Bootstrapping Sinhala Named Entities for NLP Applications

K.L. Jayasinghe

Bootstrapping Sinhala Named Entities for NLP Applications

K.L. Jayasinghe

Index : 14000512

Supervisor : Mr. W.V. Welgama

Submitted in partial fulfillment of the requirements of the
B.Sc. (Hons) in Computer Science Final Year Project (SCS 4124)

December 2018

Declaration

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

Candidate Name: Kasun Lakmal Jayasinghe

.....
Signature of Candidate
Date:

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

Principle Supervisor's Name: Mr. W.V. Welgama

.....
Signature of Supervisor
Date:

Co-Supervisor's Name: Dr. R. Weerasinghe

.....
Signature of Co-Supervisor
Date:

Abstract

Popular languages have lots of data pools to use in linguistic data applications. But languages like Sinhala have lack of data. Because of that researchers conducted studies to increase labeled data as part of speech words, Named entities and other semantic categories. Most of their studies are based on supervised learning or statistical methods which require big effort to label the train data. The proposed solution tries to design a method that requires less effort and increase the labeled Sinhala named entity data in average accuracy. It is a semi-supervised bootstrapping method which uses an iterative seeding mechanism to extract named entities in person and location categories.

The complete process conducted in two main phases. First one was the bootstrapping process and outputs of the process used to train the supervised learning process which is the second phase. So evaluation was also conducted in two phases. The first intermediate bootstrapping result shows 91% accuracy and the second phase result is also shown the intended accuracy level.

Keywords - Sinhala Named-entity recognition, Bootstrapping, Semi-supervised learning

Preface

A novel mechanism for increase Sinhala named entity data without putting more effort is introduced in this dissertation. The concept of bootstrapping that uses iterative seeding mechanism is taken from previous studies. But internal design and implementation was solely my own work and has not been proposed in any other study related to Bootstrapping. The evaluation model introduced in this dissertation is also a novel evaluation model and has not been proposed in any other work in the domain of Bootstrapping.

Acknowledgements

Special thanks go to my supervisor, Mr W.V Welgama and co-supervisor Dr Ruvan Weerasinghe, senior lecturers of University of Colombo School of Computing for guiding me on the topic, for offering invaluable advice and for the support given to me throughout the research.

I would like to thank Dr D.D. Karunaratne, senior lecturer of University of Colombo School of Computing and Dr H. A.Caldera, senior lecturer of University of Colombo School of Computing for providing feedback on the research proposal and interim report to improve the study.

My sincere thanks go to Dr H.E.M.H.B.Ekanayake, the Computer Science research coordinator, University of Colombo School of Computing for the support in carrying out the research.

Thank you very much for all members of Language Technology Research Laboratory (LTRL) for providing me required resources and giving me honest feedback about the research.

Many thanks to my beloved mother and my dear father for always being my strength, showing me the correct direction, and making me who I am today. Finally, I would like to appreciate the support of all the volunteers and my friends who helped me to complete this research successfully.

Table of contents

List of figures

List of table

List of Acronyms

BASILISK Bootstrapping Approach to SemantIc Lexicon Induction using Semantic Knowledge.

CRF Conditional Random Fields.

DIPRE Dual Iterative Pattern Relation Expansion.

ML Machine Learning.

NER Named Entity Relations.

NLP Natural Language Processing.

POS Part Of Speech.

TF-IDF Term Frequency–Inverse Document Frequency.

Chapter 1

Introduction

1.1 Background to the Research

Human inventions have been artificial since the beginning of human evolution. Usually other people had to adapt to use new tools that someone invented. But After millions of years, now people are getting the sense that nature is more capable of doing things and human inventions are way primitive than the natural inventions. So instead of adapting ourselves to new technologies, now the trend is to adapt technology according to natural behaviours and learn from those phenomena.

We can agree that Machine learning is one of the key studies which direct dull old rule based technologies to the machine intelligence era. Even though the human language is also a human creation, it is less artificial than typing all we want to do on a keyboard with some command language. That's why we call human languages as natural languages. Natural language processing is one of the hottest topics in time. There are lots of commercial applications using NLP technologies like Digital assistants, Speech to text or text to speech apps, OCR documentation apps etc. So it needs huge annotated data collection to use in those applications. Non-Latin western languages have a huge collection of annotated data but Indic languages still don't have that amount of annotated data.

This research falls under the natural language processing category and the named entity extraction subcategory. Named entity recognition task was introduced during the 6th Message Understanding Conference (MUC) [1], and in MUC- 7 [2]. As defined in the conferences, there are few prominent named entity categories. They are person, organization, location, time expression, and numeric expression. But the concern of this research is not to extract all kind of named entities from given text. The main idea of the research is to increase named entity corpus in two categories (Person names, Location names) by using a semi-supervised bootstrapping method.

1.2 Research Problem and Research Questions

The Sinhala language is lack of categorized data to do experiments on NLP area. Supervised learning methods need lots of training data and unsupervised learning methods are not that accurate. So the aim of the research is to check whether a semi-supervised bootstrapping method can be used to increase Sinhala named entity data with the use of different word features.

Questions.

1. What types of methods are used in linguistic bootstrapping?
2. What are more suitable bootstrapping methods for Named Entity categorization?
3. How do those methods should change to use in Sinhala Named Entity categorization?
4. What are the practical word features that can use in Sinhala corpus?
5. What are the problems that will happen when implementing the system with those word features?

1.3 Research Aim and Objectives

The main Intention of this research project is to classify named entities from unstructured documents using existing seed data list. So the project aims to find a proper bootstrapping solution for extract Sinhala named entities and increase the corpus. The following objectives will be achieved throughout this project.

- Review literature to identify different approaches, tools and technologies of Bootstrapping.
- Identify common Sinhala word features and sentence structures to find if existing bootstrapping methods are suitable for categorizing Sinhala named entities or have to change them to suit.
- Apply, found method to the corpus to find more named entities from decided categories.
- Evaluate the outcomes to know how successfully achieve the intended task.

1.4 Justification for the Research

When we consider the linguistic data applications, major languages have lots of data pools to use in those applications. But languages like Sinhala and Tamil have lack of data. So to increase data pools we can use cyber data and few semantic information extraction methods. There are two main types of semantic relationship extraction methods; traditional and open information extraction methods.[3]

Open information extraction methods extract all possible relationships from data and because of that, we can't use them to find exact accurate data. Ruled based, Supervised, Semi-supervised and Distantly supervised methods are those traditional information extraction methods. The rule-based method shows high precision and low recall. But because of using hand-made rules, it's hard to maintain. In Supervised learning, we need a training dataset and types of relationships are also limited.

Bootstrapping is a semi-supervised learning method and It takes advantage of unlabeled data. Because of leveraging unlabeled data and usage of few seed instances of known relationships, bootstrapping is more suitable for extract patterns from complex languages which are lack of tagged data.

Because of that, even though named entity categorization for Sinhala was already done by several researchers using supervised learning and statistical methods, this can be used to increase categorized data with very less effort compare to them.

