NAMED ENTITY BOUNDARY DETECTION FOR SINHALA

Yapa Hetti Pathirannahalage Prasan Priyadarshana

208073P

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree

Master of Science (Major Component Research)

Department of Information Technology

Faculty of Information Technology

University of Moratuwa

Sri Lanka

March 2022

Declaration

I declare that this is my own research thesis, and this thesis does not incorporate without acknowledgement any material previously published submitted for a Degree or Diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Signature: *UOM Verified Signature* Date: 18/03/2022

I have read the final thesis and it is in accordance with the approved university final thesis outline.

Signature of the supervisor: UOM Verified Signature Date: 18/03/2022

Acknowledgement

I would take this opportunity to express my sincere gratitude to everyone who guided me directly and indirectly to achieve this milestone. First and foremost, I am extremely grateful to my supervisor, Dr. Lochandaka Ranathunga for the continuous support, invaluable advice, kind guidance and patience during this whole journey. His vast experience and extensive knowledge have encouraged me throughout the time of my academic research and life. I also thank the research assistants who have been employed under the Social Media Analytics Research Group and the Accelerating Higher Education Expansion and Development (AHEAD) project, University of Moratuwa who helped me in annotating the dataset. I would also like to thank Dr. Surangika Ranathunga and all the research assistants who have been attached to the National Languages Processing Centre (NLPC), University of Moratuwa for providing me various corpus related to named entities and shaping the overall idea to identify the novelty. Finally, I would like to express my gratitude to my beloved parents, my wife and my friends for their motivation, inspiration, and continuous support throughout the difficult times.

Abstract

Named entity recognition (NER) can be introduced as a sequential categorizing task which contains a potential gravity in novel research arena. NER can be mentioned as the foundation for accomplishing most common natural language processing (NLP) tasks such as information extraction, information retrieval, semantic role labelling etc. Even though plenty of attempts have been employed on NE type detection, still there are plenty of avenues to be discovered under the NE boundary detection. Analyzing Sinhala related contents which have been published in social media can also be considered as one of the rising factors due to several human involvements in the recent past. The ultimate goal which is to obtain a constructive deep neural framework for determining named entity boundary detection has been achieved in a comprehensive manner and the model has been tested using Sinhala related statements which have been extracted through social media. Several objectives have been determined to accomplish this task considering the existing baselines. Several novelties have been identified to show off the uniqueness of this approach. Specifically, the novel concept *"Boundary Bubbles"* has been used to identify the specific entity mentions considering each head word for the identified named entities. Various experiments have been conducted based on multiple evaluation criteria and the named entity boundary detection model performs well with an average of 71% in Precision, 67% in Recall and 63% in F1 over the existing benchmarks. Hence this novel framework can be accepted as a vital solution for determining named entity boundary detection under forecasting various computational activities in social media.

Keywords:

Deep neural network, named entities, named entity boundary, named entity recognition, named entity type

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
NER	Named Entity Recognition
NE	Named Entities
NLP	Natural Language Processing
ORG, PER, LOC	Identified Some of the Named Entities
CNN	Convolutional Neural Networks
LSTM	Long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
GPU	Graphics Processing Unit
GRU	Gated Recurrent Units
Bi-GRU	Bidirectional Gated Recurrent Units
BERT	Bidirectional Encoder Representations from Transformers
MRC	Machine Reading Comprehensive
BOW	Bag of Words
RNN	Recurrent Neural Network
MLP	Multi-Layer Perceptron
CRF	Conditional Random Fields
NED	Named Entity Disambiguation
IE	Information Extraction
IR	Information Retrieval

Chapter 1
Introduction

1.1 Problem Domain

Named entity recognition (NER) has been recognized as a growing research area under natural language processing (NLP) [1]. Further, it can be introduced as a sequential labeling task which contains a potential gravity in novel research arena. The overall process of NER is to identify the textual spans and classify them considering specific pre-defined categories such as *Person, Organization* and *Location*. NER can be introduced as a mandatory task under Information Retrieval (IR) and Information Extraction (IE). Named Entity Type and Named Entity Boundary can be visualized as the two major components of Named Entities (NE).

On the basis of type aspect, NER can be divided into two groups such as coarse-grained NERs and fine-grained NERs. Fine-grained types focus on larger NE types while coarse-grained types have been recognized as the most prominent group for deriving niche NE types along with their respective boundaries. Some of those coarse-grained NER types would be *Person*, *Organization*, *Location* etc. On the other hand, deriving the relative NE boundaries would be much complicated for the Fine-grained types. On the basis of boundary aspect, another two groups can be generated such as Flat NER and Nested NER.

So far, many research attempts have been showcased to uncover the innermost of Flat named entities. Considering the most popular languages such as French, German, English, Chinese etc. Flat NER has already been addressed and many of such approaches have shown promising results in the recent past. Those approaches and frameworks have been presented in constructive manner under Chapter 2. On the other hand, nested NER can be introduced as a growing research area which has plenty of unreached milestones up to the pinnacle stage. Hence, plenty of researchers have kept their eyes towards this niche area. This is common for morphological rich languages such as Sinhala where there is very limited number of natural language resources

available and languages such as English where there is an abundant resource which have been developed under computational linguistics.

Nested named entity boundary detection is an unsmoked research area in which plenty of researchers have kept their eyes on. This can be further mentioned as one of the rising research areas of named entity recognition in Sinhala language. Mainly there are three main paradigms of named entity recognition such as knowledge-based unsupervised, feature based supervised and neural-based systems [2]. Most of the time the neural systems have been adapted due to the availability of latent features and the supportive of the sequential labeling. This is common for boundary detection as well where most of the time, the neural-based systems have been adapted in those mechanisms. In most cases, entity typing, and entity boundary detection have been considered as two sub tasks inside the name entity recognition main task. Additionally, interest for nested named entity detection has been grown up since decades yet most of the prevailing state-of-the-art baselines and models have been limited to one specific flat level at a specific time [3]. So, there is no doubt to mention nested named entity recognition as the main upcoming research area under named entity recognition.

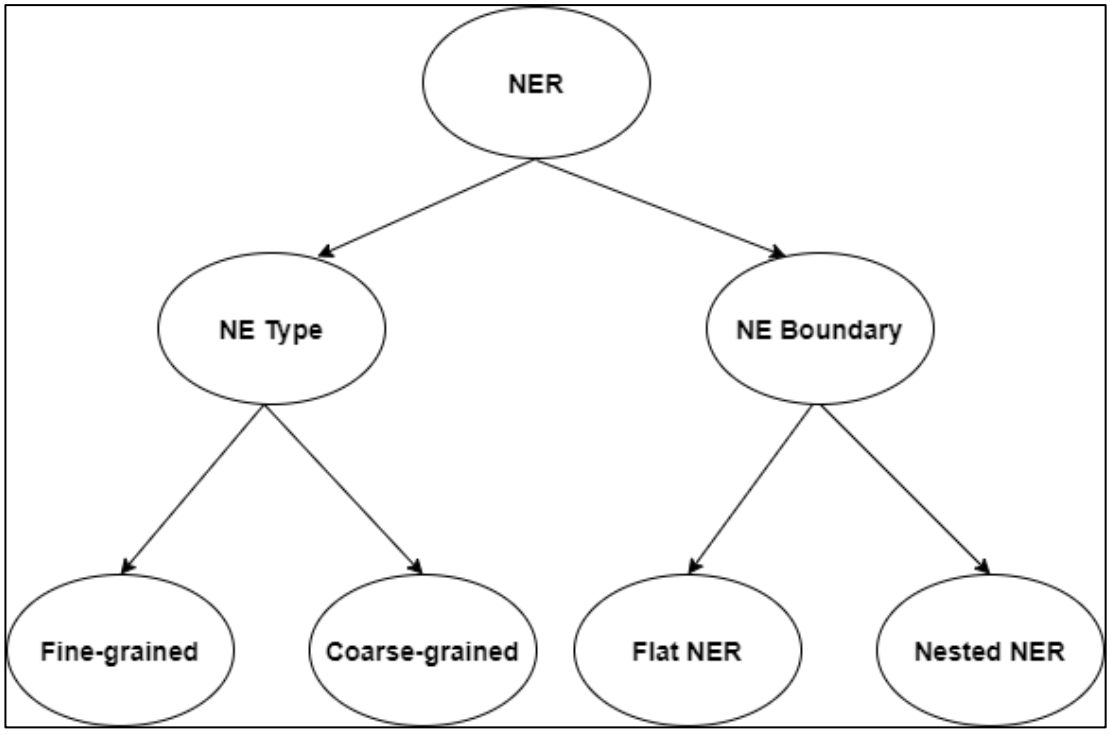

Figure 1.1. Hierarchy of NER

1.2 Background

Named entity boundary detection is worthwhile for some major NLP related activities such as sentiment analysis, information retrieval, information extraction and machine reading comprehension. So, it is really worthwhile of capturing this niche domain in terms of facilitating so many other aspects in computational linguistics. It is important to have a constructive scientific framework when managing the content which has been circulated in social media. Sinhala has been identified and defined as a morphological rich language under the category of Indo-Aryan languages [4]. It is spoken as the first language around 16 million set [5]. Even though Sinhala is a morphological rich language, still it can be categorized as a low NLP based resourceful languages [6]. Named entity boundary detection still can be recognized as an unknown area considering Sinhala language. In terms of manipulating the social media contents which have been published in Sinhala, it is really worthwhile to undercover these types of niche domains. There are so many rules and regulations have been established for maintaining the good governance of the social media for English contents but still there is a vacuum for such type of a constructive system for Sinhala.

1.3 Motivation

Targeting statements, expressions, disputes etc. in web-based analysis carries a huge demand in this field of business. Almost all the solutions which have been invented so far for analyzing social media content would be fallen into English language preferred approaches so that there is an enormous demand for a universal framework which provides facilities for low NLP resources languages such as Sinhala. The ultimate motivation of this research attempt is to develop such a framework for named entity boundary detection for social media analysis. Detecting both NE boundary and NE type for named entities as an aggregate mechanism will eventually tune up the accuracy and performance of NE linking to knowledge bases. This proposed novel framework allows to capture the named entity boundary detection along with their respective named entity types where the overall solution is capable of handling multiple domains including Sinhala.

1.4 Problem Statement

The usage of social media for various different kind of purposes has been increased in an alarming rate. There is a high demand for a sustainable computational framework for identifying and extracting such kind of various inputs which have been spread out in social media, especially in low NLP resources-based contexts like Sinhala. Identification of different kinds of named entities and named entity boundaries would generate a huge impact when extracting such contents. The ultimate research question is whether the proper identification of the named entities and their respective named entity boundaries would be a leverage in extracting the contents based on different scenarios, use cases and other related aspects which have been spread out in social media under Sinhala context so that it would lead to prevent social issues and other related subjective matters.

1.5 Research Aim

Application of a novel framework which is capable of detecting named entity boundaries along with the respective named entity types for Sinhala related posts, comments and statements which have been circulated in social media such as Facebook, Twitter, and YouTube. This framework can be further introduced as a specialized version for the domain Sinhala and is capable of enhancing up to a more generalized version considering multiple domains.

1.5 Research Objectives

In terms of achieving the above-mentioned research aim, several research objectives, which are the milestones of the overall study, have been listed down as follows.

Identify the existing mechanisms of NE type detection and NE boundary detection.

Through the accomplishment of this objective, it can be determined to discover on all the recently developed systems and tools considering the NE type and NE boundary detection. Considering the NE type detection, as it's already been explained in detail manner, the recently implemented systems on fined grained NER and course grained NER have to be examined. On the other hand, considering NE boundary detection, the latest research on flat NER and nested NER should be examined. Additionally, the

combined approaches where both the NE type and NE boundary detection have been considered as an aggregated manner, should also be examined. Since there are so many attempts have been made on these respective domains for different various languages, the approaches should be analyzed for different languages. As an example, most of the research are based on rich computational linguistics resources languages such as English, French, Chinese etc. Since the scope has been specified to Sinhala domain, it is a must to examine such systems as well since those are useful for setting up the fundamentals.

Identify the complexity, metrics, and relationships of statements in Sinhala.

Next important task is to examine and understand the structures of Sinhala related statements. Sinhala language is a sub section of Indo-Aryan language category, which can be mentioned as a rich morphological language but lack of the availability of computational linguistics tools and other logistics [7]. Due to that aspect, there are some complexity structures which have to be examined well before implementing something on top of it. According to the research problem, considering social media related statements, is something which should be analyzed considering the ground level rules of the Sinhala language. In terms of understanding such ground level rules, the complexity matrices and relationships of such matrices should be well determined. Such complexity indexes are varied from understanding the character level information to word level and even up to sentence level. As the second objective, such complex structures and matrices should be determined to discover the linguistic patterns of social media statements in Sinhala which have been spread-out in multiple platforms.

Implement a novel system for NE boundary detection considering NE types of the Sinhala related statements.

As it's been already discovered the novelty through the comprehensive literature review which is considering the NE boundary detection for the identified NEs under Sinhala social media statements, should have to be clearly defined. According to the comprehensive literature survey, it's been concluded that even though there are many attempts have been made for the sake of NE type detection for flat NEs, yet the scenario is completely different for capturing the boundaries. Considering the domain of

Sinhala, this particular segment has not even been touched in the research arena [8]. Also, most of the implemented systems consider NE type and NE boundary aspects as two separate components while an aggregated mechanism is demanded. That the ultimate goal of this research where to fill that overall gap. Still the scope has been narrowed of analyzing statements in social media for demonstration and respective experimental purposes but since the target is to come up with a deep neural model, the trained model should be capable of handling other avenues as well.

Implement an enhanced mechanism for Named Entity linking considering NE boundary detection.

NE linking or NE disambiguation is a rising research area which is important for information retrieval from the available knowledge bases. NE linking is a mechanism in which to map the identified NEs to a knowledge base. There are three main categories under NE linking namely direct mapping, indirect mapping, and disambiguation. Even though there are three main categories under NE linking, only direct mapping has been used as the NE linking mechanism up to date. This research will not go further and use indirect mapping or disambiguation in addition with direct mapping as a novelty but will be enhancing the direct mapping procedure. Through the novelty of NE boundary detection, the number of hidden entities which is going to be captured will be going up. Due to this reason, eventually the number of identified NEs will be going up. As a matter of fact, due to the enhancement which is expected to be delivered for NE boundary detection, the NE linking mechanism will be enhanced. The enhancement of the NE linking will be showcased using the direct mapping NE linking technique as the baseline.

Evaluate the implemented novel NE boundary detection framework and the enhanced NE linking procedure.

Conducting a major evaluation process can be identified as one of the topmost mandatory requirements of any research activity or procedure. The whole process of introducing a novel framework for NE boundary detection and enhancing the existing NE linking approach through the novel NE boundary detection should be properly evaluated using the existing benchmarks. Due to the shortage of some of the Sinhala

linguistic features such as missing some of the Sinhala based NEs, the major evaluations have been conducted considering the English domain but the whole idea is to showcase that the novel approach is universal and can be used with multiple domains including Sinhala. The whole evaluation procedure has been followed to demonstrate that the novelties have been proposed for satisfying the requirements in both existing and new domains. The respective benchmarks have been selected in a way to satisfying both the NE boundary detection process and NE linking process. This objective has been satisfied and demonstrated under the evaluation section.

1.6 Scope of the Study

The scope has been narrowed to perform the named entity boundary detection considering the shared posts, comments, and statements in social media under Sinhala domain for demonstration and respective experimental purposes. Hence, the developed deep neural framework can be introduced as a specialized framework in social media analysis. Due to the shortage of some of the Sinhala linguistic features such as missing some of the Sinhala NEs, the NE boundary detection and NE linking approaches have been evaluated under English domain, but the overall approach is applicable for multiple domains including Sinhala.

The next section elaborates a comprehensive analysis for the literature survey on NE boundary detection considering both nested and flat NER models.

Chapter 2

Literature Review

As it has already been mentioned, this overall research has been based on named entities. Before moving towards the literature review, it is better to focus on the two main categories of named entities such as flat NEs and nested NEs.

2.1 Introduction for Flat NEs and Nested NEs

Named Entity Type and Named Entity Boundary can be showcased as the two major components of Named Entities. Named entity boundary can be further subcategorized as Flat NEs and Nested NEs. Flat NEs are directly accessible and directly classified. Nested NEs can be introduced as a growing research area which has plenty of unreached milestones up to the pinnacle stage. Even though plenty of research have been conducted based on Flat NEs, the real usage of named entities can be demonstrated through Nested NEs. As an example, if *"University of Moratuwa"* has been considered, it can be classified as an *ORG* named entity in a much straightforward manner. But if you pay your attention closely, it can be concluded as an *ORG* named entity nested with *LOC* named entity. Hence, Nested NEs should be the more focusable segment under the category of named entity boundary.

2.2 Overview

There has been considerable amount of work done on named entity recognition and named entity boundary detection even though still there is not a proper constructive way of defending named entity boundary detection in a given context. The overall literature survey can be categorized into two major sub sections such as named entity boundary detection considering flat NEs and nested NEs and analyzing the statements which have been circulated in the social media platforms. The analysis of both categories has been done as follows.

