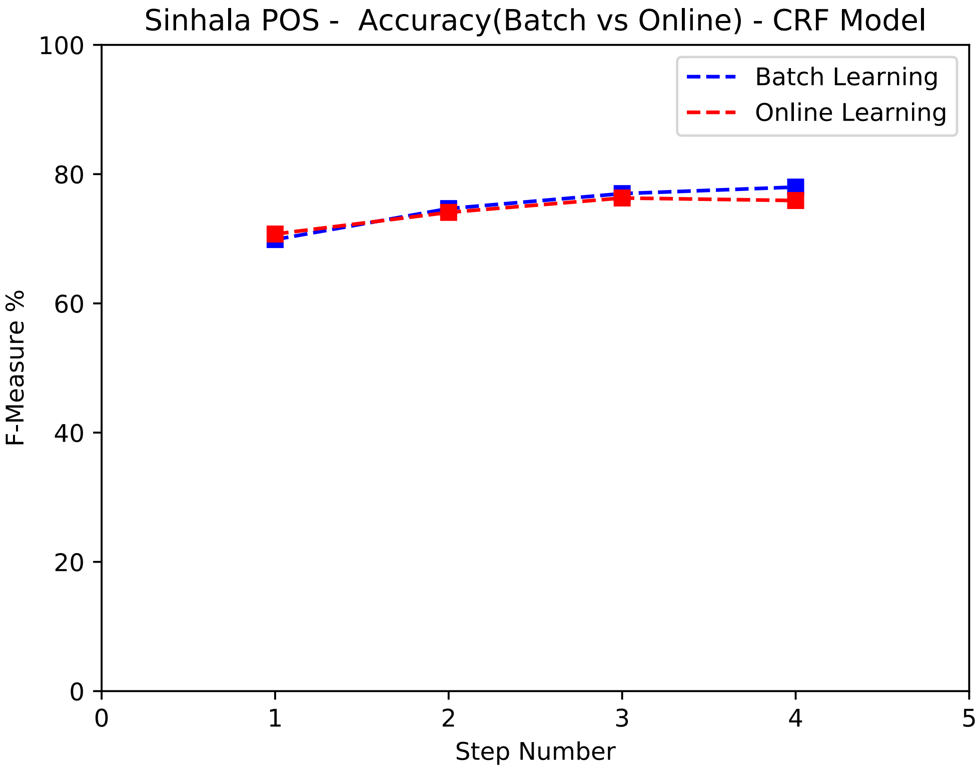


Figure 5.3 – Accuracy value variation of four steps using the CRF model - Sinhala POS

Table 5.2 – Accuracy value variation of four steps using the CRF model - Sinhala POS

Mini Batch	Accuracy %	
	Online	Batch
1	70.7307	69.8753
2	74.0802	74.6676
3	76.3063	76.9865
4	75.9147	77.9965

Table 5.3 – Precision, Recall, and F-Measure value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala NER

Mini Batch	Precision %		Recall %		F-Measure %	
	Online	Batch	Online	Batch	Online	Batch
1	40.9262	40.5507	69.8718	69.379	51.618	51.1848
2	52.3154	56.9462	79.9235	84.4156	63.2375	68.012
3	70.2128	69.9625	79.3494	82.8148	74.502	75.848
4	73.3417	74.5932	87.5934	86.5022	79.8365	80.1075

Precision values of the online learning model were closely related to the batch learning technique. Precision value of the first and third steps of the online learning technique gave a higher value than the batch learning technique. In other steps, batch learning had bit higher values. When examined the recall values, the first and fourth steps of the online learning technique, gave higher recall values and in other steps, batch learning technique had higher values. However, the recall values of online learning method have closely related to batch learning technique. In the first step, online learning had higher F-measure value, and for the rest, batch learning had higher values.