1.5 Methodology

Since the general purpose of this research is to quantification of data and use those to generalizations of results from a sample to an entire population of interest, we can categorize this research as quantitative research. Here we use bootstrapping which is a semi-supervised learning method, to grow Sinhala named entity corpus.

We can define bootstrapping as a method for harvesting "instances" similar to given "seeds" by recursively harvesting "instances" and "patterns" by turns over corpora using the distributional hypothesis (Harris, 1954).

So, the performance of bootstrapping algorithms also depends on the selection of seeds. Although various bootstrapping algorithms have been proposed, randomly chosen seeds are usually used instead. Kozareva and Hovy (2010) [4] reports that the performance of bootstrapping algorithms depends on the selection of seeds, which shows the importance of selecting a good seed set. So they proposed a seeding framework that works iteratively and ranks the goodness of seeds in response to current human labelling and the characteristics of the dataset. So few initial human seeding methods were suggested by researchers and one of the famous ones is selecting the most frequent words

for each relation manually. This study is also using that seeding method for the initial run.

There are many bootstrapping methods that were introduced by researches such as Snowball, Basilisk, NOMEN, AutoSlog-TS, MetaBoot etc. From the literature review, we will find the most suitable bootstrapping method to grow named entities. Then we are going to test that algorithm using a few types of word features and evaluate it.

1.6 Outline of the Dissertation

The rest of this thesis is organized as follows: The second chapter is dedicated to the literature review to discuss related work on different methods which are used to do bootstrapping. The third chapter elaborates the design and the methodology. The potential ways in which this research could be conducted have been discussed in this chapter. The fourth chapter discusses the implementation phase of this project. This chapter describes research challenges, implementation strategies, proposed solutions, etc. Evaluation phase comes under the fifth chapter. It consists of the main evaluation strategies used in the study and evaluation results. The last chapter presents the conclusion and future work. It mentions the future advances possible with the results of this study.

1.7 Delimitations of Scope

- The research considers only Sinhala text and tries to categorize words to three classes(person, location, non name).
- Not all kind of Sinhala text will be processed. This is only for some newspaper articles which are in the news category.
- Single words that can be recognized by considering only itself are categorized as a person, location or non-name.(BIO encoding is not using)
- Since stemming and lemmatization mechanisms for sinhala words are still not very accurate, all forms of names(not only base words or root words) that belongs to each category, recognize as name entities.
- Few other named entity relation extraction methods were tested with Sinhala articles by some other researchers and achieved good accuracy. But Concern of this research is not to achieve an accuracy more than those methods. The main idea is to test whether the bootstrapping method can categorize Sinhala named entities in an average accuracy. (Because named entity extraction using bootstrapping method was tested with few other Latin and also non-Latin languages and succeed.)

- Since word features like POS tags and morphological tags are also still not that accurate for Sinhala text, research will continue using simple word features like n-gram features, word suffix features and context word features.

1.8 Summary

Increasing Sinhala named entity data using less effort compared to supervised learning is the main intention of the study. It is going to achieve by a semi-supervised bootstrapping method. In order to use in this method, practical Sinhala word features should be found. The scope of the research is limited to identify only single words that belong to person and location names.

Chapter 2

Literature review

2.1 Brin's method

In this research, we are interested in extracting relations from Unstructured data sources and use those relations to find more related words. So the first system that uses bootstrapping for Relational Extraction was introduced by Brin (1999) [5]. He presented a technique which use the connection between sets of patterns and relations to grow the target relation starting from a small sample. This method gives two entities and finds words before the first entity (BEF), words between the two entities (BET), and words after the second entity (AFT) to generate extraction patterns by grouping contexts based on string matching. They used book author relationship to the research and their general goal was to be able to extract structured data from the entire World Wide Web by leveraging on its vastness. Their method(DIPRE) has proven to be a remarkable tool in the simple example of finding lists of books. The initial sample set of that test was 5 books and it expanded to a high-quality list of over 15,000 books with very less human intercession. They thought that the same tool may be applied to a number of other domains such as movies, music, restaurants, and so forth. Also, they predict a more sophisticated version of this tool is likely to be able to extract people directories, product catalogues, and more. But that method did not stop from there.

2.2 Snowball

Agichtein and Gravano (2000) [6] developed Snowball, which is inspired by Brin's method of collecting three contexts for each occurrence, but computing a TF-IDF representation for each context. The seed contexts are clustered with a single-pass algorithm based on the cosine similarity between contexts using the three vector representations. It Scores the patterns and ranks the extracted instances to control the semantic drift. They experimented on finding organization-location pairs over 300,000 newspaper articles. The only input to the Snowball system during the evaluation on the test collection were

the five seed tuples. They evaluated their method with two methods; DIPRE and Baseline. The result was Snowball and DIPRE shows significantly higher precision than Baseline and Snowball have at all occurrence levels significantly higher recall than DIPRE and Baseline do.

2.3 Bootstrapping semantic word classes

However, separating named entities into few classes is the concern of this research. So when we seeking about separating words into semantic categories, MetaBoot [7] algorithm is one of the initial bootstrapping methods. MetaBoot is a multilevel bootstrapping algorithm that generates both the semantic lexicon and extraction patterns simultaneously. As input, this technique requires only unannotated training text and a handful of seed words for a category. It was originally designed for semantic word categorization tasks. It is seeded with words belonging to the desired class and then creates extraction patterns by instantiating templates that extract every noun phrase in the corpus. The patterns are scored based upon the number of seeds extracted. The best pattern is saved and then the head noun from every extracted noun phrase is accepted into the category. The patterns are then re-scored and the cycle repeats. After each cycle completes, all nouns put into the dictionary during the cycle are scored based on the number of patterns that extracted it. Only the five best are allowed to remain. There are other methods like NOMAN which described by Yangarber et al. (2002). It required significant preprocessing part before choosing seeds. This algorithm was used to learning generalized names in the biomedical context. AutoSlog-TS introduced by Riloff and Wiebe (2003) Identify subjective expressions. All these algorithms and more bootstrapping approaches [8] [9] [10] were briefly explained in the paper ”A survey of bootstrapping techniques in NLP “– Deniel Waegel [11] .

2.4 Basilisk method

Even described methods are the initial bootstrapping methods for semantic relation extraction, Basilisk method is more related to the proposed solution. The Basilisk algorithm (Bootstrapping Approach to SemantIc Lexicon Induction using Semantic Knowledge), developed by Thelen and Riloff(2002) [12]. It begins with an unannotated corpus and a list of seed words for six semantic categories, which are then bootstrapped to learn new words for each category. They seeded their algorithm by sorting the words in the corpus by frequency and manually identifying the 10 most frequent words for each of the six semantic categories. After extract patterns, they used pattern scoring and candidate scoring methods in a similar way to other bootstrapping methods.

Apart from introducing the Basilisk method, these researchers have compared previously described MetaBoot algorithm with Basilisk algorithm. They used both Basilisk and MetaBoot to learn semantic lexicons for six semantic categories: building, event, human, location, time, and weapon. Both algorithms were run each semantic category at a time for 200 iterations so that 1000 words were added to each lexicon (5 words per iteration). All of their experiments involve the MUC-4 terrorism domain and corpus. The result was Basilisk outperforms meta-bootstrapping for every category, often substantially.