2.3 NE Boundary Detection based on Flat NEs and Nested NEs

2.3.1 Deep Transfer Learning based approaches

Deep transfer learning can be adapted and used in some of the NLP related applications. Some of the deep transfer learning techniques have been adapted to identify the starting and ending tags of named entities [9]. This particular approach has been developed considering the morphological and orthological aspects of a given named entity. A neural network approach has been developed to overcome two issues of any fine-grained entity type classifications such as removing the burden of noisy training data and avoiding the usage of more hand-crafted features [10]. One of the objectives behind the usage of deep transfer learning is to filter the noisy feature data as much as possible. Although this deep transfer learning approach has shown some progression in NE type detection, the overall model was a failure when considering both NE types and NE boundaries together in a noisy environment [11]. So, still there is a high demand for a constructive approach for identifying the named entity boundaries.

2.3.2 Entity Linking based approaches

Several different types of named entity linking related NE type detection mechanisms have been introduced [12]. The respective lexical and morphosyntactic features have been generated through the named entity linking procedures for enhancing the performance of named entity recognition. The main concern of their approach is to determine an approach of classifying textual segmentations for a given list of categories such as person, location and organization considering the informal written style which is used in Twitter. The proposed system has been functioned with the NER baselines as a sequential labeling task. The overall process has been performed well with a f-score value of 61% for setting up baselines. Applying both NE type detection along with the respective boundary detection has not been followed.

2.3.3 Stack Pointer Networks based approaches

As a unique different approach, a stack pointer network has been introduced to detect named entity mention boundaries [2]. The stack pointer network has been used as the

respective tag decoder of the named entity recognition system rather than using the most common way of adapting a conditional random field (CRF) layer. Additionally, deep adversarial transfer learning has been used for sequence labelling, with the intention of using the gained knowledge transfer nature from known system to unknown one by using unannotated corpus [13]. By doing so, the authors have gained another advantage of getting rid of the need for having larger annotated corpus as per the training of testing purposes. Since the approach has been dominant for a corpus in which the named entity types have been defined considering the knowledge base, the entity type detection through the named entity recognizer, has been dropped out. The approach has been simply based on the introduced pointer network which was the tag decoder of the overall system. This same pointer network has been designed as the key segment to derive both the start and the end boundary of a given entity regardless of the type. Due to a limitation of the implemented system, this approach was incompetent in capturing both the NE boundary and designated type aspects. Hence still there is a demand for an aggregated system to handle both requirements.

2.3.4 LSTM, Bi-LSTM and Stack LSTM based approaches

Jason P.C et. al have introduced a novel deep learning based neural encoding approach on the basis of the OntoNotes 5.0 and CoNLL-2003 corpus [14]. The underline motivation of this combined process was to handle certain limitations and drawbacks such as avoiding issues of capturing the short distancing relations between the considering corpus and the inability of considering the explicit character level features in a deep neural way [15]. As per the proposing model, a new encoding scheme on the lexicon has been introduced while the main contribution was to introduce a hybrid novel model by combining a bi-directional LSTM model with a CNN to tackle both word-level and character-level features. The prime objective of introducing a CNN on top of the LSTM layer was to extract the character level features. However, during the matching process, only the exact matches within the same category or the entity type have been considered and the partial matching has been ignored. Due to the minor coverage of the inner entities, noisy environment and ambiguity, the proposed approach got affected on cheap performance in mapping entity mentions. This has also been affected for distinguishing the partial matching Vs. the exact matching in entity

linking. Even though the authors have attempted to improve the approach by introducing an improved convolutional neural network on top of Colbert et al [15] baselines, still there is a large room to be enhanced considering the boundary aspects of name entities. So, it can be concluded that there is a potential demand for a consolidated approach considering both exact and partial matching on top of named entity type and boundary detection.

A novel deep neural based model for named entity boundary detection which is called as BDRYBOT has been introduced [1]. Through this solution, specially the conflict situations on capturing the named entity mention types using on the basis on NERs and the respective entity types which have been linked through disambiguated approaches have been mapped. For the input representation phase, a CNN has been used instead of a Bi-LSTM considering the low computational cost. A sliding convolutional layer has been used as the enhancement in terms of capturing character n-grams in a given word. Furthermore, the input representational phase has been consisted with both the character level and word level component architecture. As the encoding phase, a Bi-GRU has been adapted instead of a Bi-LSTM due to the low computational consumption. As per the final phase of the proposed model, a unidirectional GRU hidden layer (hidden size 2H) has been used as the decoding phase. however, considering the whole approach, there is a clear gap when it comes to detect the entity boundary along with the impact on entity types. Another interesting point to be denoted is that this model has not been trained to detect timelines or time expressions such as entities like *"Monday"* won't be considered in a given context. This also can be considered as a future avenue. Another future avenue can be denoted as enhancing language potentials using named entity level information including boundary level and type.

Changmeng Zheng et al. have contributed for introducing the novel entity mention boundary aware solution to capture named entities by adapting a deep multitask learning [16]. This model can be taken as an aggregate approach where a layered sequential label model has been combined into an exhaustive regional classification model considering the inner and outer layers of nested named entities. By considering the detected named entity boundaries, the identified named entities would be further

classified into their respective categorical labels. This approach further utilizes specified boundary relevant regions in order to predict the most specific categorical label with the baselines of sequential classifications where it also has helped in minimizing the computational cost. They have overcome some of the major issues in nested named entity recognition such as extracting the inner and outer layers according to the most suitable extracting way, mitigating the error propagation and consideration of the explicit boundary information. By enhancing the sequence labelling model, it has been leveraged considering related boundary information in terms of locating the related entities. A single layered sequential labeling model has been applied for capturing the named entity boundaries with the assumption of the aspects of named entities could share the same named entity boundary labels. Additionally, due to the adaptability of the multitask learning, the dependencies of the named entity boundaries have been well captured. Although this deep neural model has been accepted as a first-time novelty for named entity boundary detection, only the implicit entity regions have been considered. In other words, this approach won't consider the explicit dependencies among the named entity regions. So, still there is a room for such avenues in introducing a novel approach by considering both the implicit and explicit dependencies among named entities.

Guillaume Lample et al. have contributed for named entity recognition domain by introducing a hybrid entity tagging architecture considering multiple neural models on transitional based stack LSTMs and bidirectional LSTMs [17]. The bidirectional LSTM has been designed in terms of performing tagging based on the scores which have been calculated per each token. This approach has been adapted the IOBES entity tagging scheme, which means a modification on IOB (Inside, Outside and Beginning) [18] format by adding annotations such as explicitly marking the end of entities which is E and the single entities by S. Even though the performance has been improved in IOBES scheme, this approach has been limited only to the explicit variations of the named entities without considering anything at all on implicit implications. On top of the ordinary LSTM CRF, another LSTM consisted with a stack pointer which is called as stack LSTM has been introduced as the architecture of the model for the purpose of computing the fixed dimensional embeddings. By this adaption, through the

dependency of stack LSTM model towards character-based representations, a competitive performance has been achieved. Still, even though this neural based model out forms the state-of-the-art existing baselines, there is still some room of development for entity boundary detection. Even though the entity boundary detection has not been enlightened through this approach, there is a proposal of enhancing the model further considering the hybrid nature to capture character-based entity boundaries.

An exhaustive Bi-LSTM approach for named entity recognition has been introduced by Sohrab and Miwa [19]. A Bi-LSTM approach has been adapted by combining the boundary representation of a region where employs the label of a particular region as the type of the entity. So, both entities mention type and boundary perspective has been taken into the account. Through this approach, a novel deep exhaustive model which represents all the possible entity mention regions on the basis of the entity length, has been developed on top of shared Bi-LSTM layer. Both the entity mention region and inside representations have been used for the task of boundary detection. In the methodology, the Bi-LSTM has been constructed in terms of accepting distributed word embeddings and related characters to generate the hidden sequences of vector representations. All possible related representations have been shared by applying the exhaustive combinations. The entity mention region has been represented by dividing an entity mention into respective inside representations and boundary related aspects. The corresponding output of the Bi-LSTM layer will be averaged for equality purposes. As the last phase, the output layer will be classified into the particular entity mention type through a softmax output layer. Even though the time complexity factor has been problematic for most of the approaches which have been dedicated for entity mention tasks [20] [21] [22], this exhaustive model has overcome such drawbacks. Still this overall model has been limited for dealing with inside entity mention representations, although the model is capable of outperforming most of the existing benchmarks.

The discussion for nested named entity mentions has come to the arena due to the usages of same named entity in different usages. Even though there are multiple approaches based on flat named entities and their usages, still the scenario is fresh for

nested named entities. A framework called Anchor-Region Networks has been introduced for considering the hard-drive contextual phase structures in nested named entities [23]. The basic assumption behind this solution has been set off as to nominate any nested entity mention along with any mention which has been considered apart from the head word which won't be shared along with any other mention. A novel neural network called anchor detector network has been employed to identify and extract the head entity mentions out of the nested entities while another neural network called region recognizer network has been dedicated for determining the named entity boundaries considering the extracted head named entities. Also, since the major datasets available on NER have not been annotated with such anchor or head words, a Bag Loss objective function has been adapted to train the Anchor Region neural network without considering the specific anchor word annotations. During the training process, the bag loss function can detect the most optimal anchor word for a selected entity mentions. For an example, if an instance such as, "*the department of education*", the bag loss function will eventually choose "*department*" as the head anchor word with the specific entity type ORG. There are two parts of the methodological approach for anchor region networks. First, the specific anchor words will be identified and classified to the specific entity type by the anchor detector network. As the second step, another neural network called region recognizer network will be applied to identify the whole entity mention centering at the particular identified anchor head mention. Both these neural models have been trained using the bag loss function. The anchor detector has been identified by obtaining the particular context aware representations using a Bi-LSTM layer. Then a probability score will be generated for the identified anchor words and later these probabilities will be normalized with the adaption of a softmax layer. A pointer-based mechanism has been applied to identify the entity boundaries of the identified anchor words. Apart from applying a Bi-LSTM layer, the local semantic features (such as a noun before a verb) also have been used to obtain the mention boundaries of the identified anchor words. Experiments have been carried out against three nested entity detection related benchmarks to show off the novel mechanism has been out formed the state-of-the art baselines. Considering the whole approach, there are so many assumptions have been made such as a particular mention can only be an anchor word in which the innermost named entity containing, where

such set of pre-defined rules won't let to cater most of the real-world scenarios. So, still there is a room for an improved version of this anchor region network.

Even though there is a long history for the usage of span-based mechanisms for capturing nested named entities, plenty of such approaches have been abandoned due to various computational issues such as expensiveness, low performance in explicit boundary detection etc. [24] [25] [26]. Apart from such previous approaches, Chuanqi Tan et al [27] have come up with a span-based model in terms of recognizing nested entity mentions through a classification of sentence subsequences. As per the methodology, this approach follows a multitask learning technique where after applying word level encoding with token-level representation, both the given span classifying model and mention boundary detecting model will be trained parallelly under multitask learning framework. During the first stage, BERT has been adapted for the word level embeddings and a Bi-LSTM layer has been used as per the character level embeddings. The aim of the second stage which is the boundary detection stage is to track whether a given word is either first or the last of an entity mention. A multilayer perception classifier and a SoftMax layer have been adapted to calculate the probability of a word being first or the last of a given entity mention. As per the third stage, the span classifier has been used with the aim of span classification to the specific semantic level tags. Experiments have been carried out against three main baselines and the experimental results show that this mechanism has out formed over the other main span related nested named entity detection mechanisms. Compared with the other major two approaches, hypergraph mechanisms and sequential labeling framework, still the span-based mechanisms lack in detecting named entity boundaries well so that this issue remains the same. So, still there is a room for such advancements as future avenues under this domain.

2.3.5 MRC based approaches

Assigning entity mention labels for both flat NER and nested NER on the basis of a sequential labelling approach cannot be considered as a practical approach. To address this limitation, another approach called Machine Reading Comprehensive or simply MRC has been introduced to tackle both the flat NERs and nested NERs. To date,

plenty of NER approached have been developed on the basis of sequential labelling theoretical approaches. This novel approach has been deviated from those where Xiaoya Li et al [28] have used a question and answering based MRC approach for accomplishing both flat and nested NER. Under this novel approach, sets of natural language related queries have been defined at first and the particular entity mentions will be extracted by analyzing the previously defined queries. The objective of adapting these so-called MRC models is to extract particular answer spans from a given context which includes the target entity with a starting and ending boundary limit. In other words, MRC models can be trained to extract the entity types and related boundary regions regardless of flat or nested on nature. The demand for such kind of an approach was the overwhelming trend of transforming the related NLP activities to the MRC models [29]. As the initial step, the data set construction process is there and each sentence will be categorized under three sets to discover the particular question, answer and the related context. Next step is to generate the respective questions and for accomplishing this task, an annotation guideline mechanism has been used. In terms of extracting the boundaries, BERT pretrained model [30] has been used as the foundation. The related probabilities will be calculated to determine the starting and ending index region matrices. Compared to the existing MRC model, this novel approach has performed well with respect to the main baseline flat and nested datasets. Still this mechanism would perform well with the coarse-grained NER type such as person, organization, location etc. which will be limited when generating the related questions. So, still there is a need for accounting with fine-grained NERs as well and yet this also can be even improved.

2.3.6 Regression based approaches

Yanping Chen et al. has introduced a multi objective framework, called as boundary regression model, which is based on the computer vision theories under the object detection [31]. According to their approach, such kind of structures can be divided into three main categories such as flattened, discontinues and nested. A concept called named entity bounding boxes has been developed where nested named entities detection has been done through the nature of overlapping in each bounding boxes. These bounding boxes have represented the named entity representation as an abstract

manner. In terms of object detection, a deep neural based approach called convolutional feature maps has been implemented. These maps are specialized and capable of identifying the discontinuous structures of the nested entity mentions and detect the respective length and the boundaries. This could be even mentioned something which has been adapted from computer vision where such techniques have been used recently for textual information retrieval from movies [32] and [33]. When evaluating the region boundaries, the theory of deep exhaustive models by Sohrab et al. [19] has been adapted. The bounding box regression model approach has been compared with the existing nested named entity recognition architectures and according to the experimental results this approach has overcome the other respective baselines in most of the cases. Although this overall framework has been introduced to demonstrate the respective named entity boundaries still it has been failed in capturing the respective NE type and the demand to develop a constructive framework for dealing with both NE types and NE boundaries remains the same.

2.4 Analysis of the Computational Linguistics Structures in Social Media Context

Even though Sinhala is a morphological rich language, very limited amount of NLP resources has been introduced considering both micro level and macro level tasks in computational linguistics. Hence, very minimum facilities can be observed for accomplishing major activities like named entity boundary detection for Sinhala context. As the very first task, analyzing the context is very important and those can be divided into sub sections as follows.

2.4.1 Complex Structures and Matrices in Sinhala Social Media Content

The next very important segment of this research is to discover the complexity matrices and parameters in Sinhala social media content. This overall research has been narrowed down to tackle with the analysis of the circulated posts, comments, and statements in social media. Focusing on the structures and complexity measures of Sinhala social media analysis can also be considered as a vital component. These matrices are relevant and important for identifying the entity boundaries of the identified named entities. When considering “මොරටුව විශ්වවිද්‍යාලයේ තොරතුරු තාක්ෂණ පීඨ සංගමයේ අභිනව සභාපතිවරයා ලෙස දිනිදු

සඳරුවන් මහතා තේරී පත් විය” , මොරවුව විශ්වවිද්‍යාලයේ තොරතුරු තාක්ෂණ පීඨ සංගමයේ අභිනව සභාපතිවරයා should be recognized as PER mention while තොරතුරු තාක්ෂණ පීඨ සංගමයේ or මොරවුව විශ්වවිද්‍යාලයේ තොරතුරු තාක්ෂණ පීඨ සංගමයේ will be fallen into ORG and මොරවුව should be recognized as LOC in the nested entity mention domain. When considering the spread-out Sinhala related social media content, there are various types of categories. Misbehavior usage of social media has reached to a position where it needs to be controlled by the authorities. There are some resent practical examples to show-off the extreme misbehavior of the social media. The pinnacle of this was the attack on 21st April 2019 where social media got banned for some time for the ordinary people. Even though the social media has been banned for some time, people found so many alternative ways of accessing them via using virtual private networks [34]. So, it is obvious that there should be a way of controlling the usage of such contents in social media in a more practical way.

2.4.2 Categories of Sinhala Social Media Analysis

Sinhala social media analysis can be classified into two major computational paradigms and those can be mentioned such as lexicon-based approaches and machine learning approaches [34]. A lexicon-based approach has been adjusted to analyze online forums, blog posts and comments which have been extracted from two data sources [35]. The first pool was consisted with published blog postings based on 10 popular web sites while the other one was consisted with blog post contents related to Israel-Palestinian disputes. Considering both of those categories, out of the available two different sets of corpora, 30% has been annotated as a preprocessing task which can be used for further analysis and next steps. This approach has been conducted with three main sections. As the initial phases, subjectivity detection has been carried out using the defined sentiment lexicon. In terms of determining whether the considering sentence is fallen into subjectivity or else, a publicly available Sentiwordnet [36] has been adapted. By extracting the particular negative or positive scores from the particular sentiment word, the respective score has been generated considering the positive and negative scores. Based on the calculated value, it was concluded whether

the sentence could be categorized into either positive or negative regions. As the second phase, Sinhala based social media lexicon has been constructed considering the nouns and verbs in the context. As per the first category, all the related verbs which have been categorized into offensive speech lexicon set, have been gathered. Then, related nouns have been collected on the basis of types such as religion, nationality, and race. A named entity recognizer has been used for identifying those particular nouns and added to the lexicon. As the final phase, due to the absence of a Sentiwordnet for Sinhala, the pre-build lexicon has to be used directly with the corpus to identify the related corpus. This can be identified as a restrictive limited approach since the whole process has been determined by the identified and build lexicon.