5.2.2.2 Sinhala-POS Tagging Experiment

This experiment observed the applicability of the bidirectional LSTM-CRF model into the Sinhala POS tagging. The accuracy variation of online and batch learn-

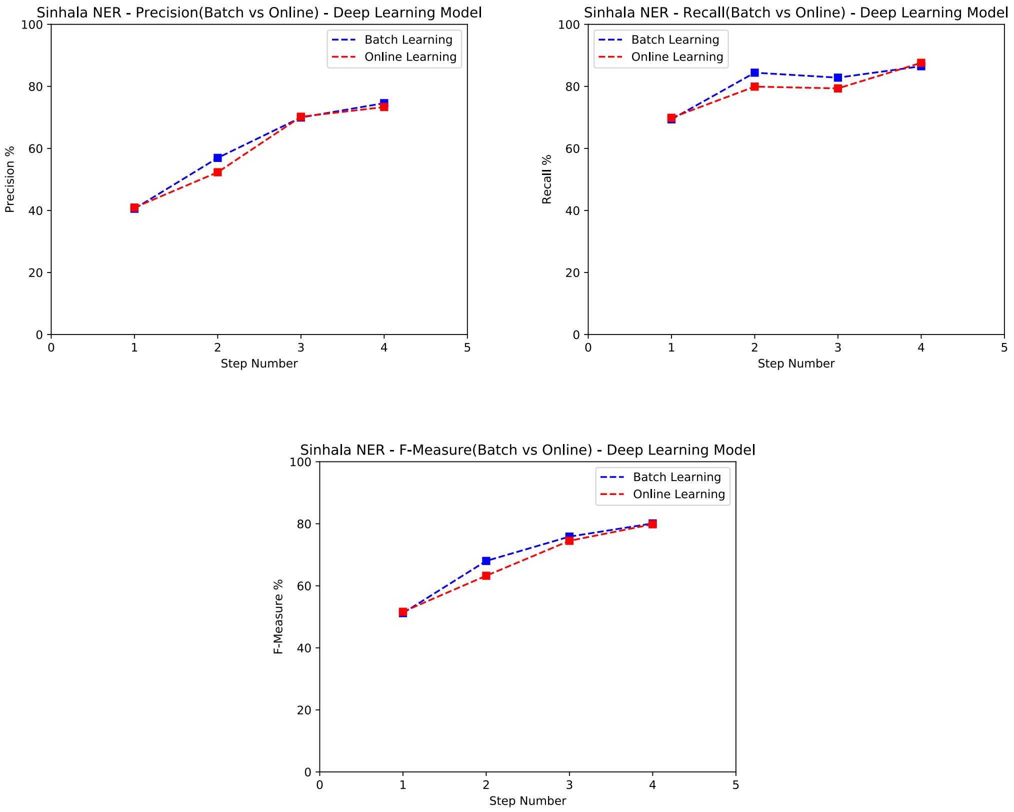


Figure 5.4 – Precision, Recall, and F-Measure value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala NER

ing strategies show in Figure 5.5. The numerical values of the accuracies include in Table 5.4. The accuracy values of the online learning and batch learning techniques are closely related in Figure 5.5. In the second and third steps, online learning gave higher accuracies, and in the other two steps, batch learning had higher accuracy values.