Not only that, but they also explored the idea of bootstrapping multiple semantic classes simultaneously. Their hypothesis was that errors of confusion between semantic categories can be lessened by using information about multiple categories. The problem of that hypothesis was word cannot belong to more than one semantic class. That is not actually right for every word. However, to take advantage of multiple categories they use this ”one sense per domain” constraint in their experiment. To achieve that, they had to use two simple conflict resolution rules. First one was, if a word is hypothesized for category A but has already been assigned to category B during a previous iteration, then the category A hypothesis is discarded. The second one was, if a word is hypothesized for both category A and category B during the same iteration, then it is assigned to the category for which it receives the highest score. So they ran both methods for multiple categories and result was an unexpected one. Improvement for meta-bootstrapping was much more pronounced than for Basilisk. Which means that Basilisk was already doing a better job with errors of confusion, so meta-bootstrapping had more room for improvement.

As they explained further, Simple conflict resolution helps the algorithm recognize when it has encroached on another category’s territory, but it does not actively steer the bootstrapping in a more promising direction. So more intelligent way to handle multiple categories is to incorporate knowledge about other categories directly into the scoring function. For that, they modified Basilisk’s scoring function to prefer words that have strong evidence for one category but little or no evidence for competing categories. After using that scoring function, the overall result was, this version of Basilisk performs best, showing a small improvement over the version with simple conflict resolution. Both multiple category versions of Basilisk were better than the multiple category version of meta-bootstrapping. So considering all the techniques that Basilisk used and the results it got, we can conclude that this algorithm is more suitable for our problem than other bootstrapping approaches.

2.5 Bootstrapping for NER

In this research, we are using named entities as semantic categories. When it's come to NER, gazetteers are more helpful. But ambiguities that happen when identifying the categories cannot resolve from those gazetteers like in bootstrapping. The closest research to our approach is "Semi-supervised Bootstrapping approach for Named Entity Recognition" [13] which was done in Anna University, Chennai. In that paper, they used the Basilisk bootstrapping method for identifying named entities. So they presented a semi-supervised NER approach that starts with identifying named entities with a small set of training data. The set of training documents is manually annotated with the categories of named entities. The annotated training set is previously processed by identifying the characteristics of the word. Since they do bootstrap both English and Tamil languages, they used POS tag and semantic constraint (SC) as English word features and add the morphological suffix (MS) as an extra feature for Tamil language words. They stated that when considering a three window context, the ambiguity of the type of named entity occurs and to overcome the issue they went for a five window context (wi-2, wi-1, wi, wi+1, wi+2). The Named Entity Types used there were defined in the classification of MUC-7 namely person, organization, location, date, time, money and percentage.

Using the identified named entities, the word and the context features are used to define the pattern. This pattern of each named entity category is used as a seed pattern to identify the named entities in the test set. The score of a pattern and the score of tuple value enables the generation of new patterns to identify named entities in the test set. They have evaluated the proposed system for the English language with the dataset of tagged and untagged named entity corpus and for the Tamil language with the documents from the FIRE corpus and yield an average f-measure of 75% for both the languages. So we also can use the same kind of approach with different word features. Because we don't have a perfect morphological tagger or POS tagger in the Sinhala language, we can test different n-gram features and Clue Words as alternatives.

2.6 Sinhala NER

Since the research is to increase Sinhala linguistic corpus, it's better to explain few before attempts of building Sinhala corpus. First one is called as UCSC Text Corpus of Contemporary Sinhala consisting of 10 million words [14] which was done in Language Technology Research Laboratory, UCSC. The corpus represents the modern usage of Sinhala in three categories (Creative writing, Technical writings, News reportage). They used Parts of Speech and Morphological methods to classify words into five classes namely, Postpositions, Particles, Conjunctions, Determiners and Interjections. At the initial stage,

a substantial effort was made to train the manual classifiers to classify words according to the predefined set of classification criteria. In order to automate this process, the high-frequency words were first classified into their respective parts of speech and then certain word ending patterns peculiar to each class were identified. These patterns were used to classify the rest of the list automatically by running regular expression matching followed by manual cleaning. But the problem of this solution is, it covers very few Sinhala sources and it is not updated.

To resolve that problem group of researches in University of Moratuwa introduced a solution name Sinmin [15]. The developed a corpus, which is continuously updating, dynamic and covers a wide range of topics for the Sinhala language. They used news, academic, creative writing, spoken and gazette as five source categories. Sinmin contained about 119 million words belonging to the time period of 2004-2014. So this research might help someone to access a large corpus. But Named entity categorization was not the main idea of both corpora.

For that, we can find the research paper "Named Entity Recognition for Sinhala Language" [16] which was written by J.K. Dahanayaka and A.R. Weerasinghe in University of Colombo School of Computing. Since there had not been much previous work based on NER for Sinhala in that time, the concept and the needed resources had been built from scratch. That paper tried to find out the effectiveness of using data-driven techniques to detect Named Entities in Sinhala text. Conditional Random Fields (CRF) and Maximum Entropy (ME) model were applied to this task. It is found that the former outperformed the latter in all experiments. A CRF model is able to detect Sinhala Named Entities with very high precision (91.64%) and reasonable recall (69.34%) rates. However, in this research, they only tagged words as named and non-named entities.

There are Later attempts with the same approaches CRF and ME, named "Conditional Random Fields based Named Entity Recognition for Sinhala" [17], in 2015 and "Ananya - A Named-Entity-Recognition (NER)System for the Sinhala Language" [18] in 2016. Both classified named entities into three classes as Person, Location and Organization. Second paper focused on identifying the effectiveness of using data-driven techniques along with possible combinations of language features in detecting Named Entities in Sinhala text. They experimented on different linguistic features such as orthographic word level and contextual information that are effective with both CRF and ME Algorithms. Their conclusion was that the most suitable feature set that is beneficial in detecting Name Entities in Sinhala text is the combination of Clue Words (ex-MAH-ha-TH-aa, MEH-na-wi-ya), Context words (window size = 1), n-gram (n-gram size = 10), gazetteer, start-end word and word length feature with a cut off value of 5. Importantly they showed that the percentage of accuracy is significantly higher when using n-gram with other features. So we can consider using a few of those language features (special n-gram feature) for preprocessing part of our research. As far as We found out,

this is the state of art research in Sinhala named entity tagging.

There was another survey paper which is very resourceful to find more details about named entity tagging. It is a survey of named entity recognition and classification [19] between period 1996 to 2006. It consists of how the language factor [20] [21] [22], domain factor and entity type factor in researches changed between that period. It consists what were the learning methods [18] that used to name entity classification. And also it contains what were the features [23] [24] that used to classify named entities. Lots of related research papers were also found on the reference section of this survey paper.