2.4.3 Machine Learning approaches in Social Media Analysis

On the other hand, as an alternative approach, machine learning has been used for Sinhala social media analysis in the recent past. Machine learning adaption for social media identification can be further categorized into two segments such as supervised learning approaches and unsupervised learning approaches. There is an ongoing usage of machine learning approaches to differentiate speech content from German dataset [37]. Due to the absence of a proper corpus, web scrapping mechanisms have been used. This approach has been pioneered with another previous approach where the related corpus was English. So, a type of transfer learning approach has been conducted to facilitate with the baselines. A Bag-of-Words (BOW) approach has been adapted to improve the overall process of extracting and identifying the social media driven contents regarding the differences between languages. Even though multiple machine learning approaches have been applied, a minimal work has been conducted on the basis of deep learning techniques due to the lack of resources and other related tools in supervised Sinhala natural language processing. This can be even worse when it comes to the specific domain of Sinhala social media analysis.

Feature extraction mechanism can be identified as one of the major important tasks in natural language analyzing and processing. There are several existing approaches in which the feature extraction methodology has been adapted in Sinhala natural language processing. Several algorithmic approaches have been applied considering the word

level and character level information which are essential for extracting feature level information in detecting social media content [38]. The idea of this approach was to determine the character and word level linguistic features could be able to determine promising results in terms of accuracy considering solo approaches namely Bag of Words (BOW). As expected, 78% accuracy gain has been obtained considering the character 4-gram feature on top of Support Vector Machine classifier. So, regardless of the language medium, a considerable accuracy level could be gained by applying mixture of different feature extraction techniques such as BOW, word n-gram, character n-gram and word skip gram linguistic features. Another work has been carried out to show off the applicability of the linguistic features of unigram, bigram, and trigram with the TF-IDF mechanism [39]. According to the experimental results, it has been concluded that the Logistic Regression performed well compared to other existing baselines. A recent approach has been conducted for social media analysis targeting the Indonesian language [40]. In this approach also, the word n-gram and character n-gram linguistic features have been applied as the fundamentals of the feature extraction process. The best model has been revealed when the word n-gram linguistic features have been applied on top of random forest (RFDT) classifier. Regardless of the language medium, language pre-processing techniques are essential similar to the importance of feature extraction techniques hence stop word removal and stemming should also be taken into consideration. The importance and relevant methodological steps for stemming and stop word removal will be discussed in detail manner under methodology. Considering these fundamentals, all of those have used macro-Sinhala linguistic features such as n-grams etc. There is a high demand for considering micro-Sinhala linguistic features such as POS tags, word counts, punctuation marks etc. A hybrid model for feature extraction considering both micro and macro-Sinhala linguistic features can be taken as a novelty considering the complexity measures and matrices in Sinhala social media analysis. Another future avenue in complexity measures and matrices in Sinhala social media analysis could be the consideration of both situations of stop word removal and without stop word removal. This comparison would be really handy and worthwhile since Sinhala language can be considered as a rich morphological language.

When it comes for social media analysis, there is a unique place for social media analysis related to religious conflicts. Religious conflicts and disputes have been ranked as the fifth under the top ten issues in current world according to the latest rankings by World Economic Forum. [41]. As it's already been explained above, there are so many existing methodologies focusing on various aspects when tackling with the religious unhealthy statements that has have been published in popular social media such as Facebook, Twitter, and YouTube. These statements are very important when it comes for shaping the people's mindset towards something. When it comes for the lack of local resources and annotators, crowd sourcing can be taken as a viable method of filling the gaps. Annotating large datasets through crowdsourcing has been practiced by so many scenarios and some of them have shown some considerable results as well. [42] and [43]. Some fuzzy techniques have been considered on the reliability of the annotators who have been employed for content annotations where the conclusion was the fuzzy logic would require the ground rules of better guidelines and definitions for optimal annotation procedure [44]. So far, we have concentrated more on the micro and macro feature extraction processes in linguistic features. Another handy approach under the linguistic feature extraction would be the availability of word embeddings. Word embedding has been nominated as one of the most prominent state of the art mechanisms in shaping textual oriented machine learning models [45]. Another approach has used word embedding along with some other pre-defined linguistic features in terms of training a regression model [43] while some latest approach has used word embedding, with character gram linguistic features for training a logistic regression model [46]. An ensemble classifier based deep neural model has been adapted in configuring multi-level classification to filter out offensive related content from a racist dataset [47]. Recently, a novel LSTM based deep neural recurrent model has been implemented and adapted in categorizing social media related contents for Italian domain [48].

Considering all of these attempts in both NE boundary detection based on flat NEs and nested NEs along with the respective NE types and analyzing different types of statements which have been circulated in social media context, it can be concluded that

a constructive scientific framework is essential to recognize the named entity boundary detection for the extracted comments, posts, and statements in the social media context.

The next section elaborates the way of the execution of the methodology, comparing with possible other alternatives and the uniqueness of the chosen methodological approach.

Chapter 3

Research Methodology

This section describes the methodological approach to determine the research objectives that have been stated above. The overall methodological approach can be categorized into two main components such as discovering the Sinhala related posts, comments, and statements which have been spread out in social media context through mandatory features, complexity measurements and other related aspects and constructing a novel framework for capturing named entity boundary detection considering the respective named entity type. These respective components can be demonstrated as follows.

3.1 Methodological Approach for Sinhala Social Media NE Boundary Detection

The ultimate goal is to turn up with a neural based methodology to discover and analyze the Sinhala related social media context analysis based on a supervised approach. Since, as per the comprehensive literature review, it can be concluded that there is an absence of a proper system to conduct NE boundary detection by extracting the posts, comments, and statements in Sinhala domain. Considering all aspects, data construction has been conducted as initial step of the entire methodological approach.

3.1.1 Data Construction

This is the initial phase of the construction of Sinhala social media related corpus. Since an ultimate neural model should be developed based on supervised learning mechanism, the availability of a proper constructive dataset is essential. A considerable amount of data collection is required to construct a valid deep neural model. The validity will be totally depending on the correlation between the considering context along with the nature of the dataset. The main bottleneck is the unavailability of an online corpus for Sinhala named entity boundary detection purposes. Considering languages such as English, French, Arabic, or some Indo-Aryan languages such as Hindi too are consisted with available online corpus for such social media analysis.

First of all, sets of posts, comments or statuses which have been circulate on three (3) social media platforms have been collected. Those three platforms are such as

Facebook, Twitter, and YouTube. In terms of accomplishing the web crawling purposes, well-established crawling programs have been adapted. Once the statements have been crawled, those have been collected to a pool to be distributed among multiple set of annotators who are responsible for conducting manual annotations. This is because the whole approach has been designed as per a supervised learning approach. Before the annotation procedure, each annotator has been given with a constructive training on the procedure of identifying the respective content as offensive category or not. Each annotators outcome will be accepted as unique and based on the majority vote per each collected statement, the category would be decided. As per the categorization, each statement would be separated into two groups such as offensive or neutral based on the outcome. More than 10,000 social media statements have been crawled for this purpose. This sample has been used as the implementation dataset for achieving the later steps.

3.1.2 Preprocessing

Once the annotated statements have been constructed, then the next sets of preprocessing tasks will be determined. The following mandatory tasks can be listed down as removing HTML elements, removing stop words, removing special characters, removing non word entries, and stemming.

Stop words can be mentioned as most common characters or words which don't carry a weight for computational linguistics. In other words, those can be further identified as useless when it comes for natural language processing tasks. Even though stop words are useless for NLP related tasks, those are essential for maintaining the grammatical relationships and the structure of a sentence. When it comes for morphological rich languages such as Sinhala, such stop words are critical for specially tasks such as sentiment analysis. Due to that reason, even though plenty of NLP tasks are commenced with removing stop words, this approach would consider the stop words as well at some point. This approach will conduct a comparison to evaluate the two conditions of with considering stop words and without considering stop words at some point. In terms of dropping the dimensionality of the feature vectors, the extracted stop words have been used. A bag of stop words has been employed to accomplish removing the stop words. These bag of stop words have been extracted

from the public repository[1] in which contains more than 4000 stop words. Set of bags of stop words from the extracted list can be listed as follows.

Table 3.1 : BAG OF EXTRACTED STOP WORDS

ඛවට	වෙතට
මෙන්	වශයෙන්
සේ	මෙම
විට	අනුව
මෙලෙස	තව
ලෙස	සහ

Then as per the following step, removing special characters, removing HTML elements, and removing non word entries can be showcased. These special characters also should be removed before the training purposes since those are also can be recognized as useless. In order to remove the special characters, a unique regular expression statement has been used. The extracted characters will be collected to a list in which will be known as "*Cleaned List*". Comparing with the stop words list, even though there can be an impact for morphological rich language from the stop words, the special elements and characters cannot be considered as such so that those can be removed entirely.

$$\sum_{n=1}^{k} (\prime\backslash@\backslash w\backslash!\backslash*\backslash\#\backslash\$\backslash\%\backslash\&amp\backslash[0-9]\prime) \tag{3.1}$$

Next important phase is stemming. Stemming is to derive the base or root form of any word. In terms of extracting the root form, the added prefixes and/or suffixes have to be stripped off. Stemming can be further described as a core functionality under the boundary regions identification of entity mention nuggets. This is really useful for

[1] Stop words list has been used from the public repository under https://github.com/nlpcuom/Sinhala-Stopword-list/blob/master/stop%20words.txt

boundary detection of Sinhala sentence nuggets. The stemming lets to extract the base of a word helps to reduce the feature vector dimensionality as well. Since Sinhala is a low NLP resource language, there is not a proper stemming tool such as Porter's stemming kit for English. For that reason, in this research, it has been used a developed shallow tool for stemming purposes for especially Indo-Aryan languages. Since the implementation has not been available public, it has been modified and implemented for the fulfillment of this research. Some of the famous stems and their related variations which have been drew out from the dataset can be tabulated as follows.

Table 3.2 : STEMS AND VARIATIONS

Stem	Words
තස්ත්‍රවාදීන්	තස්ත්‍රවාදීන්ගෙන්, තස්ත්‍රවාදීන්ට, තස්ත්‍රවාදීන්ගේ
විද්‍යාලය	විද්‍යාලයේ, විද්‍යාලයට, විද්‍යාලයෙන්
කොල්ලා	කොල්ලෙක්, කොල්ලාට, කොල්ලොන්ට

These extracted stems will be saved for another list which will be used for feature extraction in next steps. After this process, the pre-processing tasks have been determined and the corpus is ready for the feature extraction procedure.

3.1.3 Feature Engineering

Several features have been identified and can be listed down as word n-gram features, bag of words (BOW) and character n-gram features. These extracted features have been fed to the proposed deep neural based model at the early phases. In the BOW approach, set of identified lists of named entity related words will be kept in a file to be cross checked with the corpus. For this purpose, Google offensive word list has been used [49]. In terms of facilitating for Sinhala, Google offensive word list for English will be translated and used. This can be identified as a famous approach and has been used in plenty of previous related approaches which have been critically analyzed before [49]. On the other hand, word n-gram and character n-gram features capture the structure or dimension of a sentence. N-gram features can be categorized into different classes such as unigram, bigram, trigram or even combination of the

mentioned three variations. Since the similar social media analysis approaches based on other languages have shown promising results with the adaption of gram features in word level or character level, it would increase the accuracy considering such gram features apart from using only BOW features.

When it comes for inputs such as social media related statements, it is essential to detect the patterns where character n-grams features provide that facility. Even for scenarios such as detecting misspelled words, it is required to extract the n-gram features. So, the character-based n-gram features can be mentioned as a really useful factor in feature extraction for social media related named entity identification.

3.1.4 Parts-Of-Speech (POS) Tagging

When considering the computation linguistics feature extraction, there are two main components of features namely micro features and macro features. Word n-grams and character n-gram features can be recognized as macro features. This is high time to consider the micro features as well. parts-of-speech (POS) tagging can be showcased as the best example for micro level feature engineering. POS tagging is an important sub task. When the NEs have been tagged using POS tagging approach, each NE would be considered as a proper noun. In that case, POS tagging, and the related POS tag set would really be interesting to have. Even though the natural language toolkit (NLTK) Python package has provided 17 languages for POS tagging and other POS tag sets, Sinhala has not been listed among the 17 languages. It is strongly believed that the POS tagging, and the related POS tag sets are important factors as micro level linguistic features in Sinhala named entity detection. So, as per the implementation purpose, a unique tagging mechanism has been implemented targeting Sinhala. The respective pseudocode will showcase the POS tagging technique for Sinhala context.

POS Tagging

Input: SOV extracts (subject, object, verb patterns)

NP: Noun Phrase

VP: Verb Phrase

SO list

Output: POS tag output

```
    do NP:
    {<NNPI>*<JJ>*<NNN>*<NNN><NNPA.*>*<NNPI.*>*<PRP.*>*}
    VP:
    {<JVB>*<NVB>*<V.*>*}
  if in NP or VP or NP ^ VP
    While add to SO list
  return SO list
```

Figure 3.1. Pseudocode for POS tagging

3.1.5 Define the Neural Model for Sinhala NE Identification

The final step under the initial section of the methodological approach is defining the deep neural model. Since the approach is to follow and adapt a deep neural based model from the project inception stage, it is important to discuss the structure, respective type, and the required levels of hyper-parameter values. As the deep neural model, a recurrent neural network (RNN) has been proposed due to various reasons. Out of most deep neural models. RNN can be introduced as the most dedicated model to capture the relationships, hidden patterns, and other related factors for linguistic features. RNN has shown promising results under linguistic domain from the recent past for various use cases and scenarios, so that this has been decided as the type of the model.

In RNN also there are two different types and for this purpose, long short-term memory (LSTM) type has been selected due to its linguistic applicability. LSTM is well known for encoding long sequences of linguistic related features and compared to other RNN types, LSTM is unique in behaving independently for evaluating the most recent inputs. As per the hyper parameters, number of 50 embedding layers have been used for smoothening the embedding procedure, Dense layer has been used as the most primary initial sets of nodes, "*relu*" has been chosen as the activation function, a dropout layer has been proposed for mitigating the overfitting issues, a dense layer has been selected as the output layer and "*sigmoid*" function has been used as an activation layer for smoothen the output layer. The model has been developed using a dedicated PC which consists with high GPU processing power for the training purposes. The

experimental results of the training procedure will be employed as per the confusion matrix values and those will be explained later.

3.2 Methodological Approach for Deriving Named Entity Boundary Detection

Once the initial methodological step is ready by identifying and classifying the named entity identification for Sinhala, then the second phase which is determining the NE boundary detection can be initiated. Here, several steps have to be followed in terms of successful execution of the flow. Those dedicated steps will be discussed as the next steps. First of all, the proposed architecture of the entire procedure for deriving named entity boundary detection can be showcased as a high-level prototype as follows.

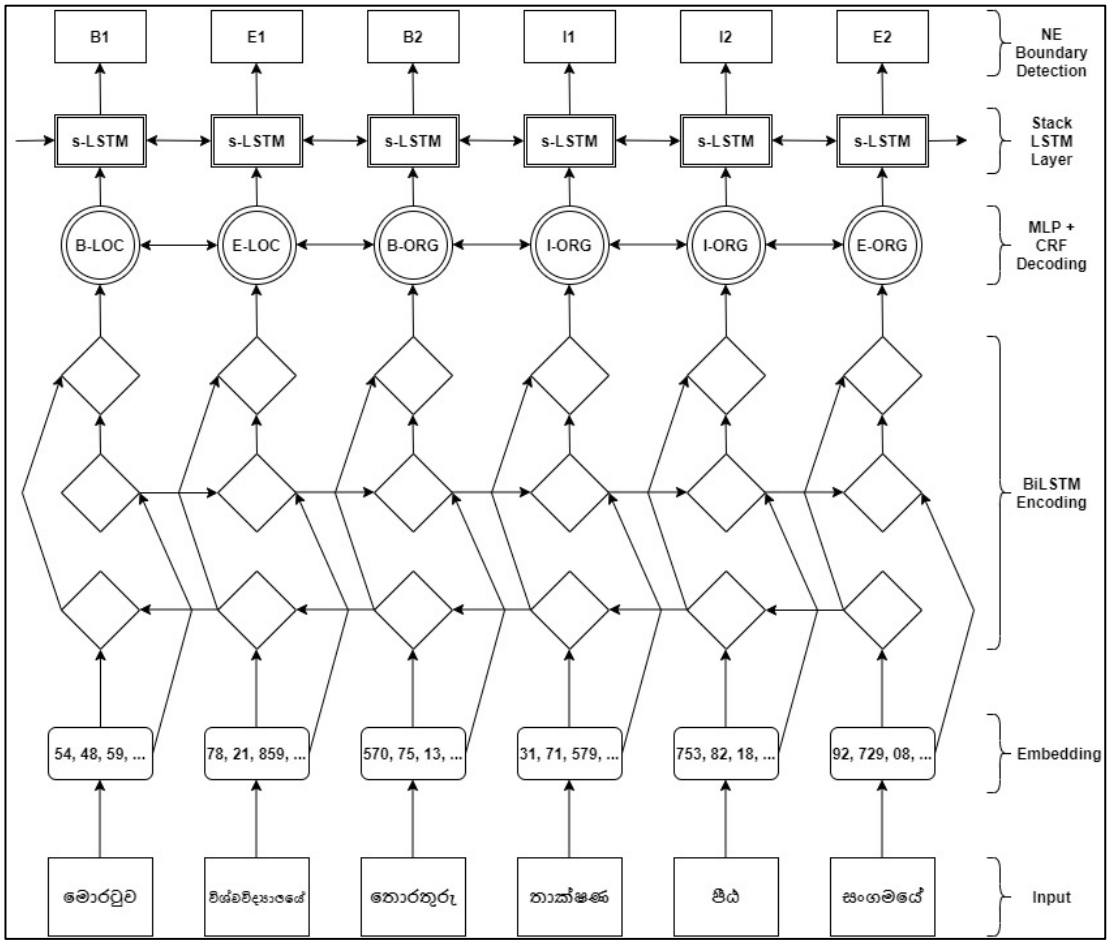

Figure 3.2. Proposed architecture for named entity boundary detection

Each of the major components such as Word Embedding, Bi-LSTM encoding, MLP+CRF decoding and Stack LSTM Layer which have been showcased as per the above prototype can be described as per the following sub sections.