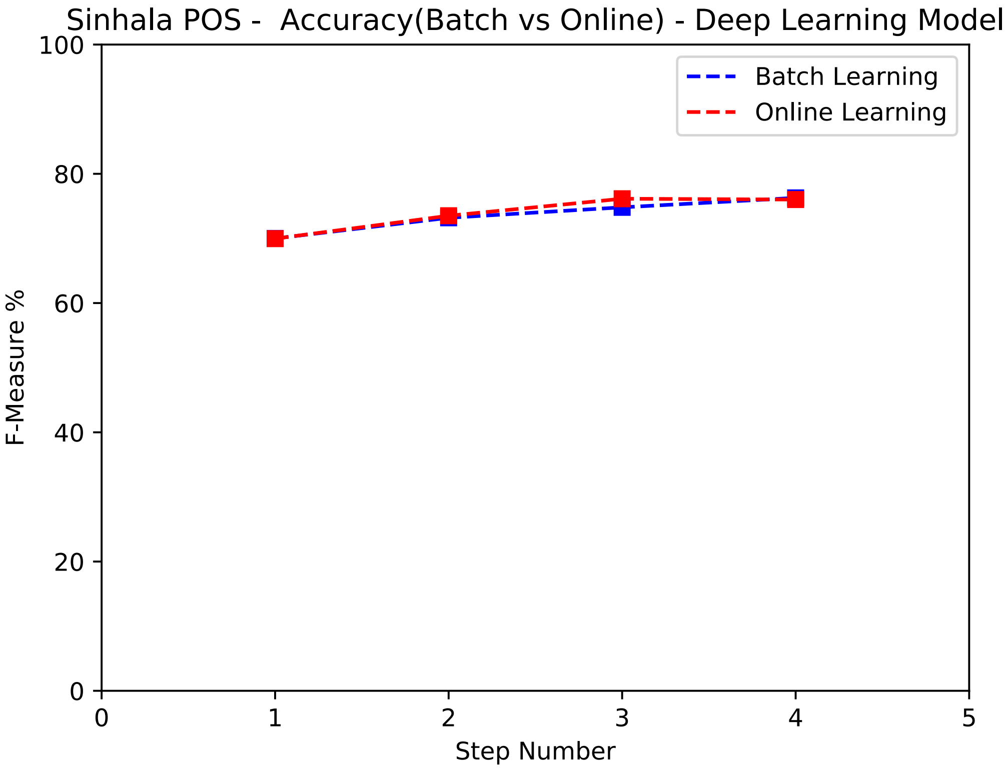


Figure 5.5 – Accuracy value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala POS

Table 5.4 – Accuracy value variation of four steps using the Bidirectional LSTM-CRF model - Sinhala POS

Mini Batch	Accuracy %	
	Online	Batch
1	69.9681	69.9989
2	73.5133	73.1939
3	76.1414	74.8119
4	76.0177	76.2754

5.3 Efficiency

This section observes the training time variation of two strategies: batch learning and online learning. Figure 5.6 shows the training time variation of the four experiments.

1. CRF Model
    - Sinhala-NER Experiment
    - Sinhala-POS Tagging Experiment
2. Deep Learning Model(Bidirectional LSTM CRF model)
    - Sinhala-NER Experiment
    - Sinhala-POS Tagging Experiment

As shown in Figure 5.6, 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.

5.4 Predictions

This experiment tries to predict the dataset size which gives the maximum performance from Sinhala NER and POS tagging tasks. As shown in Figure 5.1, the overall research consists of four experiments. Those experiments contain two NER tasks and two POS tagging tasks. In order to analyze the performance variation of those models, this research examines the central performance metrics of each model. For the NER predictions precision and recall can be used as the performance metric. However F-measure value variation used to the prediction because it is the single measurement which is a combination of precision and recall. For the POS tagging experiments, the accuracy value used as the performance metric.