2.7 Summary

So as the summary, there are lots of bootstrapping methods to extract relationships from an unannotated corpus and few of them improved to extract predefined semantic categories from given seed lists and domain corpus. From discussed methods, Basilisk algorithm is more related to our problem and they proposed a method to extract multiple semantic categories simultaneously. There was another research which used the Basilisk algorithm for named entity extraction and yields an average f-measure of 75% for both English and Tamil languages. So that can be used as one of our main reference researches. There were few attempts to do Sinhala named entity tagging with Conditional Random Fields and Maximum Entropy models. One of them used Clue words, Context words, n-grams, gazetteers and word length as feature categories and showed that n-gram feature can use to improve the accuracy of results.

Chapter 3

Design

As discussed in the upper parts of the thesis, bootstrapping method of this research is based on Basilisk algorithm. Figure 3.1 shows the main phases that included in the research design.

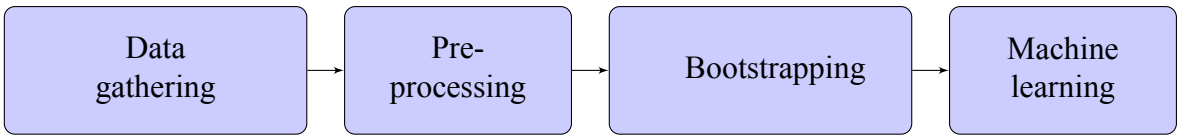

Figure 3.1: Main phases of the research

3.1 Data gathering

Since Sinhala Named entity categorization research that used a bootstrapping method was not conducted before, in this research we are going to check whether if bootstrapping can actually get accurate results in Sinhala NER. So all kind of sources are not covered in data gathering part. We selected an online newspaper “Sunday Lankadeepa” and 700 articles that are in the news category(because news articles tend to contain more named entities)

3.2 Pre-Processing

3.2.1 Data cleaning

After data gathering, the data cleaning process can be done in a few steps. They are removing special characters, numbers, zero width joiners(because Sinhala words contain zero width joiners and they can cause troubles in text processing), unwanted spaces and tokenize the corpus. Finally, remove stop words from the processed corpus. Stemming or lemmatization will not perform because of previous mention limitations. After that,

the cleaned text should be sent to seed selection.

3.2.2 Seed selection

One of the critical tasks in a bootstrapping process is selecting seed words for each category. So this seed selection needs to be done carefully because it can cause heavily on the end results. Proposed solution divides named entities into two categories which are person names and location names. So initially, two seed lists are needed to be selected from the corpus. That process is done by sorting all the words of the corpus by there frequency and manually selecting the most frequent N words for each category. A bootstrapping framework that works iteratively and ranks the goodness of seeds in response to this initial seeding and the characteristics of the dataset, will continue extracting new seeds to initial lists.

3.2.3 Building pattern corpus

Before bootstrapping process begins, a Pattern extraction from the corpus should be done. Normally it can be done using a morphological parser, POS tagger, n-gram feature extractor or any available word feature extractor. Since there is no well-developed parser or POS tagger for Sinhala, it has to be done in word level. In order to get an idea about Sinhala word level features, following are some examples of the analysis.

Some word frequencies of corpus before pre-process
මහතා: 0.5%, මහත්මිය: 0.03%

Some Sentences from corpus before pre-process
- වැටුප් විෂමතාවක් ඇතිවන බව නියෝජ්‍ය ඇමැති අශෝක් අබේසිංහ ”මහතා” කියයි -
- දකුණු පළාත් ආණ්ඩුකාරවරිය වූ කුමාරි බාලසූරිය ”මහත්මිය” සමග -

As suggested by previous researches that were mention in the literature, “මහතා”, ”මහත්මිය” and other post-words can be features for a pattern, because there are considerable frequencies of them in the corpus and previous words of them are most of the time person or location name. Pre-word can also be helpful because there can be some unidentified relations between previous word and current word. So even though Pre-words seems not that useful, in this research both pre-words and post-words are considering as context word features to a pattern.

In sinhala language suffixes can be very helpful to identify named entities. Following

are few examples to location name suffixes.

Suffix -ගල

කුරුණෑගල, මොනරාගල , රන්දෙණිගල, දිඹුලාගල, බෝගල, පිටිගල, තලගල, රංගල, නෙල්ලිගල, කොග්ගල etc.

Suffix -ගම

මහරගම, බියගම, කතරගම, හෝමාගම, අලුත්ගම, ආසිරිගම, තඹුත්තේගම, මාවතගම, වැලිගම, ගොඩගම etc.

There are more Sinhala(Srilankan) location suffixes like -පිටිය, -වල, -වෙල, -ගොඩ, -මුල්ල, -පුර, -දෙණිය, -වත්ත etc.

Most of Sinhala person names are also contain common set of suffixes. Following are some examples to person name suffixes.

-සිංහ, -වර්ධන, -සූරිය, -පාල, -නායක, -රත්න, -සේන, -දාස, -වතී, -සිරි, - තුංග etc.

So these are the word features that will select to build patterns.

- Context word (window size = 1)
  Previous researches recommend it gives more accurate results with window size 1
- Word suffix (size = 4,5)
  Sinhala suffixes can be average 4 or 5 character long(with UTF-8 Unicode)

In order to capture these features, a pattern learner should run over the corpus which generates patterns for every word phrase in the corpus. In the original Basilisk approach, they first identify nouns from the corpus and then run pattern extractor to generate patterns for only noun phrases. But since there is no accurate noun identifier for Sinhala language, patterns should be generated for every word in the corpus.

Structure of a pattern define as follows.

Current_word * <Pre-word>:<Suffix_of_the_current_word>:<Post-word>

After that, all patterns should be divided in to training and testing sets. Since data set is not very large, 90% of data use as training set and 10% data use as testing data. In order to fine tune the model we are going to use K-fold cross validation for training set.

3.3 Bootstrapping

The bootstrapping process begins with two seed lists(person and location) and pattern corpus which was selected as the training set. In order to avoid implementation complications, the whole bootstrapping process is designed to do separately for each category. So now on, the design will be described for a single category.

In the first step, the system has to select all the patterns from training pattern set that contain seed word as the current word in each pattern. So Intention is to identify an initial subset of patterns that belong to the relevant named entity category. After that, these extracted patterns are used to find more similar patterns from pattern corpus. The similarity between the two patterns are calculated by considering their pre-words, post-words, and suffixes. As an example, consider the following two patterns.Features of these patterns are starting after the * sign. They are in order pre-word, suffix, and post-word.

- Original pattern - වීරසිංහ * <ගාමිණී>:<රසිංහ>:<මහතා>
- New pattern - ජයසිංහ * <මධුර>:<යසිංහ>:<මහතා>

By analyzing all patterns that generated and using native Sinhala knowledge, we can identify that named entity words are more related to their post-word than previous word. So when scoring the previous word similarity and post-word similarity, post word score should be weighted than previous word. As analyzed in previous section suffix feature is most important feature from all. So weight of the score should be higher than other two features. In this example, post-words of both patterns are equal and suffixes are partially equal. Pre-words are entirely different. According to that similarity, a scoring algorithm will generate a score for the new pattern. Sometimes that can be more than one original patterns which will score the same new pattern. In that case, only the highest score will assign to the new pattern. So new patterns keep their score and id of original pattern that gave the highest score. Original patterns keep count of new extracted patterns by itself.