3.2.1 Word Embedding Procedure

As per the word representation, word level embedding, and character level embedding have been followed. Character level embeddings set the phase for deriving of capturing the flat NER task and word level and character level mixture embedding would be led for facilitating the nested NER task. Since the embedding process is at a basic stage for Sinhala, Glove word embedding which has been developed based on English corpus will be used as the platform. In terms of transforming the obtained embedding mechanism for the Sinhala context, a constructive deep transfer learning methodology has been adapted. The character level embedding details have been determined as the initial step. Then, the obtained embedding values have been fed to a Bi-LSTM layer in terms of generating the backward and forward value sets. Finally, the word level embeddings have been determined by combining the above mentioned two value sets in a comprehensive computational way. Since the system architectures which have been determined on the next steps are in common nature and settings, deep transfer learning can be adapted to gain promising results within minor adjustments.

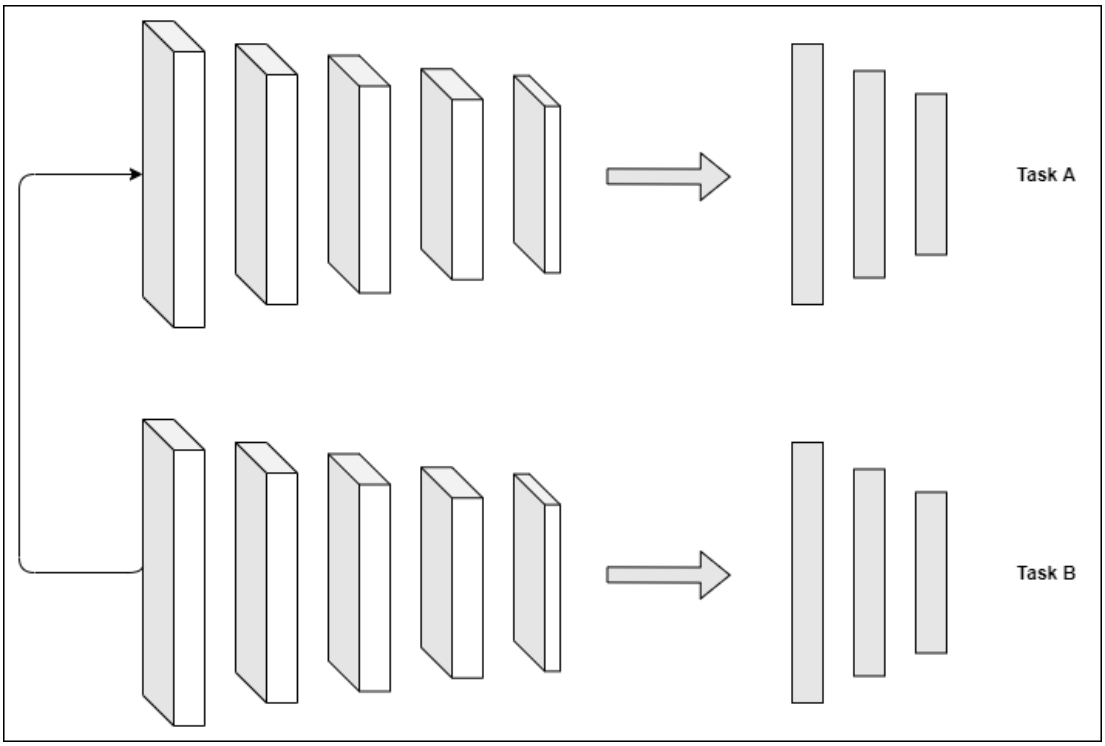

Figure 3.3. Deep Transfer Learning process

Usage of deep transfer learning-based approaches are common in nowadays where some advanced neural based models have been introduced rather than designing something from the scratch or by applying any available neural model in the market directly. So, most of the hyperparameters and other settings which have been acquired from the parent architectural model will not get changed and would be directly applied in the new paradigm. These are essential for detecting the respective NE type and the boundary scope details in the next steps.

3.2.2 Head Word Mention Detection using MLP and CRF along with NE Type

Special LSTM based deep neural model has been developed as the overall NE detection end to end framework. A Bi-LSTM layer would be used to derive the word level classifiers to obtain the respective head word which acts as the dominant words of the pre-defined context. As an example, considering English context, "*Mr. Silva, The President of Sri Lanka*", where *President* can be denoted as the master word in terms of acting as the dominant keywords of the given sentence nugget. Even though there are two NEs such as *Silva* denotes as a PER mention and *Sri Lanka* denotes as a LOC related mention, *Silva* acts as the main or the head word and *Sri Lanka* acts as the named entity mention. Considering the whole nugget, *Mr. Silva, The President of Sri Lanka*, it acts as a PER type mention as a whole. So, the first step is to determine the head word of each unique sentence nugget.

As the overall procedure, all of the mentions in a given sentence have been mapped in a subsequent manner of word mention representations such as n_1, n_2, n_3, … n_i in terms of deriving the respective word embedding, POS embedding and character embedding.

$$N = n_1, n_2, n_3, \ldots, n_i \tag{3.2}$$

Then adapting a Bi-LSTM layer, related contextual representations for each mention have been obtained. Those contextual level representations have been fed into a structured classifier called multi-layer perceptron (MLP) along with the conditional random fields (CRF). MLP is capable in nature to derive the inner representations of the word level contextual representations. This can be further identified as a feed forward neural network consists with three types of layers such as input layer, output layer and hidden layer. A combined classifier has been used to improve the level of

accuracy and this combined mechanism can be mentioned as a unique approach which has not been used for any of the previous NE boundary detection frameworks. CRF layer is also dedicated for determining the respective NE type of the head word. So, the usage of CRF on top of MLP layer would be a win-win situation.

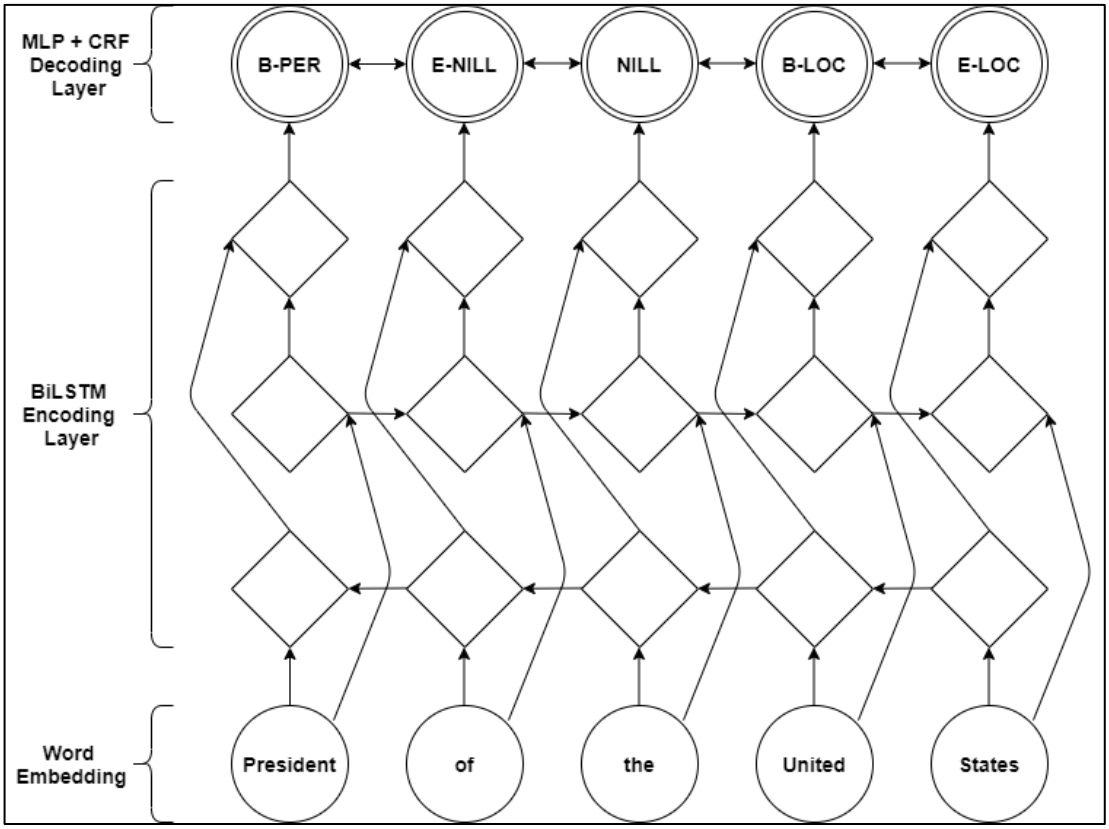

Figure 3.4. Main system architecture of NE type detection

Once the related NE types have been determined, a *SoftMax* layer has been used in terms of normalizing the generated scores to related probabilities. Here, throughout the process, there is one assumption which has been followed such as, even though there can be multiple sentence nuggets per each sentence, a particular head word won't be shared among more than one sentence nuggets. Sharing a particular head word among multiple sentence nuggets has been kept as per future work.

3.2.3 Identifying Entity Mention Nuggets for the captured Head Words

Once the respective head words have been derived, the next important step is to locate the specific mention nuggets per each dominant word. Here a novel unique theory called *boundary bubbles* has been introduced. Boundary bubbles can be demonstrated

as abstract representations for visualizing the sentence mention nuggets along with the respective identified head words.

$$[[The\ \boldsymbol{President}]_{DES} of\ the\ [\boldsymbol{United\ States}]_{LOC}]_{PER}$$

$$[[\text{මොරටුව}]_{LOC}\ \text{විශ්වවිද්‍යාලයේ}]_{ORG}[\text{නොරතුරු තාක්ෂණ පීඨ සංගමයේ}]_{ORG}$$

Figure 3.5. Abstract representation of boundary bubbles

A specific algorithm has been developed as per capturing the mention nuggets. In terms of developing the algorithm, several assumptions have to be made considering some mandatory consequences.

1. A mention can be considered as the dominant word of the mention nugget if and only if the same word could have been considered as the head word of the underlying innermost entity.
2. It is a must to avoid considering most functioning words of a sentence (*of, a, the*) as head words where the whole process would lead to a failure by introducing a remarkable noise.

Considering the above-mentioned assumptions several points have been set off to define the underlying algorithm of *boundary bubbles*.

1. A particular word can be denoted as a head word of the considered innermost entity mention of it where words under a named entity mention would not be considered as a head word of the respective outermost mentions.
2. When considering all the identified entity mentions, at least one word should be determined as the head of the mention nugget.
3. Considering the *boundary bubbles*, at least one word in a *bubble* will be considered as per its head word.

Considering the above-mentioned assumptions, the specific algorithm for identifying mention nuggets can be determined. In terms of identifying the mention nuggets, respective left boundary region and right boundary region of each identified head word should be determined. Both of the parameters should be considered as a tuple of $y =$

(y_i, y_j, y_k, c_i), where $y_j, \ldots, y_k$ would be the respective two aspects of left boundary and right boundary region notations. Here the loss function has been notified as the respective head word detection loss function. If y_i will not be considered as a head word, the respective loss value could be determined as head word detector loss and will be assigned as *NIL*. The collective loss would be generated as the weighted sum value of the respective situations where the head word detector would be set off into *NIL*. The mention nuggets identifier can be obtained and written as:

$$\tau(x_i; \emptyset) = v_i.[-logP(c_i|x_i) + L^R(x_i; \emptyset)] + (1 - v_i).[-logP(NIL|x_i)] \quad (3.3)$$

Here, $-logP(c_i|x_i)$ can be identified as the mention head detection loss. The weighted loss value which would consider the left boundary region and right boundary region of each identified head word along with the region $x_i; \emptyset$ can be determined as:

$$L^R(x_i; \emptyset) = L^{left}(x_i; \emptyset) + L^{right}(x_i; \emptyset) \quad (3.4)$$

This would result the proper mention nuggets measuring the two boundary ends where per each right boundary and left boundary centered detected head word. Still, the marginal loss also should be focused where it could be an impact for the overall procedure.

Besides, v_i under equation 3.3, determines the correlation between the entity mention x_i for the respective type of *bubble* c_i. Considering the other words under the same *boundary bubble*, an entity mention should be consisted with higher v_i if there would be a strong association with the respective *bubble* type. This is an important measurement with respect of obtaining the respective mention nuggets which are surrounded by the respective identified head words. If the concept has been developed with the intention of obtaining multiple head words per mention nugget, this whole process could have been jeopardized. Considering all these assumptions, v_i can be further determined as:

$$v_i = \left[\frac{P(c_i|x_i)}{max_{x_t \in B_i} P(c_i|x_t)}\right] \alpha^v \quad (3.5)$$

Where B_i determines the respective *bubble* x_i belonging to all the identified mention nuggets which share the most common innermost head word with x_i. α can be denoted as a special hyper-parameter, in terms of manipulating the way of a mention could be accepted as a head word rather than considered as *NIL*. The most basic scenario of $\alpha = 0$ could determine the special case where all the captured mentions could be annotated with the respective *bubble* type. Along with that, $\alpha \rightarrow +\infty$ would determine the head word detection loss could evaluate the mention with the highest $P(c_i|x_i)$ as the respective head word, while the rest of all other mentions under the same *bubble* would be labelled as *NIL*. These measurements would derive the whole process until reaching out for the particular entity mention nuggets targeting the respective captured head words. Altogether, head word detection loss value determines that at least one head word (considering the highest $P(c_i|x_i)$), has been determined for each *boundary bubble* in the next phase. For the rest of the mentions which have not been linked with the respective *bubble* type, head word detection loss could result as *NIL* in an automatic way during the training procedure.

There can be situations such as overlapping of abstract representations but those can be sorted out adapting to the earlier assumption of having one head word per sentence nugget. This is just to get a scope of the identified mention nuggets where the actual region classification under NE boundary detection would be determined through the next step. The importance of this step is to reduce the computational complexity which can be occurred during the training procedure. The respective experimental results under various hyper-parameter assignments will be demonstrated later under experiments.

3.2.4 NE Boundary Regional Classification using Stack LSTM

The final segment of the whole process of named entity boundary detection and type determination will be the boundary region classification. According to the constructive literature survey, there are various approaches have been applied to accomplish the region classification for flat NEs. But those mechanisms cannot be used for determining the region classification for nested NEs due to the incompatibility. According to the region scopes which have been determined through the boundary bubbles from the previous step, the region classification will be fulfilled. For obtaining

the respective boundary regions, a novel scope allocation mechanism called IOBE (inside, outside, beginning, end) has been adapted. In terms of facilitating the training purposes, a labeled mention dataset along with the IOBE scoping pattern has been used. IOB scoping mechanism has already been used in some of the existing solutions but the usage of IOBE pattern together with stack pointer networks can be denoted as a unique first-time approach.

Similar to the previous step, a specialized Bi-LSTM has been applied here for the sake of determining the regions for NE boundary detection. This is completely a novel mechanism which deviates from pure Bi-LSTM approaches and LSTM-CRF models. Multiple enhanced algorithms have been developed for achieving the competitive edge compared to the existing baselines for capturing the NE boundary detection. First of all, in terms of focusing on the character-based representations, h_r^R of a word x_i have to be determined. In terms of capturing the innermost semantics, a convolutional matrix has been applied upon h_r^R :

$$e_i = \tanh\left(W h_{r-j:i+j}^R + d\right) \tag{3.6}$$

Where $h_{r-j:i+j}^R$ would be the aggregation of vectors from h_{r-j}^R to h_{r+j}^R , W and d are the representations of the convolutional matrix of kernel aspect and bias terminology, respectively. The overall architecture can be showcased as follows.