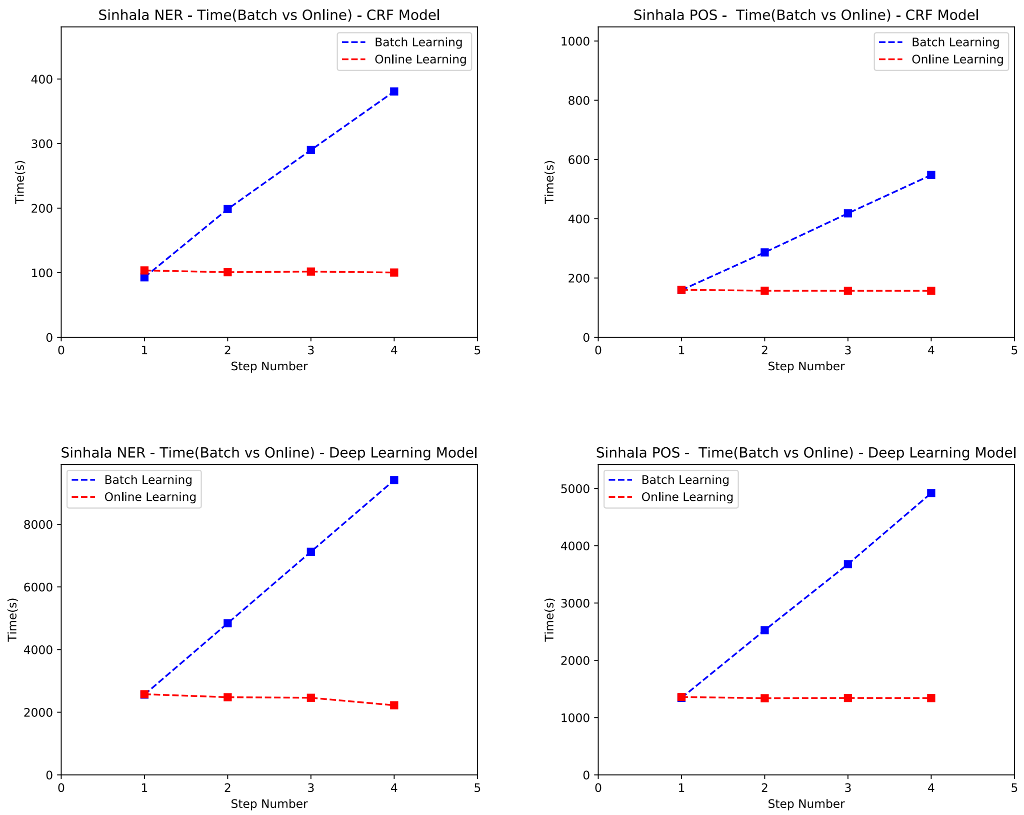


Figure 5.6 – Training time variation of the four experiments

The F-measure variations of the NER experiments in Figure 5.2 and Figure 5.4 follow a logarithmic variation. Hence, the variation of performance predicted using logarithmic regression. Table 5.5 shows the predictions of the NER. The first column of Table 5.5 contained the experiment name. The second column contained the approximated function using logarithmic regression. These logarithmic functions are defined under the range of (0-100) exclusively. The independent variable of these functions is the number of steps in the experiment. Each step in the Sinhala NER experiment, train using a separate data source (mini-batch) of 662 sentences. Thus the third column contained the number of data sources(a data source size is 662 sentences) need to obtain near maximum performance. The final column shows how many sentences need to obtain near maximum performance. In other words, the final column contained the multiplication of a number of data resources by the size of a single data resource (662 sentences).

The Sinhala NER currently has a dataset which has 3268 sentences. In order to obtain better performance from the CRF model, Sinhala NER needs 5296 sen-

Table 5.5 – 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

tences, and the bidirectional LSTM-CRF model requires 7282 sentences. Both experiments conclude that approximately it has to double the Sinhala NER dataset in order to obtain maximum performance.

Next step of the research tried to apply the same logarithmic regression prediction to the Sinhala POS tagging. Sinhala POS tagging had not large enough dataset in the training phase. Thus the incremental improvement was not shown in the POS tagging experiments. Hence, this dataset size predictions was not useful to apply for Sinhala POS tagging experiment.

5.5 Summary

This chapter mainly discussed the evaluation model of the research and the results of the final output. Section 5.1 discussed the evaluation plan of the research. Section 5.2 contained the final results of the two online learning model which implemented in this research. The comparison of the results of online learning methods with batch learning have included in section 5.2. Section 5.3 contained the observations of the running time of the online and batch learning techniques. Section 5.4 consisted of the results obtained to predict which amount of data size need to obtain maximum performance from proposed models.

Chapter 6

Conclusions

This chapter presents conclusions about the research problem and the research question. After that, the chapter examines the limitations of the research. Finally, the chapter ends with giving details about the further implications of this research.

6.1 Conclusions about Research Question

The main aim of this research was to find out a Named Entity Recognition (NER) model which can adapt and improve the NER model by using the data that comes at different time intervals. The proposed NER model should enhance the performance incrementally. This research proposed two online learning models to solve the research question: 1). Online Conditional Random Fields(CRF) model and 2). Bidirectional Long Short Tem Memory-Conditional Random Field(LSTM-CRF) model. The comparison of the proposed two approaches along with the batch learning models shows how these online learning techniques achieve accuracies closer to the batch learning techniques. Chapter 5 showed the incremental performance of proposed models, by using data sources which available as a stream.

The Sinhala NER experiment using Online CRF model improved the F-measure value from 31.4914% to 75.9259%. Using bidirectional LSTM-CRF model, the Sinhala NER improved the F-measure value from 51.6180% to 79.8365%. As described in chapter 5, the F-measure values of proposed models obtained the performance near to the batch learning techniques.

One of the main objectives was to find out the applicability of proposed models into POS tagging. Research design cast the NER problem as a structured prediction task. This study also considered the POS tagging task as a structured prediction task. After that, proposed models applied to the Sinhala POS tagging task. The obtained results, discussed in chapter 5. The Sinhala POS tagging using Online CRF model improved the accuracy from 70.7307% to 75.9147%. Using bidirectional LSTM-CRF model, Sinhala POS tagging improved the accuracy from 69.9681% to 76.0177%. The state-of-the-art Sinhala POS tagging approach proposed by Gunasekara et al. [13] have obtained 72% accuracy. In order to obtain that accuracy, Gunasekara et al. have used a hybrid approach which is a combination of rule-based one and an HMM model. Proposed Online CRF model obtained 75.9147% accuracy value, and the Bidirectional LSTM-CRF model have obtained 76.0177% accuracy value. The Online CRF model achieved **3.9147%** improvement from the state-of-the-art model. Further, the Bidirectional LSTM-CRF model achieved **4.0177%** improvement from the state-of-the-art model.