After that, patterns that scored more than some predefined score, will be selected as candidate patterns. Current words that contain in those patterns will be added to the seed word list for next round. If those candidate patterns are selected by original patterns that select number of N minimum new patterns, then those candidate patterns will be added as initial patterns of next round. (Intention of this is to select the most capable patterns from scored patterns) Next round will continue with new_seeds+old_seeds and set of initial patterns. This process will continue looping until a maximum predefined loop count or no more new patterns are identifing. After bootstrapping process stopped, final seeds will be added to seed list. Final patterns will be tagged with the relevant named entity and keep in a file. After doing this bootstrapping process for both named entity categories, result patterns with their named entity tag will be sent to supervise learning model. Following figure 3.2 shows the process of bootstrapping for single named entity category.

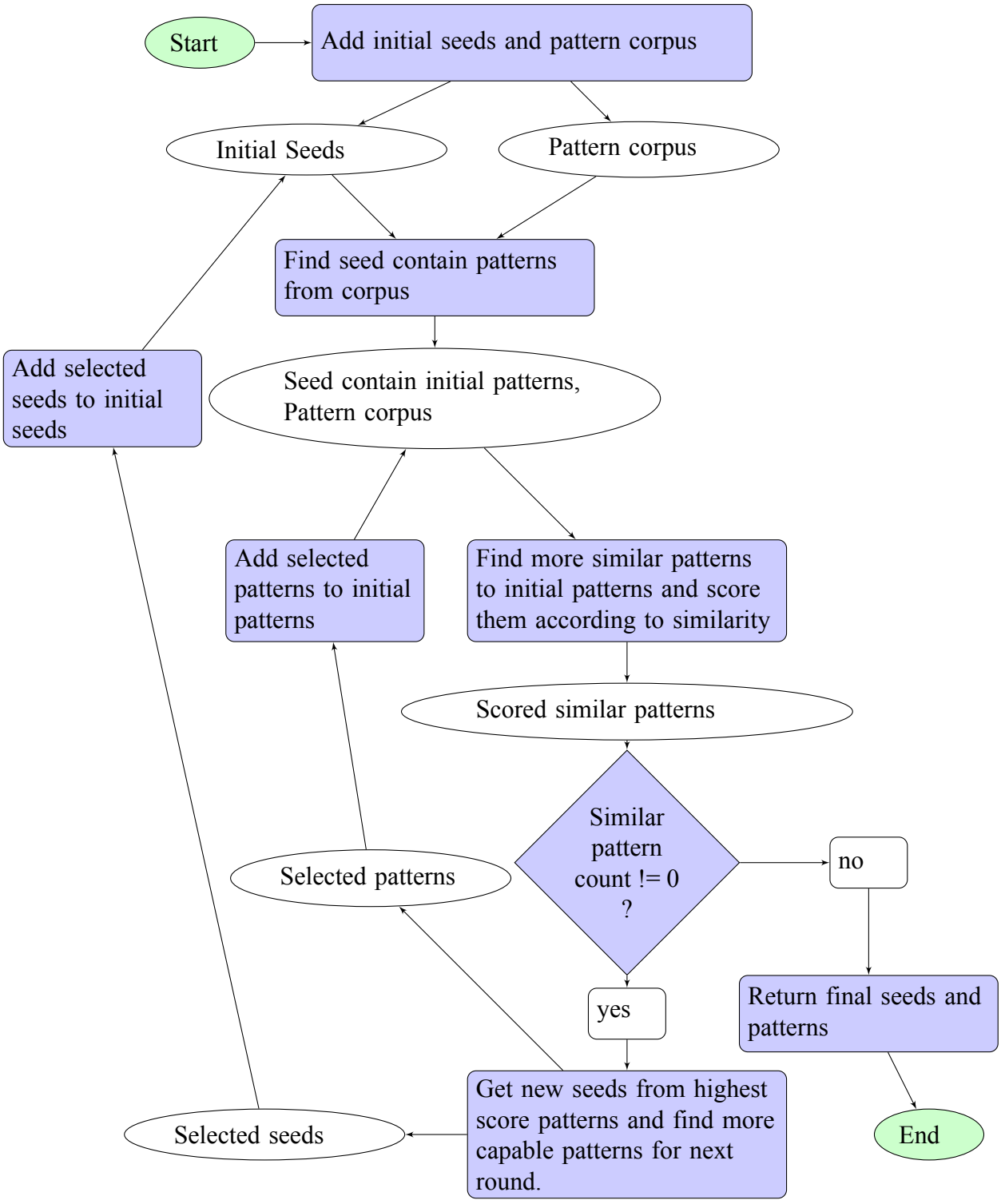

Figure 3.2: Bootstrapping process for a single Named entity category

3.4 Machine learning

In order to get better accuracy in a supervised learning model, it needs human tagged data set to train the model. But the intention of the proposed solution is to find a way to reduce human contribution and get an average accurate result. So supervise learning

model of the solution use data set that automatically tagged by the bootstrapping process. Even though it can be less accurate than human tagged data, for proposing ML models it is assumed to be 100% accurate.

Machine learning part will continue with hierarchical manner, based on the rational idea that person and location names are closely related and non-names are more distantly related with location and person names. So two models will be designed and first one is to predict name and non-name entities from test patterns. Second one will get only named entity patterns that predicted by first model and predict person and location name patterns out of it. Both models will be trained by patterns and their labales which provided by bootstrapping module.

Following figure 3.3 shows how machine learning models work.