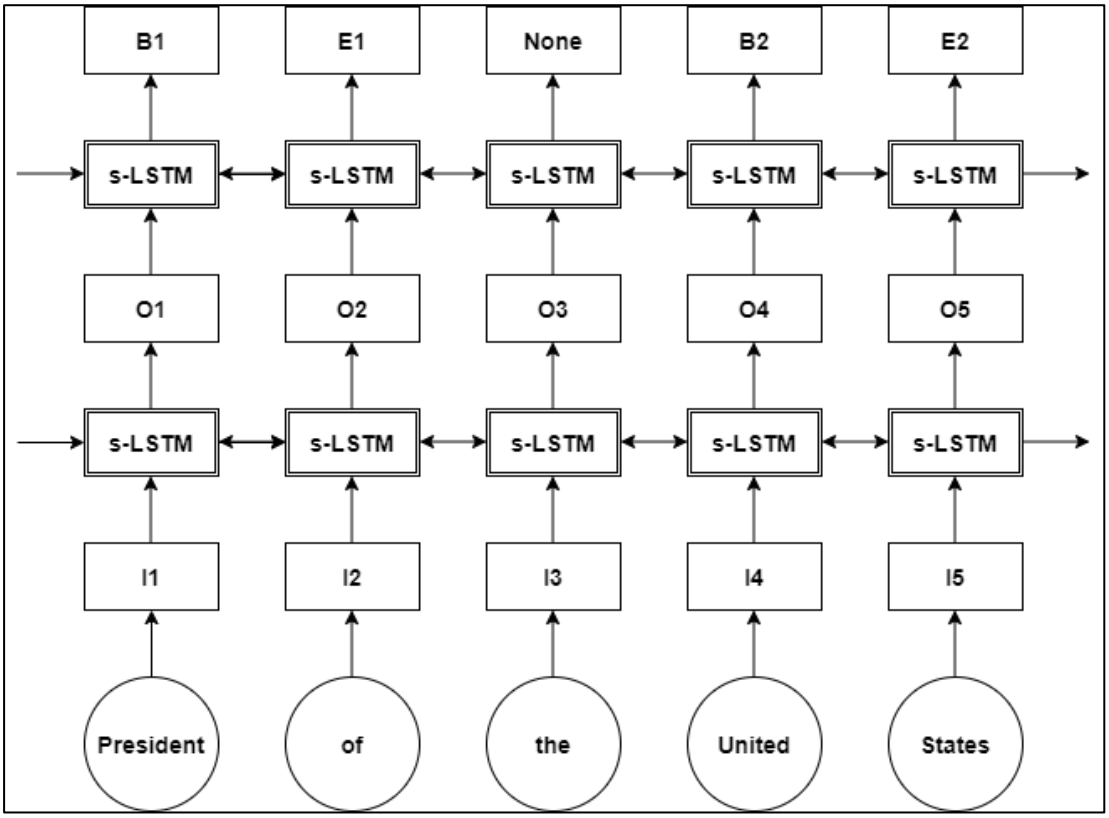

Figure 3.6. Stack LSTM architecture of NE boundary detection

Finally, for each of the underlying head word x_i the left entity mentions boundary value score L_{ij} and the respective other right entity mentions boundary value score R_{ij} would be determined as:

$$L_{ij} = \tanh\left(r_j^T \omega_1 h_{i+j}^R + G_1 r_j + b_{i+j}\right)$$
$$R_{ij} = \tanh\left(r_j^T \omega_2 h_{i+j}^R + G_2 r_j + b_{i+j}\right) \quad (3.7)$$

Where ω derives the correlation of the collection of mentions h_{i+j}^R, G and b represent special hyper-parameters for deriving respective boundary value scores respectively.

Considering the above mentioned two functional equations, the first option would compute the regional classification score of mention x_j considering as left boundary region and right boundary region centering x_i. Second equation would model the probability of x_j itself performing as the universal boundary region. Afterwards, the

novel stack LSTM layer would be generating the best fitting left mention boundary region for x_j and the respective right mention boundary region for x_i head word.

Stack LSTM, as which has already been mentioned, is something totally different and unique compared with the nominal bidirectional LSTM model or LSTM-CRF model which has been used for previous steps. Stack data structure can be mentioned as the target architecture under this step. In terms of achieving the competitive edge, stack LSTM has been used considering stack LSTM's main focus on character-based representations. During the process, stack LSTM architecture accepts available corpus as one by one word for the processing.

The implemented stack LSTM model consists with main three layers along with 150 dimensions for the considering each stack. This model can be further explained as a greedy model in terms of facilitating optimal actions in terms of processing the entire sentence. There are several hyper-parameter settings which have been set off to compute the overall functionality of the underlying model. Some of the main hyper-parameters can be named as *detection_learning_rate, boundary_learning_rate, weight_decay_round, hidden_dim, output_dim,max_margin, train_detection_in_joint, word_dict_file, pos_dict_file* etc. All the required composition functions and embedding dimensions have been determined considering the most optimal values. Those value generations, comparisons and evaluations will be described in detail under the experimental results and evaluations sections.

3.2.5 Enhancing the NE Linking

NE linking can be identified as a vital task in information retrieval from knowledge bases. This can be further explained as an aggregate task which has to considered along with NE detection. There are three main methodological levels of NE linking such as direct mapping, indirect mapping, and disambiguation. Even though NE linking is going to be considered in this research, a novelty will not be achieved through NE linking process. Instead of that, the plan is to enhance the direct mapping.

NE linking procedure through enhancing the NE detection procedure would be obtained. The respective direct mapping relation algorithm which has already been proposed for flat NE detection in knowledge bases would also be used for nested NEs

as well for accomplishing the aggregated NE linking process. The respective direct mapping algorithm can be mentioned as:

$$p(c_{ij}|e_i) = \frac{t(e_i, c_{ij}) + 1}{\#e_i + |C_i|} \tag{3.8}$$

Where $p(c_{ij}|e_i)$ determines the probability of the context distribution of c_{ij} given on top of the entity mention e_i. This has been formulated as the number of occurrences on the entity mention e_i which has been linked with the corresponding context. The possibility of the probability becoming zero has been avoided by adapting the Laplace smoothing. The whole process has been divided into four main steps such as implementing the glosses, building the context, implementing the vector representations, and building the concept ranking via direct entity linking.

The next chapter elaborates the overall implementation of the main components which have been identified under the comprehensive methodological section. The programming languages, tools, frameworks, libraries, and other mandatory components which are required for carrying out the overall implementation will be described in detail manner.

Chapter 4
Implementation

This section describes the way of implementation to fulfill the methodological approach for determining the above-mentioned research objectives. The implementation of the overall project has been carried out considering the main sections which have already been mentioned and described comprehensively under research methodology section. Those main sections would be determining the whole implementation phase and those key aspects would be described under this section. Considering all these aspects, the whole implementation phase can be separated into few components and those can be listed down as follows.

- Implementation on Sinhala NE identification.
- Implementation on deep transfer learning for word representation.
- Implementation on NE mention head word detection.
- Implementation on the entity mention nuggets identifier and the head detection loss function.
- Implementation on the NE boundary detection regional classification.
- Implementation on the NE linking.

4.1 Technology Stack

The entire implementation has been conducted using Python as the programming language and Python 3.7 has been used as the respective Python version which has been adapted. The reason behind choosing Python 3.7 is that version 3.7 can be mentioned as the most stable version of Python under the releases of Python 3 series, and it provides the most inbuilt support and capability for the usage of the mandatory frameworks such as TensorFlow 1.15 which has been heavily used as the core framework of this entire implementation. Apart from that, few other important Python libraries dedicated for accomplishing core computational linguistics purposes such as Torch 1.7, Spacy 2.1.9, NLTK 3.4.5, Allennlp 0.9.0, Keras 2.2.4 etc. have been used for fulfilling the implementation purposes.

This whole process would eventually cover the implementation of the entire product. As per the description under research methodology section, the novelty has been reflected under some of the parts. Now, let's focus on each implementation section which has already been mentioned above. The respective algorithmic implementation will be represented in terms of pseudocodes as follows. Pseudogen[2], has been used as the special tool for converting the Python coding implementation to the respective pseudocode.

4.2 Implementation on Sinhala NE Identification

According to the methodological procedure for Sinhala named entity identification, the respective Python implementation has been followed and the related pseudocode can be represented as follows.

Algorithm 4.1: NE Identification

```
FOR max_len IN max_len_arr:
   SET startTime TO time.time()
   SET tok TO Tokenizer(num_words=max_words)
   tok.fit_on_texts(X_train)
   SET train_sequences TO tok.texts_to_sequences(X_train)
   SET train_sequences_matrix TO sequence.pad_sequences(train_sequences,
maxlen=max_len)
   DEFINE FUNCTION RNN():
      SET INPUTs TO Input(name='INPUTs', shape=[max_len])
      SET layer TO Embedding(max_words, 50,
INPUT_length=max_len)(INPUTs)
      SET layer TO LSTM(10)(layer)
      SET layer TO Dense(256, name='FC1')(layer)
      SET layer TO Activation('relu')(layer)
      SET layer TO Dropout(0.5)(layer)
      SET layer TO Dense(1, name='out_layer')(layer)
```

[2] Pseudogen is a SMT-based pseudo-code generator (https://ahcweb01.naist.jp/pseudogen/)

```
    SET layer TO Activation('sigmoid')(layer)
    SET model TO Model(INPUTs=INPUTs, outputs=layer)
    RETURN model
```

According to the respective methodological approach, a novel RNN structure has been defined to train the named entity identification flow. An enhanced algorithm has been applied to detect the Sinhala NE identification. As usual, a pretrained word embedding model has been applied. The deep neural network has been consisted with different hidden layers and optimal conditions under the hyper-parameters. "*Dense*" layer has been adapted as the introductory layer which governs the whole neural layer. "*Relu*" and "*Sigmoid*" have been adapted as the core activation functions. "*Dropout*" layer has been used to avoid the over fitting issues which can be occurred during the training process. As the next algorithmic implementation, implementation on deep transfer learning for word representation can be showcased.

4.3 Implementation on Deep Transfer Learning for Word Representation

Here the word representation mainly focusses on the word and character embeddings. A pretrained model which has been developed using Glove word vectorization approach has been used to accomplish the purpose. The relevant core algorithm and the optimal hyper-parameter settings can be listed down as follows.

Algorithm 4.2: Character and Word Embedding

```
DEFINE FUNCTION transform(self, X, y=None):
  SET word_ids TO [self._word_vocab.doc2id(doc) FOR doc IN X]
  SET word_ids TO pad_sequences(word_ids, padding='post')
  IF self._use_char:
    SET char_ids TO [[self._char_vocab.doc2id(w) FOR w IN doc] FOR doc IN X]
    SET char_ids TO pad_nested_sequences(char_ids)
    SET features TO [word_ids, char_ids]
  ELSE:
    SET features TO word_ids
```

```
    IF y is not None:
        SET y TO [self._label_vocab.doc2id(doc) FOR doc IN y]
        SET y TO pad_sequences(y, padding='post')
        SET y TO to_categorical(y, self.label_size).astype(int)
        SET y TO y IF len(y.shape) EQUALS 3 else np.expand_dims(y, axis=0)
        RETURN features, y
    ELSE:
        RETURN features
```

Special deep transformer architecture has been adapted for accomplishing the transfer learning objectives where the approach can be mentioned as a first-time adaption under Sinhala context. Considering the hyper-parameter settings and coding tune-ups, the relevant algorithmic implementation can be showcased as follows.

Algorithm 4.3: Transfer Learning Tune-ups

```
DEFINE FUNCTION __init__(self, lower=True, num_norm=True,
            use_char=True, initial_vocab=None):
    SET self._num_norm TO num_norm
    SET self._use_char TO use_char
    SET self._word_vocab TO Vocabulary(lower=lower)
    SET self._char_vocab TO Vocabulary(lower=False)
    SET self._label_vocab TO Vocabulary(lower=False, unk_token=False)
    IF initial_vocab:
        self._word_vocab.add_documents([initial_vocab])
        self._char_vocab.add_documents(initial_vocab)
DEFINE FUNCTION tranform(self, X, y):
    self._word_vocab.add_documents(X)
    self._label_vocab.add_documents(y)
    IF self._use_char:
        FOR doc IN X:
```

```
        self._char_vocab.add_documents(doc)
    self._word_vocab.build()
    self._char_vocab.build()
    self._label_vocab.build()
    RETURN self
```

Next main aspect is to focus on the implementation on NE mention head word detection.

4.4 Implementation on NE Mention Head Word Detection

Entity mention head word detection plays one of the core roles here in this entire solution. For fulfilling this task, as it has already been described under the methodological approach, a novel, special Bidirectional Long Short-Term Memory (LSTM) neural approach has been invented as the overall NE detection end to end framework. The Bi-LSTM model is dedicated for determining the word level-based classifiers to obtain the head word which acts as the head/dominant words of the considering contextual domain. The novelty is the adaption of MLP on top of the famous decoder, CRF. As it has already been mentioned, this attempt can be determined as a first-time approach in determining the NE mention head word detection for any domain. The respective pseudocode implementation can be showcased as follows.

Algorithm 4.4: NE mention head word detection

```
super(BiLSTM).__init__()
    SET self._char_embedding_dim TO char_embedding_dim
    SET self._word_embedding_dim TO word_embedding_dim
    SET self._char_lstm_size TO char_lstm_size
    SET self._word_lstm_size TO word_lstm_size
    SET self._char_vocab_size TO char_vocab_size
    SET self._word_vocab_size TO word_vocab_size
    SET self._fc_dim TO fc_dim
```

```
  SET self._dropout TO dropout
  SET self._use_char TO use_char
  SET self._use_crf TO use_crf
  SET self._embeddings TO embeddings
  SET self._num_labels TO num_labels
  DEFINE FUNCTION build(self):
    SET word_ids TO Input(batch_shape=(None, None), dtype='int32',
name='word_INPUT')
    SET INPUTs TO [word_ids]
    IF self._embeddings is None:
      SET word_embeddings TO
Embedding(INPUT_dim=self._word_vocab_size,
                    output_dim=self._word_embedding_dim,
                    mask_zero=True,
                    name='word_embedding')(word_ids)
    ELSE:
      SET word_embeddings TO
Embedding(INPUT_dim=self._embeddings.shape[0],
                    output_dim=self._embeddings.shape[1],
                    mask_zero=True,
                    weights=[self._embeddings],
                    name='word_embedding')(word_ids)
    IF self._use_mlp:
      SET char_ids TO Input(batch_shape=(None, None, None),
dtype='int32', name='char_INPUT')
      INPUTs.append(char_ids)
      SET char_embeddings TO
Embedding(INPUT_dim=self._char_vocab_size,
                    output_dim=self._char_embedding_dim,
                    mask_zero=True,
```

```
                name='char_embedding')(char_ids)
        SET char_embeddings TO
TimeDistributed(Bidirectional(LSTM(self._char_lstm_size)))(char_embedding
s)
        SET word_embeddings TO Concatenate()([word_embeddings,
char_embeddings])
    SET word_embeddings TO Dropout(self._dropout)(word_embeddings)
    SET z TO Bidirectional(LSTM(units=self._word_lstm_size,
RETURN_sequences=True))(word_embeddings)
    SET z TO Dense(self._fc_dim, activation='tanh')(z)
    IF self._use_crf:
      SET crf TO CRF(self._num_labels, sparse_target=False)
      SET loss TO crf.loss_function
      SET pred TO crf(z)
    ELSE:
      SET loss TO 'categorical_crossentropy'
      SET pred TO Dense(self._num_labels, activation='softmax')(z)
    SET model TO Model(INPUTs=INPUTs, outputs=pred)
```

According to the comprehensive literature survey, none of the previous attempts have focused on the combinational approach of MLP and CRF. Plenty of approaches have been driven considering the CRF as the main tag decoder. MLP has displayed some promising results for transfer driven neural based solutions. The combinational attempt of MLP and CRF has not been used to fulfill the NE detection or NE boundary detection scenarios. So, this first-time approach can be mentioned as a pure novelty in this whole approach. The experimental results which are relevant for these implementations will be discussed later in detail manner.

Next implementation component is the implementation on the entity mention nuggets identifier and the head detection loss function.

4.5 Implementation on the Mention Nuggets Identifier and The Head Detection Loss Function

This is the implementation phase regarding the main concept which drives the whole theorem of this approach, "*boundary bubbles*." The implementation can be divided into two main components such as implementation on entity mention nuggets and implementation on the head detection loss function. First, the implementation on capturing entity mention nuggets can be displayed as follows.

Algorithm 4.5: Capturing entity mention nuggets

```
FOR key IN train_config:
   SET self.__dict__[key] TO train_config[key]
SET self.word_dict TO WordDictionary()
self.word_dict.restore(self.word_dict_file)
SET self.pos_dict TO WordDictionary()
self.pos_dict.restore(self.pos_dict_file)
SET self.char_dict TO CharDictionary()
self.char_dict.restore(self.char_dict_file)
SET self.detection_model_config TO detection_config
SET self.boundary_model_config TO boundary_config
SET self.detection_model_config["word2id"] TO self.word_dict.word2id
SET self.detection_model_config["pos2id"] TO self.pos_dict.word2id
SET self.detection_model_config["char2id"] TO self.char_dict.word2id
SET self.boundary_model_config["word2id"] TO self.word_dict.word2id
SET self.boundary_model_config["pos2id"] TO self.pos_dict.word2id
SET self.boundary_model_config["char2id"] TO self.char_dict.word2id
SET self.detection_model TO DetectionModel(**self.detection_model_config)
SET self.boundary_model TO BoundaryModel(**self.boundary_model_config)
self.detection_model.to(device=self.device)
self.boundary_model.to(device=self.device)
SET self.joint_loss_fn TO BagLossWithMarginLayer()
```

```
SET self.detection_loss_fn TO BagLossLayer()
SET self.boundary_loss_fn TO MaxMarginLossLayer(self.max_seq_len,
self.max_margin).to(device=self.device)
SET self.detection_optimizer TO
torch.optim.Adadelta(self.detection_model.parameters(),
                                    lr=self.detection_learning_rate)
SET self.boundary_optimizer TO
torch.optim.Adadelta(self.boundary_model.parameters(),
lr=self.boundary_learning_rate)
SET self.result_logger TO Seq2NuggetLogger()
SET self.match_eval TO ExactMatchEvaluator()
SET self.det_eval TO DetectionEvaluator()
SET self.current_epoch TO -1
SET self.detection_lr_decay_before TO 0
SET self.boundary_lr_decay_before TO 0
IF self.is_save_config:
   self.save_config(train_config, detection_config, boundary_config)
```

Then, as the next component, implementation on the head detection loss function can be showcased.