The NLP tasks have to adapt to the current context of the natural language. The proposed two approaches can enhance the performance incrementally using several data sources. Time-to-time the proposed models can train to most recent sources. Hence, the proposed two models of this research individually solve these problems of retraining and adaptation to the current context of the natural language. The proposed models did not use any of language-dependent features for the predictions. Hence the proposed online learning models have the capability to generalizing into various languages.

One of the main objectives of the research was to find out which amount of data need to obtain maximum performances from proposed models. Section 5.4 elaborated the results of this objective. For Sinhala NER, we have to approximately double the dataset in order to obtain maximum performance. However, this prediction failed to apply for the Sinhala POS tagging. Unlike the NER experiment, the POS tagging experiment incorporated with 22 tags. Therefore, the proposed models need a large data source to see the incremental performance enhancement in POS tagging. Since POS tagging task had a small data corpus, the gradual performance enhancement in those experiments was not depicted. Therefore, the previous prediction was unable to apply for POS tagging.

The Online CRF model can obtain better performances using a little amount of data. The bidirectional LSTM-CRF model can obtain much better performances than Online CRF model as shown in chapter 5. However, bidirectional LSTM-CRF model needs a larger data source to obtain higher performances.

6.2 Conclusions about Research Problem

Section 1.2, discussed the training-time overhead when retraining the same dataset, repetitively. Batch learning models should retrain from scratch when data sources unveil as a stream. Section 5.3 explained how the training time of the batch learning techniques increases linearly in each execution step. That section also described how the training time of the online learning algorithms remained nearly constant each training step. Section 5.2 showed how the accuracy of NER and POS tagging incrementally improved in each step using the proposed two online learning models.

6.3 Limitations

The main limitation that was the lack of annotated data to train proposed models. It takes more resources to build a data corpora with several mini-batches in practical scenarios. Thus, it was impractical to wait for actual mini-batches of annotated data. The mini-batches were created using existing data corpora. Then train the proposed models by using those mini-batches. The research experiments had a problem of finding a single Sinhala NER dataset which has sentence level categorization and annotated under the person, organization and location names. Instead, the IOB annotated dataset was used in the experiments. Further, the data sources used here should adhere to the CoNLL-2003 format.

The research experiments were started using a vocabulary which depicts the frequency distribution of the words. All the words in the sentences that input into proposed models have to be present in the vocabulary. Otherwise, the preprocessing step assigns a random index value to that word because the word index of that vocabulary considered as a feature. In practical scenarios, vocabulary can be obtained using raw text data without any annotation. When initializing the embedding layer of the machine learning models, the maximum sentence size has to specify in advance.

6.4 Implications for Further Research

The main practical NLP tasks are Information Extraction, Machine Translation, Automatic Summarization, and Information Retrieval. The NER and POS tagging are prerequisites for the above tasks. Proposed online learning based solutions solve the data availability problem in NER and POS tagging. Integrating these online learning based NER and POS tagging models with other main 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 like Information Extraction, Machine Translation, Automatic Summarization, and Information Retrieval into the online learning strategy. Thus converting that major NLP tasks into online and incremental learning is another future enhancement to this research.

Bibliography

[1] A.Attia and S.Dayan. "Global overview of Imitation Learning". In: (2018).

[2] Vinayak Athavale et al. "Towards Deep Learning in Hindi NER: An approach to tackle the Labelled Data Sparsity". In: *CoRR* abs/1610.09756 (2016). arXiv: 1610.09756. URL: http://arxiv.org/abs/1610.09756.

[3] Jason Brownlee. *Dropout Regularization in Deep Learning Models With Keras*. Aug. 2018. URL: machinelearningmastery.com/%20dropout-regularization-deep-learning-models-keras/.

[4] Jason Brownlee. *How to Use Word Embedding Layers for Deep Learning with Keras*. Aug. 2018. URL: machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/.