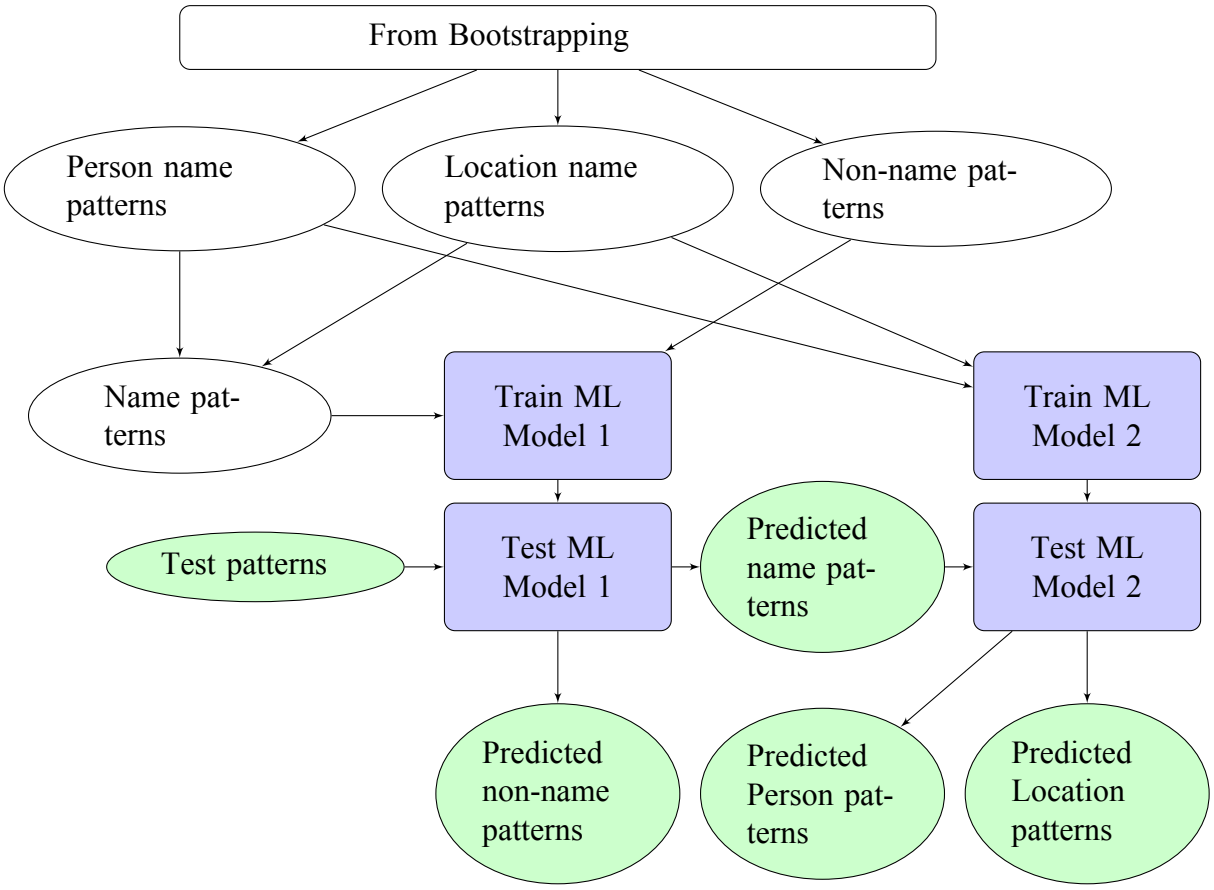

Figure 3.3: Architecture of Machine learning models

Chapter 4

Implementation

This chapter explains the steps taken in implementing the bootstrapping method proposed in Chapter 3 and describes the various tools used. Section 4.1 describes the software tools utilized for implementation, Section 4.2 presents the implementation details of the classes and functions.

4.1 Software Tools

In data gathering part, Octoparse(a web Scraping tool) tool is used to extract news article texts from Lankadeepa online web site. The proposed solution was implemented using python 3 with anaconda environment.Codecs and re python packages were used in pattern extraction and data cleaning processes. Machine learning part also implemented using python libraries which are pandas and scikit-learn.

4.2 Implementation details

4.2.1 Data cleaning and Pattern extraction

Both data cleaning and pattern extraction tasks are implemented in the below function. This function calls a custom function textFilter to remove non character symbols and zero width joiner characters. Then it filter the unwanted spaces and tokenize the text. After that it removes the provided sinhala stop words and extract patterns from result corpus. Finally, it writes extracted patterns to a file.

```
import codecs
import re

def text_cleaning_and_pattern_extraction(textfile,stopwordfile\
                                         ,output,suffix_size):
    stopwordlist = []
    with open(stopwordfile, encoding='utf-8') as s:
```

```
        for stopword in s:
            stopword= stopword.strip()
            stopwordlist.append(stopword)

    with open(textfile, encoding='utf-8') as f:
        for line in f:
            outputfile = codecs.open(output, \
                                encoding='utf-8', mode='w+')
        # Filter all non character symbols
            filtered_text = textFilter(line)
        # Filter unwanted spaces and tokenize
            wordlist = re.findall(r'\S+', filtered_text)
        # Remove stop words
            wordlist = [i for i in wordlist if i not in\
                        stopwordlist]
        # Extract patterns
            for w1,w2,w3 in zip(wordlist, wordlist[1:]\
                                , wordlist[2:]):

                current_word_suffix=w2[-suffix_size:]

                outputfile.write\
        (w2+" * <"+w1+">:<"+current_word_suffix+">:<"+w3+">\n")

    s.close()
    f.close()
```

4.2.2 Implementation of bootstrapping module

As mentioned in chapter 3, the percentages for training and testing are 90% -10%. So 700 article set divided as 630 articles for training and 70 articles for testing. Patterns of first 630 articles are used by bootstrapping process and output of that is used to train ML models. Patterns of final 70 articles are tagged by humans for the testing purpose.

The bootstrapping method is implemented with four submodules.All modules are implemented for extract named entities which belongs to a single named entity category at a time.

Module 1

Seed list and all pre-processed patterns are taken as inputs and seed contain patterns will be selected by this module.

Module 2

All initial patterns and seed contain patterns are taken as inputs and more similar patterns with seed contain patterns will be found by this module. As described in the chapter

3, this process is done by calculating a score for the similarity of patterns and it considers pre-word, suffix and post-word similarities in a pattern. Pre-word and post-word are scored in a manner that if all characters of words in both patterns are matching then it will get a predefined score and if any character is not matching then it will get score 0. Suffixes are scored comparing character by character from the right end of the suffix. The score will increase one by one for matching characters until a mismatch occurs. Below function shows the suffix score function.

```
def suffix_similarity(extracted_pattern, original_pattern):
    score=0
    e_list=list(extracted_pattern)
    o_list=list(original_pattern)

    reverse_e_list=e_list[::-1]
    reverse_o_list=o_list[::-1]

    if len(e_list)>= len(o_list):
        min_list=o_list
    else:
        min_list=e_list

    for char in range(0,len(min_list)):
        if reverse_e_list[char] == reverse_o_list[char]:
            score= score+1
            if char== len(min_list)-1:
                return score
        else:
            return score
```

This module will also calculate new pattern generation frequency by each initial pattern and finds what is the exact initial pattern that score particular new pattern. Then sends both outputs and new pattern score list for module 3.

Module 3

This takes outputs of module 2 as inputs and finds highest scored seeds and most capable patterns by analyzing inputs.

Module 4

If this module receives new patterns from module 3, extracted seed list and pattern list are sent to next round. If there are no new patterns, selected seeds are written to a text files. All final patterns with relevant label(person, location or non-name) also written to a text file.

All four modules are managed by the main module. After stopped the bootstrapping process, files that contain patterns with there labels will be referred by machine learning models.

Before implementing the machine learning model, test patterns should be labeled by humans. In order to label patterns easily, simple named entity tagger web app was built and users were instructed about labeling process. Following figure 4.1 shows the interface of the app.

Figure 4.1: Named entity tagging app

After test pattern set were labeled, files that contain patterns with there labels will also be referred by machine learning models.