Algorithm 4.6: Capturing head detection loss function

```
DEFINE FUNCTION __init__(self, NIL_index=0):
   super(BagLossWithMarginLayer, self).__init__()
   SET self.NIL_index TO NIL_index
   SET self.epsilon TO 1e-8
   SET self.alpha TO 1
DEFINE FUNCTION forward(self, y_pred, targets, packages, margin_loss,
seq_mask, weight=None):
   SET probs TO torch.nn.functional.softmax(y_pred, dim=2)
   SET B, T, C TO probs.shape
```

```
   SET golden_probs TO torch.gather(probs, dim=2,
index=targets.unsqueeze(dim=2))
   SET probs_in_package TO golden_probs.expand(B, T, T).transpose(1, 2)
   SET probs_in_package TO probs_in_package * packages
   SET max_probs_in_package, _ TO torch.max(probs_in_package, dim=2)
   SET golden_probs TO golden_probs.squeeze(dim=2)
   SET golden_weight TO golden_probs / (max_probs_in_package)
   SET golden_weight TO golden_weight.view(-1)
   SET golden_weight TO golden_weight.detach()
   SET y_pred TO y_pred.view(-1, C)
   SET targets TO targets.view(-1)
   SET seq_mask TO seq_mask.view(-1)
   SET margin_loss TO margin_loss.view(-1)
   IF weight is None:
      SET weight TO torch.ones(C,
dtype=torch.float).to(device=y_pred.device)
   SET nil_label TO torch.tensor([self.NIL_index] * (B * T), dtype=torch.long,
device=y_pred.device)
   SET golden_loss TO torch.nn.functional.cross_entropy(y_pred, targets,
weight=weight, reduction='none')
   SET nil_loss TO torch.nn.functional.cross_entropy(y_pred, nil_label,
weight=weight, reduction='none')
   SET loss TO golden_weight ** self.alpha * (golden_loss + margin_loss) + (1 -
golden_weight ** self.alpha) * nil_loss
   SET loss TO torch.dot(loss, seq_mask) / (torch.sum(seq_mask) +
self.epsilon)
   RETURN loss
```

According to the methodological description, this novel concept of "*boundary bubbles*" is mainly relevant for NE boundary detection region classification. Capturing entity mention nuggets and capturing head detection loss function are really important

considering the next major task. A novel stack LSTM neural architecture has been used as the primary setup for determining the NE boundary detection. A dedicated novel algorithm has been introduced to cater the requirements of deriving the NE boundary detection regional classification. Let's focus on the next important coding implementation in which is really useful when deriving the NE boundary regional detection.

4.6 Implementation on NE Boundary Detection Regional Classification

For accomplishing the NE region classification task, two stack-LSTM layers have been adapted in the methodological section. The respective coding implementation can be displayed as follows.

Algorithm 4.7: NE boundary region classification

```
super(BoundaryModel, self).__init__()
   FOR key IN kwargs:
      SET self.__dict__[key] TO kwargs[key]
   SET self.word_embedding TO
EmbeddingLayer(dim=self.word_embedding_dim,
trainable=self.embedding_trainable)
   self.word_embedding.load_from_pretrain(self.pretrain_file, self.word2id)
   SET self.char_encoder TO
CharLSTMEncoder(char_embedding_dim=self.char_embedding_dim,
                        word_encoding_dim=self.word_encoding_dim,
                        num_vocab=len(self.char2id))
   SET self.pos_embedding TO
EmbeddingLayer(dim=self.pos_embedding_dim, trainable=True)
   self.pos_embedding.initialize_with_random(len(self.pos2id))
   SET self.rnn_layer TO LSTMLayer(D_in=self.word_embedding_dim +
self.pos_embedding_dim + self.word_encoding_dim,
                    D_out=self.hidden_dim,
                    n_layers=1,
                    dropout=0,
```

```
                    bidirectional=True)
  SET self.dropout_layer TO torch.nn.Dropout(p=self.dropout_rate)
  SET self.conv_layer_left TO TimeConvLayer(self.hidden_dim,
self.hidden_dim, 2)
  SET self.conv_layer_right TO TimeConvLayer(self.hidden_dim,
self.hidden_dim, 2)
  SET self.nugget_bilinear_layer_left TO
BilinearAttention2DLayer(self.hidden_dim, self.hidden_dim)
  SET self.nugget_bilinear_layer_right TO
BilinearAttention2DLayer(self.hidden_dim, self.hidden_dim)
  SET self.nugget_linear_layer_left TO torch.nn.Linear(self.hidden_dim, 1)
  SET self.nugget_linear_layer_right TO torch.nn.Linear(self.hidden_dim, 1)
  SET left_boundary_mask TO torch.zeros(self.max_seq_len, self.max_seq_len)
  SET right_boundary_mask TO torch.zeros(self.max_seq_len,
self.max_seq_len)
  FOR i IN xrange(self.max_seq_len):
    SET left_boundary_mask[i, max(i - self.win_size, 0):i + 1] TO 1
    SET right_boundary_mask[i, i:min(i + self.win_size + 1, self.max_seq_len)]
TO 1
  SET left_boundary_mask TO (1 - left_boundary_mask) * -9999999
  SET right_boundary_mask TO (1 - right_boundary_mask) * -9999999
  SET Parameter TO torch.nn.Parameter
  SET left_boundary_mask TO Parameter(left_boundary_mask,
requires_grad=False)
  SET right_boundary_mask TO Parameter(right_boundary_mask,
requires_grad=False)
  self.register_parameter("left_boundary_mask", left_boundary_mask)
  self.register_parameter("right_boundary_mask", right_boundary_mask)
```

According to the development settings, several hyper-parameters can be set off. Since this model can be described as a Stack-LSTM driven model, the inherited parameter list as well as some newly introduced parameter settings would be constructed the

overall cumulative hyper-parameter settings. A dedicated special RNN layer has been enriched to compute the embedding purposes. Along with that, several other layers have been defined to optimize the full LSTM architecture. All these hyper-parameter settings and tune-ups will be critically evaluated under the experimental results section and evaluation section. The implementation phase would be concluded with the final component, NE linking.

4.7 Implementation on NE Linking

NE linking is a novel research avenue under the named entity disambiguation (NED). Further, it can be identified as a vital task in information retrieval from knowledge bases. According to the methodological description, there are three varieties of NE linking. In this approach, only the direct mapping variety has been chosen for accomplishing NE linking task. The respective pseudocode implementation for NE linking task can be exhibited as follows.

Algorithm 4.8: NE linking

```
SET mapper TO ontoLoader()
SET entities TO []
  FOR entity IN res['entities']:
    FOR canonical_name IN mapper.load_kb(model_path).entities:
      IF canonical_name.canonical_name.lstrip('#') EQUALS entity['text']:
        entities.append(entity['text'])
        entities.append(" ")
```

According to the methodological description above, a novelty will not be achieved through NE linking process hence only a performance boosting-up will be expected as a by-product of the overall process improvement compared to the existing baselines. The respective experimental results will be exhibited under the next following sections.

The next chapter elaborates the overall testing procedure of the entire system. The respective experimental settings and the experimental results will be discussed on the following section.

Chapter 5
Experiments

This section describes the overall testing procedure of the entire solution of this research. Various requirements, decisions, countermeasures, factors, and conditions have been taken into consideration when accomplishing the overall testing procedure. For a better representation, this section can be divided into three main parts such as experimental settings, experimental baselines, and experimental results. The experimental settings can be demonstrated as follows.

5.1 Experimental Settings

Experiments have been determined based on the existing NE boundary detection baselines. For accomplishing this purpose, several existing baselines have been used as fundamentals and those baselines could be mentioned as conventional CRF and MLP models, region-based computational models and hypergraph based computational models.

5.1.1 Dataset

A dedicated Sinhala social media content enriched corpus weight of 99,000 has been used as the training corpus. Regarding the IOBE pattern for NE boundary regional tagging which has already been mentioned under the methodological approach, a dedicated tagged Sinhala NE corpus has been used to train the model on identifying the NE boundary aspects. This set has been manually tagged by a set of expert human annotators[3]. This specific corpus has been enriched with 120,000 corpora as per the training purposes, another different 100,000 as per the validation purposes and another different set of 80,000 as per the testing purposes. These different datasets, which have been consisted with Sinhala corpora and famous other datasets such as ACE2005, GENIA and KBP2017 [19], have been served for testing purpose, as per those would be required for layering the foundation for supervised deep neural modelling.

[3] These human annotators are full time research assistants working on specific linguistic projects at the University of Moratuwa, Sri Lanka.

5.1.2 Tools and Libraries

Several Python related computational linguistic modules has been used to set off the platform. Those libraries have already been mentioned under the implementation section. Apart from those libraries, another some of required toolkits, word & character embedding toolkits, and several other optimization functions have been used for accomplishing the testing purposes.

5.1.3 Hardware Requirements

A dedicated high-performance graphics processing unit (GPU) machine has been acquired to set up the whole implementation of the project. The same machine has been used for both comprehensive training and testing procedures. The GPU machine has been consisted with an Intel Core i7 10^{th} generation central processing unit (CPU), 16GB of random-access memory (RAM) and a 2GB of the GPU as per the overall specification. For acquiring the optimal usage of CPU + GPU and parallel execution capabilities under the heavy training procedure, the CUDA drivers has been installed. This installation would fasten the whole training procedure by tuning up the overall execution performance using the TensorFlow framework. Several hyper-parameters have been tuned up for obtaining the competitive edge with respect to several development phases under the developed novel algorithms which have been critically analyzed and presented under the methodology chapter[4]. Those will be further discussed under next sub section on experimental results.

As it has already been noted, overall experiments have been performed under the existing NE boundary detection baselines such as conventional CRF and MLP computational models, region-based computational models and hypergraph based computational models. Those baselines would be described in detail before moving ahead with the experimental results of the overall approach.

[4] The source code and the relevant datasets of this overall project have been kept as private due to certain restrictions. The access privileges for the implementation can be granted as per a request could be made through toprasanyapa@gmail.com.

5.2 Experimental Baselines

As it has already been mentioned, three benchmarks have been defined, based on the existing NE boundary detection baselines, to evaluate the performance on the *boundary bubbles*. Those evaluation comparisons would be determined conspiring the following benchmarks as follows.

5.2.1 Conventional CRF and MLP Computational Models

Considering the overall decoders which is being prominent under the latest computational linguistics approaches, LSTM-CRF would take a special position under the classical benchmarks under the NE detection and NE boundary detection. However, these related baselines would not consider the NE boundary aspect into consideration where the so-called outer NE mentions would only be considered for deriving the specific training processes. Another variant under the conventional CRF computational models would be the Multi-CRF. Multi-CRF could be mentioned as something exactly similar to the LSTM-CRF model. The specialty of the Multi-CRF would be the ability to distinguish the multi-NE types under the NE mention nature. The independent MLP models are also well-known for performing the NE type detection considering both inner and outer NE mentions. Considering all these models, the main baselines which can be useful for underlying evaluation purposes would be LSTM-CRF on top of MLP models.

5.2.2 Region-Based Computational Models

Several set of computational models can be used as the region-based computational models. Among those models, cascaded oriented CRF, transition computational models and encoding based FOFE model would be named as the most potential benchmarks for region-based computational models. Cascaded oriented CRF models would be dedicated for determining overall NE mentions at various aspects and contexts. Transition computational models would be useful for evaluating NE mentions through a series of actions. Encoding based FOFE model would be unique for classifying all the NE mention nuggets considering the sequences of given sentences.

5.2.3 Hypergraph-Based Computational Models

Two main hypergraph-based computational models such as LSTM-Hypergraph (LH) based model and CRF-Hypergraph (CH) based model would be adapted as per the evaluation purposes. LH model would be useful for learning contextual features from LSTM models by transforming those to the respective hypergraph models. CH would be dedicated for deriving the ambiguity removal under transformational models. These two are dedicated for accomplishing such state-of-art objectives.

Apart from that, the respective evaluation has been driven considering the benchmark models that is available in the market. Models have been evaluated considering the micro level evaluation criteria such as Precision, Recall and the F1 measure.

5.3 Experimental Results

Experimental results have been generated considering the experimental baselines which have been mentioned above. Results have been generated for the corpus which has been mentioned under the experimental settings. Several testing tasks have been followed considering testing on Sinhala named entity detection, testing on word embedding through deep transfer learning, testing on NE head word detection, testing on Sinhala NE boundary detection, and testing on NE linking. Those sub tasks under the overall testing process can be demonstrated as follows.

5.3.1 Testing on Sinhala NE Identification (Algorithm 4.1)

This research has been originated with the focus on the niche sector of Sinhala named entity detection. The following measurements have been set to evaluate the model.

- *True Positives (TP):* identified named entities which have been extracted using detection model by applying the ground truth of the social media content.
- *False Positives (FP):* identified named entities which have been extracted by detection model but failing to apply the ground truth of the social media content.
- *False Negatives (FN):* identified named entities which have been annotated according to the social media related ground truth which have been failed to extract by the defined detection model.

The respective average experimental results for Sinhala named entity detection module (Algorithm 4.1) along with the required hyperparameters can be showcased as follows.

Table 5.1 : RESULTS ON SINHALA NE DETECTION

Hyper-Parameter Matrices	**Training Results**	**Testing Results**
Max_len	50 - 1000	
Precision (%)	60.14	55.01
Recall (%)	60.14	55.02
F1-Score (%)	60.11	55.03
Mean Absolute Error (%)	42.02	46.27
Precision Difference (%)	5.14	
Elapsed Time (Sec)	128.37	

The above-mentioned hyper-parameter settings have been optimized for the initial set of 20 corpus tuples for 10 epochs each. The respective human value for indicating whether the Sinhala offensive content has been included inside the corpus has been flagged using a Boolean value. The respective evaluation metric criteria can be demonstrated as follows. The optimized hyper-parametric based experiments have been conducted based on the precision, recall, F1 score, mean absolute error, precision difference and elapsed time.

Table 5.2 : HYPER-PARAMATER BASED RESULTS ON SINHALA NE DETECTION

Set	Human Value	max_words=50 LSTM layers=5 delta=0.5 drop_out=0.1 batch_size=10
1	Y	60.12 : 58.12 : 59.15 : 0.24 : 61.75 : 59.16
2	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13

3	N	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
4	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
5	Y	61.75 : 57.32 : 58.15 : 0.25 : 62.43 : 59.28
6	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
7	N	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
8	N	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
9	N	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
10	N	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
11	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
12	Y	61.75 : 57.32 : 58.15 : 0.25 : 62.43 : 59.28
13	N	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
14	Y	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
15	N	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
16	N	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
17	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
18	Y	61.48 : 59.49 : 60.76 : 0.26 : 60.68 : 59.49
19	Y	62.12 : 59.12 : 61.15 : 0.27 : 62.49 : 60.13
20	Y	61.75 57.32 : 58.15 : 0.25 : 62.43 : 59.28

5.3.2 Testing on Word Embedding (Algorithm 4.3)

Even though a dedicated word embedding mechanism could be followed for catering the Sinhala word embedding process, a transfer learning approach has been adapted for obtaining some promising results. This can be introduced as a famous approach in most of the latest deep neural based models. The respective experimental results can be exhibited as follows.

Table 5.3 : AVERAGE RESULTS ON WORD EMBEDDINGS

Criteria	Precision (%)	Recall (%)	F1 (%)
Dev Detect	81.10	76.13	78.19
Test Detect	79.15	75.14	
Dev All	0.37	0.46	
Test All	0.34	0.33	
Dev Gap	79.26	76.02	75.13
Test Gap	79.15	78.26	78.12

5.3.3 Testing on NE Mention Detection of Head Words (Algorithm 4.4)

Named entity head word identification has been described as one of the major scientific novelties of this overall approach under the comprehensive methodology. Head word detection results also have been obtained considering the same matrices as follows.

Table 5.4 : AVERAGE RESULTS ON NE HEAD WORD DETECTION

Criteria	Precision (%)	Recall (%)	F1 (%)
Dev Detect	85.98	78.53	82.09
Test Detect	78.79	81.75	
Dev All	0.22	0.24	
Test All	0.24	0.24	
Dev Gap	85.71	78.32	81.85
Test Gap	84.68	78.55	81.51

There are few things to be considered under the NE head word detection phase. Two parameters could be determined as the key factors which determine the overall procedure of head word detection such as NE mention nuggets identifier and head detection loss value. Those parameters have fully contributed for performing the

overall NE head word detection phase whereas those two should be carefully monitored under the testing phase as well. Specially, considering the independent and dependent variable of those algorithms should be carefully monitored to find out the optimal values in terms of generating the maximum testing outcome.

The parametric values for mention nuggets identifier algorithm should be considered. There are few parameters and those can be classified as independent and dependent variables considering the nature of a hypothesis as follows.