[5] Jason Brownlee. *Instability of Online Learning for Stateful LSTM for Time Series Forecasting*. Aug. 2018. URL: machinelearningmastery.com/%20instability-online-learning-stateful-lstm-time-series-forecasting/.

[6] Nikolay Burlutskiy et al. "An Investigation on Online Versus Batch Learning in Predicting User Behaviour". In: *Research and Development in Intelligent Systems XXXIII - Incorporating Applications and Innovations in Intelligent Systems XXIV. Proceedings of AI-2016, The Thirty-Sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, December 13-15, 2016*. 2016, pp. 135–149. DOI: 10.1007/978-3-319-47175-4_9. URL: https://doi.org/10.1007/978-3-319-47175-4%5C_9.

[7] Xavier Carreras, Lluis Màrquez, and Lluis Padró. "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*. CONLL '03. Edmonton, Canada: Association for

Computational Linguistics, 2003, pp. 156–159. DOI: 10.3115/1119176.1119198. URL: https://doi.org/10.3115/1119176.1119198.

[8] Jason P. C. Chiu and Eric Nichols. *Named Entity Recognition with Bidirectional LSTM-CNNs*. cite arxiv:1511.08308. 2015. URL: http://arxiv.org/abs/1511.08308.

[9] Gobinda G. Chowdhury. "Natural language processing". In: *Annual Review of Information Science and Technology* 37.1 (2003), pp. 51–89. DOI: 10.1002/aris.1440370103. URL: http://dx.doi.org/10.1002/aris.1440370103.

[10] J. K. Dahanayaka and A. R. Weerasinghe. "Named entity recognition for Sinhala language". In: *2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer)*. Dec. 2014, pp. 215–220. DOI: 10.1109/ICTER.2014.7083904.

[11] A Gepperth and B Hammer. "Incremental learning algorithms and applications". In: (2016).

[12] Kevin Gimpel, Dipanjan Das, and Noah A. Smith. "Distributed Asynchronous Online Learning for Natural Language Processing". In: *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. CoNLL '10. Uppsala, Sweden: Association for Computational Linguistics, 2010, pp. 213–222. ISBN: 978-1-932432-83-1. URL: http://dl.acm.org/citation.cfm?id=1870568.1870593.

[13] D. Gunasekara, W. V. Welgama, and A. R. Weerasinghe. "Hybrid Part of Speech tagger for Sinhala Language". In: *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*. Sept. 2016, pp. 41–48. DOI: 10.1109/ICTER.2016.7829897.

[14] H.Shah et al. "STUDY OF NAMED ENTITY RECOGNITION FOR INDIAN LANGUAGES". In: *International Journal of Information Sciences and Techniques (IJIST)* (2016).

[15] Zhiheng Huang, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF Models for Sequence Tagging." In: *CoRR* abs/1508.01991 (2015). URL: http://dblp.uni-trier.de/db/journals/corr/corr1508.html#HuangXY15.

[16] I.Augenstein, A.Vlachos, and D.Maynard. “Extracting Relations between Non-Standard Entities using Distant Supervision and Imitation Learning”. In: (2018).

[17] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. “Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data”. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML ’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289. ISBN: 1-55860-778-1. URL: http://dl.acm.org/citation.cfm?id=645530.655813.

[18] Eric Xihui Lin. *keras-contrib*. https://github.com/keras-team/keras-contrib/blob/master/examples/conll2000_chunking_crf.py. 2017.

[19] David Nadeau and Satoshi Sekine. “A survey of named entity recognition and classification”. In: *Linguisticae Investigationes* 30.1 (Jan. 2007). Publisher: John Benjamins Publishing Company, pp. 3–26. URL: www.ingentaconnect.com/content/jbp/li/2007/00000030/00000001/art00002.

[20] Christopher Olah. *Understanding LSTM Networks*. Aug. 2018. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[21] Adwait Ratnaparkhi. “A Simple Introduction to Maximum Entropy Models for Natural Language Processing”. In: *Institute for research in Cognitive Science* (1997).

[22] LIM PAO SEUN, Qiao Anna, and Zhang Zexuan. “HOW TO IMPLEMENT HIDDEN MARKOV CHAIN”. In: (2010).

[23] Erik F. Tjong Kim Sang and Fien De Meulder. “Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition”. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. CONLL ’03. Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 142–147. DOI: 10.3115/1119176.1119195. URL: https://doi.org/10.3115/1119176.1119195.