4.2.3 Machine learning models

As mention in chapter 3, machine learning will be done in a hierarchical manner. There are two machine learning models. ML model 1 predicts patterns as name and non-name. ML model 2 separate predicted name patterns to person and location categories. So both models are using binary classification. As most popular and effective classifiers [25], SVM or logistic regression can be used for the job easily. Choosing one of them is considered by the nature of the feature vector. Since we use CountVectorizer to make vectors, feature vectors of these patterns can be very sparse.(Because the number of all unique patterns are large) Because of that reason, we can hope that logistic regression will perform better than SVM. However, both methods were tested and logistic regression showed more accurate results. So the following description is about models that used logistic regression for the classification.

ML model 1

The output of the bootstrapping process was two pattern lists(person, location) with relevant named entity tag. In order to prepare the training set for ML model 1 these two pattern list should be combined and all labels that are "person" and "location" should be changed as "name". That combined pattern file is used as positive patterns to model 1. Negative pattern file will be generated by selecting the same amount of non-name patterns from all pattern file and labeling them as "non-name". (Randomly chosen patterns that are in all pattern list and not in positive pattern list are defined as non-name patterns) Testing set is generated by the human labeled golden standard.

ML model 2

Labeled person and location patterns which are outputs of the bootstrapping process, directly use to train ML model 2. Here also testing set is from the golden standard.

Both models are implemented with the same code segments. Following lines show how training set and testing set will be prepared.

```
 # making data frames from files
df_train = pandas.read_csv('train_set')
df_test = pandas.read_csv('test_set')

# shuffle pattern set
df_train= df_train.sample(frac=1).\
                reset_index(drop=True).dropna()
df_test= df_test.sample(frac=1).\
                reset_index(drop=True).dropna()
```

```
# seperate data and labels
x_train = df_train.iloc[:,0]
y_train = df_train.iloc[:,1]
x_test = df_test.iloc[:,0]
y_test = df_test.iloc[:,1]
```

K-fold cross-validating is used to validate the train set. Among the CountVectorizer's configurations, the n-gram feature is tested for several n values and bi-gram showed the best accuracy. Basic configurations that used in Countervectorizer and LogisticRegression functions is as follows.

```
# Train
kfold = model_selection.KFold(n_splits=10,random_state=7)
vectorizer = CountVectorizer(analyzer='char',\
                    ngram_range=(1, 2), max_df=1.0,\
                    min_df=1, max_features=None)

fitted_vectorizer_train = vectorizer.fit_transform(x_train)
fitted_vectorizer_test = vectorizer.transform(x_test)

cross_validated_result = model_selection.cross_val_score\
    (LogisticRegression(), fitted_vectorizer_train.toarray() \
                                        , y_train, cv=kfold)

# Test
clf = LogisticRegression(random_state=0, solver='lbfgs',\
                    multi_class='multinomial', max_iter=100)
clf.fit(fitted_vectorizer_train.toarray(), y_train)
y_pred = clf.predict(fitted_vectorizer_test)
```

So after the complete process is run, outputs are two extracted word lists that belong to the person and location categories and four predicted pattern lists that belong to name, non-name, person and location categories.

Chapter 5

Evaluation and Results

This chapter elaborates how results are evaluated and the success level of the proposed solutions. Section 5.1 describes what are the selected evaluation methods and rationality behind it and section 5.2 elaborates final evaluation results.

5.1 Evaluation method

Evaluation for the proposed solution can be done in two phases. First one is evaluating bootstrapping process before doing supervised learning and the second one is evaluating bootstrapping + supervised learning processes at the end.

5.1.1 Phase 1 evaluation

Since non-names are not suggested by bootstrapping, the first evaluation is done by considering only person and location word outputs. The job of proposed Bootstrapping module is to extract named entities for person and location categories. So this can be evaluated by manually checking if the extracted words actually belong to suggested named entity category. The following equation 5.1 will calculate the precision of a single named entity category.

$$PS = \frac{TP}{TP + FP} \qquad \text{(equation 5.1)}$$

PS = Precision of a single named entity category
TP = Number of suggested words that actually belong to the category(True positive)
FP = Number of suggested words that do not belong to the category(False positive)

Precision of the whole bootstrapping process will evaluated by following equation 5.2.

$$PC = \frac{PP + PL}{2} \qquad \text{(equation 5.2)}$$

PC = Precision of whole bootstrapping process
PP = Precision of person named entity category
PL = Precision of location named entity category

Other common performance metrics like recall, F-measure and accuracy are not possible and also not needed to calculate in this phase. Because all training patterns are not labeled by humans and intention of evaluation in this phase is only to identify how much of extracted words are actually belong to suggested named entity category.

5.1.2 Phase 2 evaluation

Results of the bootstrapping process are used in supervised learning models and final results of supervised learning models are evaluated in this phase. Because of that, the complete system will be evaluated by this phase evaluation. As discussed in chapter 4, there are two machine learning models and the results of both can be evaluated by common performance metrics precision, recall, and f1-score. Other than that, the training set will be evaluated with 10 - fold cross-validation and accuracy of testing results will be also calculated.

Machine learning model 1 has two classes that are “name” and “non-name”. Machine learning model 2 has two classes that are “person” and “location”. If we give all classes a general name class A, the following equation 5.3, equation 5.4 and equation 5.5 define precision, recall and f1-score values for class A.

$$PA = \frac{TP}{TPFP} \qquad \text{(equation 5.3)}$$

PA = Precision of class A
TP = Number of words that predicted as class A and belongs to class A
TPFP = Number of all words that predicted as class A

$$RA = \frac{TP}{P} \qquad \text{(equation 5.4)}$$

RA = Recall of class A
TP = Number of words that predicted as class A and belongs to class A
P = Number of all words that belong to class A

$$FA = 2 * \frac{PA * RA}{(PA + RA)} \qquad \text{(equation 5.5)}$$

FA = F1-score of class A

5.2 Results

Following figure 5.1 and table 5.1 shows the results of intermediate bootstrapping results.

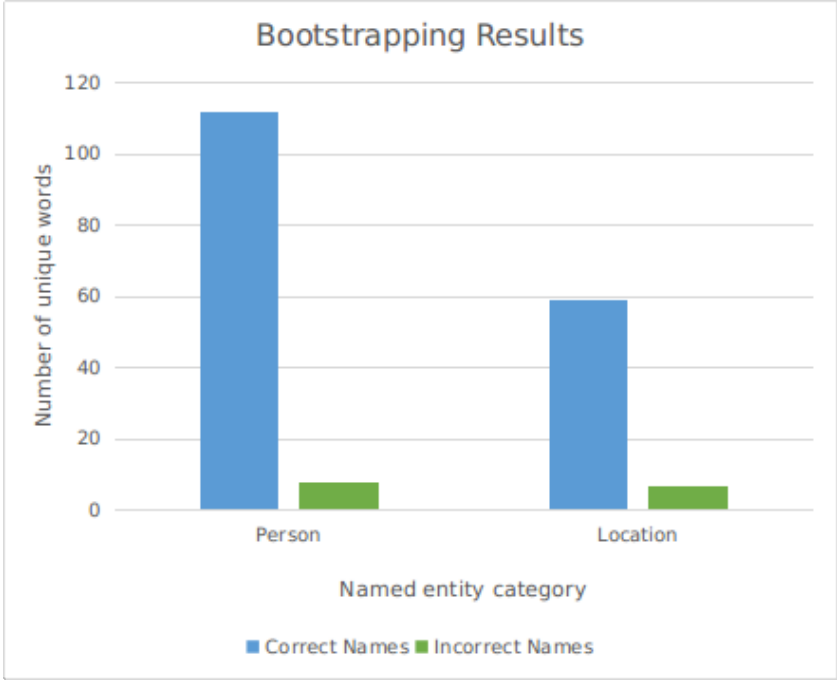

Figure 5.1: Number of unique words that extracted by bootstrapping

Table 5.1: Results of Bootstrapping process

Class	Precision
person	93%
location	89%
overall	91%

Following figure 5.2 shows precision, recall and f1-score values of both name and non-name categories and figure 5.3 shows the average results of machine learning model 1. Table 5.2 shows complete results of machine learning model 1.