Table 5.5 : PARAMETERS ON MENTION NUGGETS IDENTIFIER

Independent Variables	x_i	c_i	v_i
Dependent Variables	ω	$\emptyset$	

Considering different values for the independent variables, the behavior of the dependent variables has been analyzed. The respective behavior can be showcased as follows. The x_i value range has been taken from 0 to 1 whereas 0.5 constant has been set for both c_i and v_i.

Table 5.6 : VALUE ADJUSTMENTS ON MENTION NUGGETS IDENTIFIER

$\omega = 0.5$	$x_{i=0.2}$	$c_{i=0.5}$	$v_{i=0.5}$
$\emptyset = 0.2$	0.02	0.012	0.012
$\omega = 0.5$	$x_{i=0.4}$	$c_{i=0.5}$	$v_{i=0.5}$
$\emptyset = 0.4$	0.013	0.01284	0.012
$\boldsymbol{\omega = 0.5}$	$\boldsymbol{x_{i=0.6}}$	$\boldsymbol{c_{i=0.5}}$	$\boldsymbol{v_{i=0.5}}$
$\boldsymbol{\emptyset = 0.6}$	***0.013***	***0.012***	***0.013***
$\omega = 0.5$	$x_{i=0.8}$	$c_{i=0.5}$	$v_{i=0.5}$
$\emptyset = 0.8$	0.012	0.013	0.012

$\omega = 0.5$	$x_{i=1.0}$	$c_{i=0.5}$	$v_{i=0.5}$
$\emptyset = 1.0$	0.012	0.011	0.012

According to the outcome, it can be concluded as when the x_i gets 0.6, the output has been maximized. This can be used as the most optimal hyperparameter adjustment for the next evaluations.

5.3.4 Testing on Named Entity Boundary Detection (Algorithm 4.7)

Next core scientific novelty of this whole approach is to determine the named entity boundary detection. This is the component which drives the entire solution. The comprehensive average experimental results can be showcased as follows. For the demonstration purposes the most prominent five NE types would be listed out of 20 NE types in total.

Table 5.7 : AVERAGE RESULTS ON NE BOUNDARY DETECTION

NE Type	Precision	Recall	F1	Support
ORG	0.59	0.53	0.56	1766
LOC	0.82	0.82	0.82	2111
PER	0.75	0.70	0.72	698
PRODUCT	0.61	0.65	0.63	537
FACILITY	0.25	0.02	0.03	133
micro avg	0.60	0.55	0.57	9464
macro avg	0.56	0.55	0.55	9464

5.3.5 Testing on NE Linking (Algorithm 4.8)

The final phase of the overall testing procedure would be the testing on NE linking. For the respective linking purposes, several topics have been used under the overall

Sinhala related social media knowledge base. The respective set of topics and the relevant experimental results can be showcased as follows.

Table 5.8 : RESULTS ON NE LINKING

Topic Set	Precision (%)	Recall (%)	F1 Score (%)
Set 1	68.25	65.25	67.21
Set 2	69.16	62.44	65.23
Set 3	71.28	68.25	70.02
Set 4	70.22	68.16	69.44
Set 5	76.14	75.04	76.03
Set 6	69.26	67.16	68.25
Set 7	74.22	73.25	74.35

The overall average results which are related to the NE linking component can be demonstrated as follows.

Table 5.9 : AVERAGE RESULTS ON NE LINKING

Precision (%)	Recall (%)	F1 Score (%)
73.57	68.43	72.08

The next chapter critically evaluates the above obtained experimental results. The evaluation will be conducted considering the main NE boundary detection categories as per conventional CRF and MLP computational models, region-based computational models and hypergraph based computational models. Overall evaluation has been conducted considering the most prominent deep neural based models which have been introduced recently. Evaluation criteria consists with major aspects of the Precision, Recall, F1, performance analysis, and inference time analysis.

Chapter 6
Evaluation and Discussion

This section describes the overall evaluation which has been conducted for the derived experimental results in the previous chapter. The evaluation will be conducted considering the main NE boundary detection categories like conventional CRF and MLP computational models, region-based computational models and hypergraph based computational models. Afterwards the evaluation criteria would be critically analyzed and discussed to demonstrate the outcome of the entire research.

Again, the overall evaluation on performance can be discussed considering the subcomponents which have already been considered under the experimental results as follows.

- Evaluation on Sinhala NE identification and social media analysis
- Evaluation on word embedding representation
- Evaluation on NE mention head word detection
- Evaluation on NE boundary detection
- Evaluation on NE linking

Considering the evaluation matrices, the usual precision, recall, F1 have been employed to demonstrate the performance. For performance evaluation comparisons, the prominent benchmark models have been used.

6.1 Evaluation on Sinhala NE identification and social media analysis

As it has already been mentioned under the experimental results section, several annotators have been employed for constructing the main corpora such as classifying social media statements and grouping the identified named entities of those statements into their respective mention categories. Since the manual involvement has been used, the interrater reliability has to be evaluated before using the corpora for the computational purposes within the system. Hence, the Fleiss' Kappa values have to be measured to evaluate the reliability of the corpus which has been used in the entire system. Fleiss' Kappa has been considered as an extension of the famous Cohen's

Kappa which is applicable for the scenarios where there are three or more annotators have been employed [50].

The extracted named entities, considering 120,000 as per training corpus, 100,000 as per validation corpus and 80,000 as per testing purposes, have been shared among three main annotators for accomplishing the entity annotation process. Once the annotation process has been accomplished, the inter annotator agreement has been processed to evaluate the relevant Kappa statistics. The individual Fleiss' Kappa values have been generated considering the main predefined NE categories such as *PER, LOC, ORG, DES,* and *PRO*. These respective individual Kappa statistics can be evaluated and showcased under the *Kappa Values for Individual NE Categories* table as follows.

Table 6.1 : KAPPA VALUES FOR INDIVIDUAL NE CATEGORIES

Kappa Values for Individual NE Categories					
NE Class	**Conditional Probability**	**Kappa**	**Standard Error**	**Z Value**	**P Value**
PER	0.84	0.76	0.12	5.12	0.04
LOC	0.76	0.64	0.12	4.87	0.04
ORG	0.67	0.62	0.12	4.28	0.03
DES	0.48	0.57	0.12	3.72	0.02
PRO	0.43	0.52	0.12	3.24	0.02

Once the Kappa values have been obtained for the respective individual NE categories, then the overall Kappa statistics can be obtained and demonstrated as follows.

Table 6.2 : OVERALL KAPPA VALUES FOR NE CATEGORIES

Overall Kappa					
NE Class	**Conditional Probability**	**Kappa**	**Standard Error**	**Z Value**	**P Value**
Overall	0.78	0.72	0.11	4.86	0.02

According to the values of table 6.2, the overall Kappa value for NE identification and categorization can be recognized as 0.72. Considering the overall value and according to the classification of the Fleiss' Kappa, it can be concluded that the generated Kappa value has represented a good strength in the inter annotator agreements [51].

In terms of evaluating the model on NE type classification, set of main parameters have been defined and discussed in the previous chapter. The obtained results have been critically evaluated with the available benchmarks as per the table 6.3. The newly introduced unique model has been demonstrated as *BB2022 (boundary bubbles)* [52]. The baselines which have been demonstrated such as SH2016, ARN2019 and KBP2018 have been chosen considering several baselines which have been described under the constructive literature review [27], [28] and the respective results have been obtained considering the execution of each model with respect to the earlier mentioned corpus.

Table 6.3 : EVALUATION ON NE TYPE CLASSIFICATION

Model	**Precision (%)**	**Recall (%)**	**F1 (%)**
SH2016[5]	75.9	70.1	72.8
KBP2018[5]	72.6	73.0	72.8
ARN2019[5]	75.2	72.5	73.9
BB2022	***76.2***	***73.6***	***74.9***

As per the exhibited comparison outcomes, it can be concluded that the proposed novel model performs better than the existing NE type detection benchmarks in a competitive edge. Next, the evaluation process on word embedding through deep transfer learning has to be conducted as follows.

[5] Existing benchmarks which have been implemented and available for hate speech identification purposes.

6.2 Evaluation on Word Embedding

In terms of achieving the competitive advantage over the most prominent benchmarks, deep transfer learning mechanism has been used. The evaluation process has been conducted considering a correlational graphical process considering the performance outcome with respect to the manual annotations which has been conducted for determining character level embedding. Several word embedding transfer models such as Word2Vec, BERT and ELMo have been used for the comparison with respect to the GloVe approach. These have performed in different ways under different levels of contexts and the prevailing context is a specialized domain which is unique from any other context. So, the overall evaluation on the transfer learning should be conducted in a more sophisticated manner.

The following figure 6.1 reveals the outcome of the evaluation on word embedding through deep transfer learning over the existing baselines.

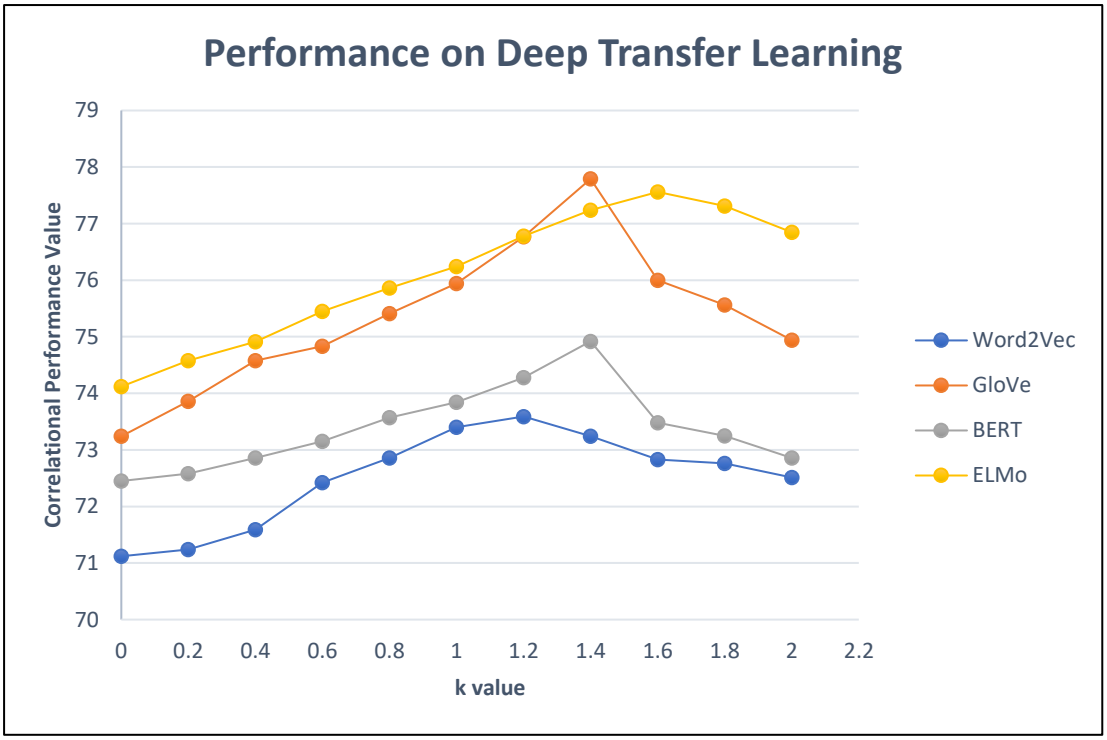

Figure 6.1. Performance comparison on deep transfer learning

As per the comparison, some of the key points can be derived as follows.

- Until the *k* value gets 1.4, there is a rising behavior of the performance in all the models except ELMo where it reaches to its maximum when *k* becomes 1.6.

- Considering Word2Vec, GloVe and BERT models, the maximum outcome has been demonstrated by GloVe. So, considering all aspects, the proposed approach has shown the best outcomes out of the given benchmarks in the market.

Later, with the introduction of cross-lingual word embeddings, some state-of-the-art word embedding models have been invented. One of such prominent mechanisms would be the usage of XLM-R for obtaining word embeddings considering major text classification tasks [53]. Unlike most of the other word embedding approaches, XLM-R supports more than 100 languages including Sinhala [54]. Hence, another evaluation approach has been applied to compare the proposed model and the XLM-R model where the results can be demonstrated as follows.

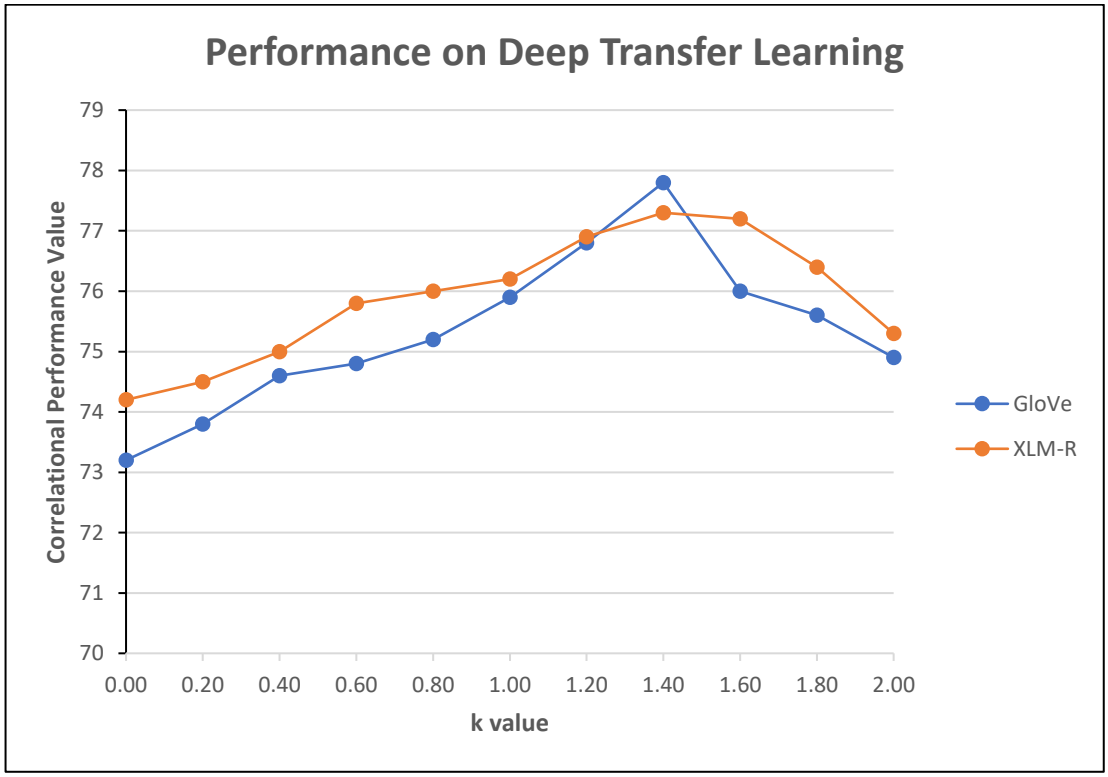

Figure 6.2. GloVe and XLM-R performance comparison on transfer learning

As per the comparison between the GloVe and XLM-R models, it can be concluded that the maximum outcome has been showcased by GloVe even though XLM-R has performed well in most of the sections in the graph distribution.

Next evaluation phase on NE mention head word detection can be exhibited as follows.

6.3 Evaluation on NE Mention Head Word Detection

NE head word identification has already been described as the major scientific novelty of this overall approach under the comprehensive methodology. The experimental results have demonstrated some promising results in the previous chapter. Those outcomes would be compared with the existing benchmarks. As per the benchmarks, models which have been invented by Sohrab et.al. [19], Ju et.al. [22] and ARN [23] have been chosen and the respective conducted evaluations can be showcased as follows. The newly introduced unique model has been indicated as BB(2022) (*BB means boundary bubbles*).

Table 6.4 : EVALUATION ON NE HEAD WORD DETECTION

Model	**Set**	**NE Head Word Detection**		
		Precision (%)	**Recall (%)**	**F1 (%)**
Sohrab et al. (2018)[6]	1	71.68	71.25	71.28
	2	71.59	70.48	71.24
	3	71.43	71.39	71.31
	4	72.69	71.59	71.84
	5	71.89	71.52	71.64
	6	72.98	71.42	71.55
Ju et al. (2019)[6]	1	69.11	68.87	68.91
	2	68.47	68.24	68.12
	3	68.35	68.29	68.23
	4	68.48	68.22	68.31
	5	68.94	68.47	68.51
	6	69.24	68.86	68.23
	1	78.13	76.21	78.06

[6] These benchmark results have been obtained from the respective publications.

ARN (2019)[6]	2	78.21	76.47	78.17
	3	78.65	76.34	78.35
	4	77.98	75.27	77.76
	5	77.83	76.39	77.76
	6	78.11	77.53	78.10
BB (2022)	1	78.95	77.35	78.54
	2	78.24	76.97	78.13
	3	78.88	77.31	78.24
	4	78.65	78.03	78.59
	5	79.16	77.68	78.88
	6	79.14	77.21	78.92

As per the critically evaluation, Sohrab model has been invented around 2018 and Ju et.al. model has been introduced around mid-2019. Considering these two models, Sohrab model perform well in terms of all three performance evaluation criteria. When it comes for ARN, which has been introduced around late 2019, has been performed way better than the above mentioned both models. So, until mid-2020, ARN could be introduced as the pioneer model for NE head word detection. Then, it can be compared the proposed model with the ARN, and the evaluation results have showcased that the novel model performs slightly better than the ARN in all three performance evaluation criteria. So, as per the discussion, it can be concluded that, *BB(2022)* performs well than the existing cutting-edge baselines in the market. Further as per a summary view, the average comparison can be showcased as follows.

Table 6.5 : SUMMARY EVALUATION ON NE HEAD WORD DETECTION

NE Head Word Detection			
Model	**Precision (%)**	**Recall (%)**	**F1 (%)**
Sohrab et al. (2018)	72.76	71.43	71.28
Ju et al. (2019)	68.49	68.34	68.19
ARN (2019)	78.24	76.57	77.39
BB (2022)	***78.82***	***77.16***	***78.53***

Next important aspect is to consider the evaluation on NE boundary detection.

6.4 Evaluation on Named Entity Boundary Detection

This can be introduced as the next major scientific novelty of this entire research attempt. As per the methodology and implementation, named entity boundary detection also has been addressed by the same solutions which have been worked on NE head word detection. The evaluation can be exhibited considering those models such as Sohrab et.al. [19], Ju et.al. [22] and ARN [23]. The respective results can be demonstrated as follows.

Table 6.6 : EVALUATION ON NE BOUNDARY DETECTION

Model	**Boundary detection**		
	Precision (%)	**Recall (%)**	**F1 (%)**
Sohrab et al.(2018)	76.61	69.20	72.71
Ju et al. (2019)	79.90	67.08	71.36
ARN (2019)	81.34	68.20	72.83
BB (2022)	***82.18***	***71.34***	***76.54***

Considering the outcomes, the novel *BB(2022)* model has performed well in F_1 measure matrix compared to all of the available benchmarks. Considering the overall

evaluation results of the NE boundary detection, the evaluation outcome of the novel *BB(2022)* model has been based on Sinhala named entity related information whereas other benchmarks have been focused mainly on English, French and German related NE corpora. The whole purpose of this comparison is to show off that the novel specialized *BB(2022)* model which has been invented based on poor NLP resources, settings and fundamentals of the Sinhala language, has performed well ahead than the other models which have been introduced based on rich NLP resources, settings and fundamentals. As a summary it can be concluded that the *BB(2022)* novel model has performed well considering all the available top notch benchmarks for NE boundary detection.

The final evaluation phase is to derive evaluation for NE linking process.

6.5 Evaluation on NE Linking

NE linking can be introduced as a major task in information extraction from knowledge bases. This has been practiced in several latest research and certain models have been introduced based on those developments. Such several models which have been used for the evaluation purposes could be listed down as DBpedia Spotlight[7], SOTA[8], and VCG[9]. The respective benchmarks have been critically evaluated with the novel model, *BB(2022)*, then the respective evaluation results can be showcased as follows.

Table 6.7: EVALUATION ON NE LINKING

Model	**Precision (%)**	**Recall (%)**	**F1 (%)**
DBpedia Spotlight	52.64	52.48	52.53
SOTA	57.26	51.24	55.24
VCG	61.25	58.26	60.18
BB(2022)	***62.47***	***60.15***	***61.27***

[7] https://www.dbpedia-spotlight.org/api.
[8] Exploring Neural Entity Representations for Semantic Information, A Runge, E Hovy - arXiv preprint arXiv:2011.08951, 2020 - arxiv.org.
[9] https://paperswithcode.com/task/entity-linking

According to the comparisons, it can be seen that the *BB(2022)* model has performed well compared to all the other existing baselines. All the performance matrices have showcased that *BB(2022)* novel approach has outperformed better than the prominent benchmarks in the market. It is a must to highlight that the NE linking has been performed only considering the direct mapping approach.

That concludes the evaluation chapter. The next section concludes the overall study which has been executed throughout this research. The main points on identifying the research gap, methodological approach, experimental results, and evaluation process have been summarized.

Chapter 7
Conclusion

This section concludes the overall attempt of this research through indicating the research gap, methodological approach, overall implementation plan, the experimental results, and the conducted evaluation process of the experiments.

According to the above-mentioned objectives, the methodological plan has been conducted. According to objective 1, a constructive literature review for analyzing the already implemented NE type and NE boundary identification approaches and systems have been conducted. In accordance with the objective 2, identifying the core linguistic structures under Sinhala named entity identification has been obtained. The dataset annotation process has been carried out considering three social media platforms: Facebook, YouTube, and Twitter. Then, aligning with the proposed timeline, the deliverables under objective 3 and 4, have been delivered. The novelty parts have been implemented along with the 3rd objective, identifying the novelty in implementing NE boundary detection considering NE types of Sinhala related statements. Finally, as per the 4th objective, NE linking part has been implemented. Overall experimental results and the evaluation process have been conducted and accomplished under the 5th objective.

After the component implementation, system integration has been carried out, and the respective system testing has been executed. In terms of accomplishing system testing, confusion matrix calculation would be set off as the most important part. A dedicated different testing corpus has been used as the dataset, which is unique than the corpus which has been used for training purposes, for this whole purpose. Precision, recall, f-measure, and accuracy have been calculated as per the experimental deliverables. Additionally, performance and execution time have also been considered for accomplishing some non-functional requirements. These would ensure accomplishing the overall evaluation procedure of this proposed system.

For choosing the testing corpus, the dataset has been consisted with Sinhala related extracted statements, posts, and comments from social media platforms. Since the deep transfer learning has been adapted, not only the Sinhala corpus but also the typical

datasets such as ACE2005, GENIA and KBP2017 [19] have also been used for the overall experimental comparison evaluation purposes. Specially during the execution of overall system testing, state-of-the-art NE boundary detection architectures have been considered. According to [24] deep neural network-based mechanisms have outperformed all non-neural models (machine learning models, rule-based models etc.). Hence, under the evaluation related tasks, only deep neural model-based solutions have been adapted for the evaluation purposes. As the conclusion, it can be accepted that the proposed novel mechanism on identifying NE boundary detection has outperformed the most prominent existing baselines in the market.

Considering the main contribution to the field of computational linguistics, as it has already been explained earlier, there are no proper computational mechanism for measuring Sinhala related NE boundary detection under social media context. Even though there are plenty of solid approaches for detecting NE boundary detection specially for English domain, NE boundary detection for Sinhala is a rising area and has not been matured. Both NE type and boundary aspects are relevant and important for NLP related tasks such as information extraction, information retrieval, NE linking etc. Social media analysis for Sinhala has been experimented using machine learning approaches but applying a deep learning mechanism for detecting Sinhala NE boundary aspects can also be mentioned as a first-time appearance. NE linking has already been done, especially direct mapping, for various domains including English, German, Arabic etc. but for most of the Indo-Aryan languages, still this is a growing research area. Regardless of the context, the proposing NE linking mechanism could be mentioned as a novel approach where useful for NE linking with knowledge bases. So, those can be introduced as the novel contribution from this research to the field of computational linguistics considering low resourceful languages like Sinhala.

Considering all these aspects, few avenues can be introduced as the possible future enhancements under the whole process of detecting Sinhala related NE boundaries along with the considered NE type. It has been assumed as a particular head word would not be shared among more than one sentence nuggets under the section of dominant word detection considering MLP on top of CRF along with NE type

detection in the constructive methodological approach. Therefore, sharing a given respective head word among several multiple sentence nuggets has been considered as a possible future enhancement under the overall methodological approach. Experimenting on the applicability of sharing a particular head word among multiple different sentence nuggets would be a major scientific research advancement of this entire domain.

References

[1] Jing Li, Aixin Sun and Yukun Ma, Neural Named Entity Boundary Detection, IEEE Transactions on Knowledge and Data Engineering, 2015.

[2] Jing Li, Deheng Ye and Shuo Shang, Adversarial Transfer for Named Entity Boundary Detection with Pointer Networks, Twenty-Eighth International Joint Conference on Artificial Intelligence {IJCAI-19}, 2019.

[3] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou, A Neural Layered Model for Nested Named Entity Recognition, In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1446–1459, 2018.

[4] M. Nandathilaka, S. Ahangama and G. T. Weerasuriya, A Rule-based Lemmatizing Approach for Sinhala Language, 2018 3rd International Conference on Information Technology Research (ICITR), 2018, pp. 1-5, doi: 10.1109/ICITR.2018.8736134.

[5] K. T. P. M. Kariyawasam, S. Y. Senanayake and P. S. Haddela, A Rule Based Stemmer for Sinhala Language, 2019 14th Conference on Industrial and Information Systems (ICIIS), 2019, pp. 326-331, doi: 10.1109/ICIIS47346.2019.9063286.

[6] R. Jenarthanan, Y. Senarath and U. Thayasivam, ACTSEA: Annotated Corpus for Tamil & Sinhala Emotion Analysis, 2019 Moratuwa Engineering Research Conference (MERCon), 2019, pp. 49-53, doi: 10.1109/MERCon.2019.8818760.

[7] Y. Kodithuwakku and S. Hettiarachchi, AdaptText: A Novel Framework for Domain-Independent Automated Sinhala Text Classification, 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS), 2021, pp. 240-245, doi: 10.1109/ICIAfS52090.2021.9605861.

[8] K. Shanmugalingam and S. Sumathipala, Language identification at word level in Sinhala-English code-mixed social media text, 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE), 2019, pp. 113-118, doi: 10.23919/SCSE.2019.8842795.

[9] Abhishek Abhishek, Ashish Anand and Amit Awekar, Fine-Grained Entity Type Classification by Jointly Learning Representations and Label Embeddings, Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017.

[10] Dani Yogatama, Daniel Gillick, and Nevena Lazic, Embedding Methods for Fine Grained Entity Type Classification, In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 291–296, 2015.

[11] J. Li, A. Sun, J. Han and C. Li, A Survey on Deep Learning for Named Entity Recognition, in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 1, pp. 50-70, 1 Jan. 2022, doi: 10.1109/TKDE.2020.2981314.

[12] Ioannis Partalas, Cédric Lopez, Nadia Derbas and Ruslan Kalitvianski, Learning to Search for Recognizing Named Entities in Twitter, Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT), 2016.

[13] C. Yu, J. Wang, Y. Chen and M. Huang, Transfer Learning with Dynamic Adversarial Adaptation Network, 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 778-786, doi: 10.1109/ICDM.2019.00088.

[14] Jason P.C. Chiu and Eric Nichols, Named Entity Recognition with Bidirectional LSTM-CNNs, Transactions of the Association for Computational Linguistics, Volume 4, 2016.

[15] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, Natural language processing (almost) from scratch, The Journal of Machine Learning Research, 12:2493–2537, 2011.

[16] Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung and Guandong Xu, A Boundary-aware Neural Model for Nested Named Entity Recognition, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019.

[17] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami and Chris Dyer, Neural Architectures for Named Entity Recognition, Proceedings of NAACL-HCT 2016, pages 260-270, 2016.

[18] Miguel Ballesteros, Chris Dyer, and Noah A. Smith, Improved transition-based dependency parsing by modeling characters instead of words with LSTMs, In Proceedings of EMNLP, 2015.

[19] Mohammad Golam Sohrab and Makoto Miwa, Deep Exhaustive Model for Nested Named Entity Recognition, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 2843-2849, 2018.

[20] Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan, Recognizing Names in Biomedical Texts: A Machine Learning Approach, Bioinformatics, 20(7):1178–1190, 2004.

[21] Arzoo Katiyar and Claire Cardie, Nested Named Entity Recognition Revisited. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 861–871, New Orleans, Louisiana. ACL, 2018.

[22] Nguyen Truong Son and Nguyen Le Minh, Nested Named Entity Recognition Using Multilayer Recurrent Neural Networks, In Proceedings of PACLING 2017, pages 16–18, Sedona Hotel, Yangon, Myanmar, 2017.

[23] Hongyu Lin, Yaojie Lu, Xianpei Han and Le Sun, Sequence-to-Nuggets: Nested Entity Mention Detection via Anchor-Region Networks, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019.

[24] Wang, B., and Lu, W, Neural segmental hypergraphs for overlapping mention recognition, In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 204–214, 2018.

[25] Wang, B.; Lu, W.; Wang, Y.; and Jin, H, A neural transition-based model for nested mention recognition, In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1011–1017, 2018.

[26] Xu, M.; Jiang, H.; andWatcharawittayakul, S, A local detection approach for named entity recognition and mention detection, In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, volume 1, pages 1237–1247, 2017.

[27] Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang and Fei Huang, Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition, The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), 2020.

[28] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu and Jiwei Li, A Unified MRC Framework for Named Entity Recognition, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.

[29] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer, Zero-shot relation extraction via reading comprehension, Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pages 333–342, 2017.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, Proceedings of NAACL-HLT 2019, pages 4171–4186, 2018.

[31] Y. Chen et al, A Boundary Regression Model for Nested Named Entity Recognition, citated at Computation and Language (cs.CL); Artificial Intelligence (cs.AI) as arXiv:2011.14330 [cs.CL], 2020.

[32] J. Sivic and A. Zisserman, Video google: A text retrieval approach to object matching in videos. In null, page 1470. IEEE, 2003.

[33] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. arXiv preprint arXiv:1911.03977, 2019.

[34] H. M. S. T. Sandaruwan, S. A. S. Lorensuhewa and M. A. L. Kalyani, Sinhala Hate Speech Detection in Social Media using Text Mining and Machine learning, 2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2019, pp. 1-8, doi: 10.1109/ICTer48817.2019.9023655.

[35] Gitari, N. D. et al., A lexicon-based approach for hate speech detection, International Journal of Multimedia and Ubiquitous Engineering, 10(4), pp. 215–230. doi: 10.14257/ijmue.2015.10.4.21, 2015.

[36] Cambria, E. et al., SenticNet : A Publicly Available Semantic Resource for Opinion Mining, Artificial Intelligence, pp. 14–18. doi: 10.1038/leu.2012.122, 2010.

[37] Köffer, S. et al., Discussing the Value of Automatic HateSpeech Detection in Online Debates, Multikonferenz Wirtschaftsinformatik, (October), pp. 83–94. doi: 10.1111/j.1365-2923.2008.03277.x, 2018.

[38] Malmasi, S. and Zampieri, M., Detecting Hate Speech in Social Media, pp. 467–472. doi: 10.26615/978-954-452-049-6_062, 2017.

[39] Davidson, T. et al., Automated Hate Speech Detection and the Problem of Offensive Language, (Icwsm), pp. 512–515. doi:10.1561/1500000001, 2017.

[40] Alfina, I. et al., Hate speech detection in the Indonesian language: A dataset and preliminary study, 2017 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2017, 2018–October, pp. 233–237. doi: 10.1109/ICACSIS.2017.8355039, 2018.

[41] D. S. Dias, M. D. Welikala and N. G. J. Dias, Identifying Racist Social Media Comments in Sinhala Language Using Text Analytics Models with Machine Learning, 2018 18th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, 2018, pp. 1-6, doi: 10.1109/ICTER.2018.8615492.

[42] W. Warner and J. Hirschberg, Detecting hate speech on the world wide web, in Proceedings of the Second Workshop on Language in Social Media, LSM ’12, 2012, pp. 19-26.

[43] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad and Y. Chang, Abusive language detection in online user content, in Proceedings of the 25th International Conference on World Wide Web, WWW’16, 2016, pp. 145-153.

[44] B. Ross, M. Rist, G. Carbonell, B. Cabrera, N. Kirowsky and M. Wojazki, Measuring the reliability of hate speech annotations: The case of the European refugee crisis, in Proceedings of the Workshop on Natural Language Processing gor Computer Mediated Communication, 2017, pp. 6-9.

[45] N. Djuric, J. Zhou, R. Morris, M. Grbovic, V. Radosavljevic and N.Bhamidipati, Hate speech detection with comment embeddings, in Proceedings of the 24th International Conference on World Wide Web, 2015, pp. 29-30.

[46] Z. Waseem and D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in Proceedings of NAACL-HLT, 2016, pp. 88-93.

[47] P. Badjatiya, S. Gupta, M. Gupta and V. Varma, Deep learning for hate speech detection in tweets, in Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 759-760.

[48] F. Vigna, A. Cimino, F. DellOrletta, M. Petrocchi and M. Tesconi, Hate me, hate me not: Hate speech detection on facebook, in The First Italian Conference on Cybersecurity, 2017, pp. 86 – 95.

[49] S. Sulpizio et al., Are you really cursing? Neural processing of taboo words in native and foreign language, Brain and Language, Volume 194, pp. 84-92, 2019.

[50] Falotico, R., Quatto, P. Fleiss' kappa statistic without paradoxes, Qual Quant 49, 2015, pp. 463–470.

[51] Gwet, Kilem L, Large-Sample Variance of Fleiss Generalized Kappa, Educational and Psychological Measurement 81, 2021, pp. 781-790.

[52] Y. H. P. P. Priyadarshana, L. Ranathunga, C. R. J. Amalraj and I. Perera, HelaNER: A Novel Approach for Nested Named Entity Boundary Detection, IEEE EUROCON 2021 - 19th International Conference on Smart Technologies, 2021, pp. 119-124.

[53] A. Alcoforado et al., ZeroBERTo: Leveraging Zero-Shot Text Classification by Topic Modeling, 15th International Conference on Computational Processing of Portuguese, 2017.

[54] Efsun Sarioglu Kayi, Linyong Nan, Bohan Qu, Mona Diab, and Kathleen McKeown, Detecting Urgency Status of Crisis Tweets: A Transfer Learning Approach for Low Resource Languages, In Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 4693–4703.