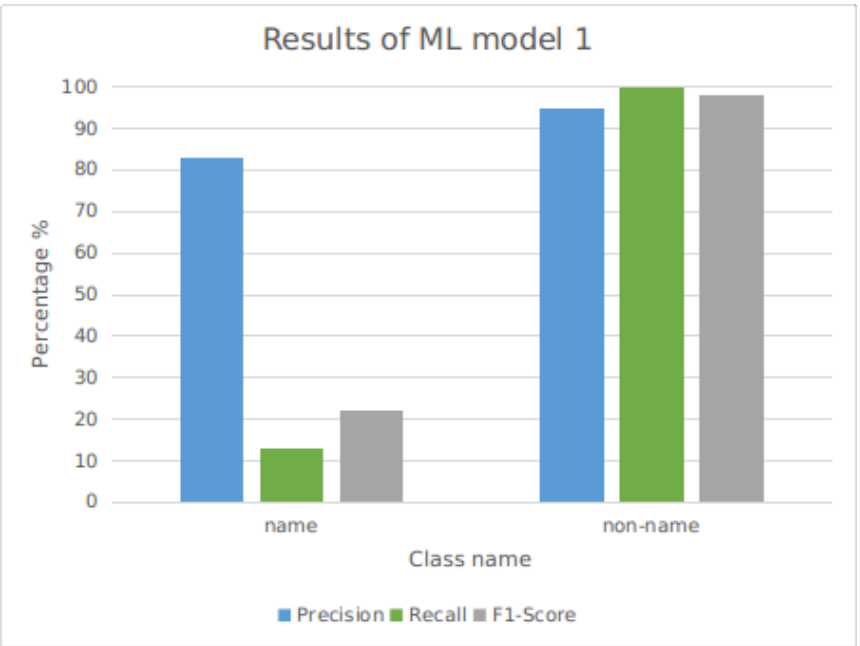

Figure 5.2: Name and non-name category results of machine learning model 1

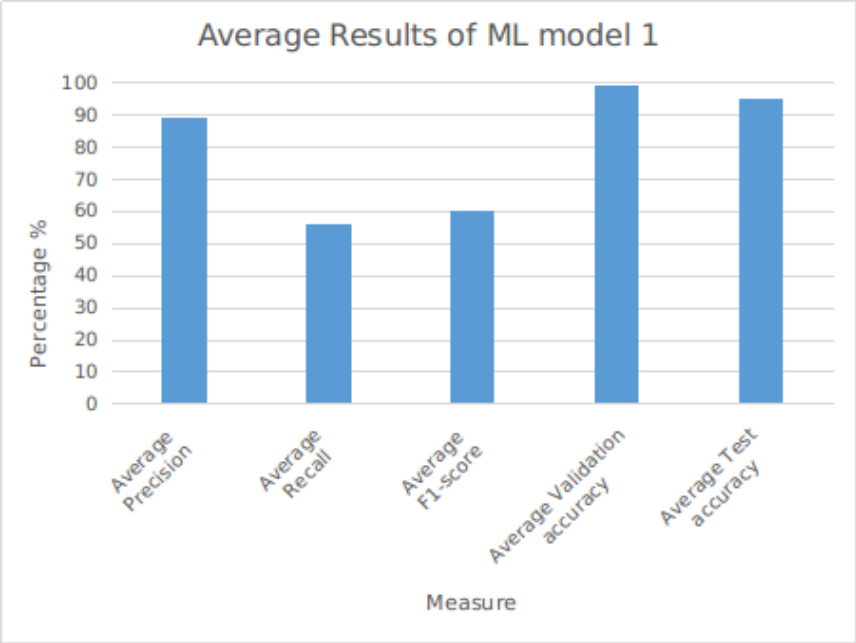

Figure 5.3: Average results of machine learning model 1

Table 5.2: Results of Machine learning model 1

Class	Precision	Recall	F1- Score	Cross validation accuracy	Test Accuracy
name	83%	13%	22%	99%	95%
non-name	95%	100%	98%		
Average	89%	56%	60%		

Following figure 5.4 shows precision, recall and f1-score values of both person and location categories and figure 5.5 shows the average results of machine learning model 2. Table 5.3 shows complete results of machine learning model 2.

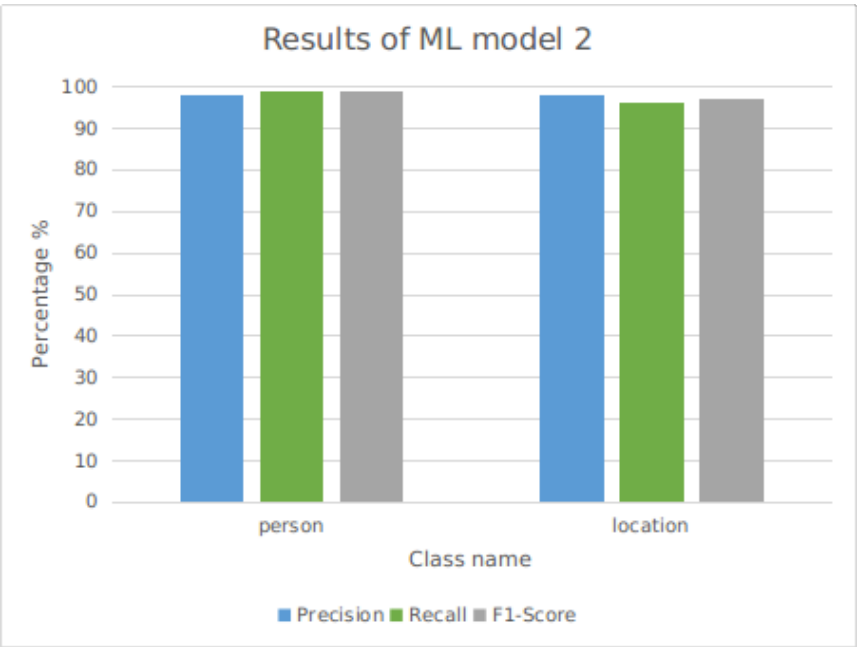

Figure 5.4: Person and location category results of machine learning model 2

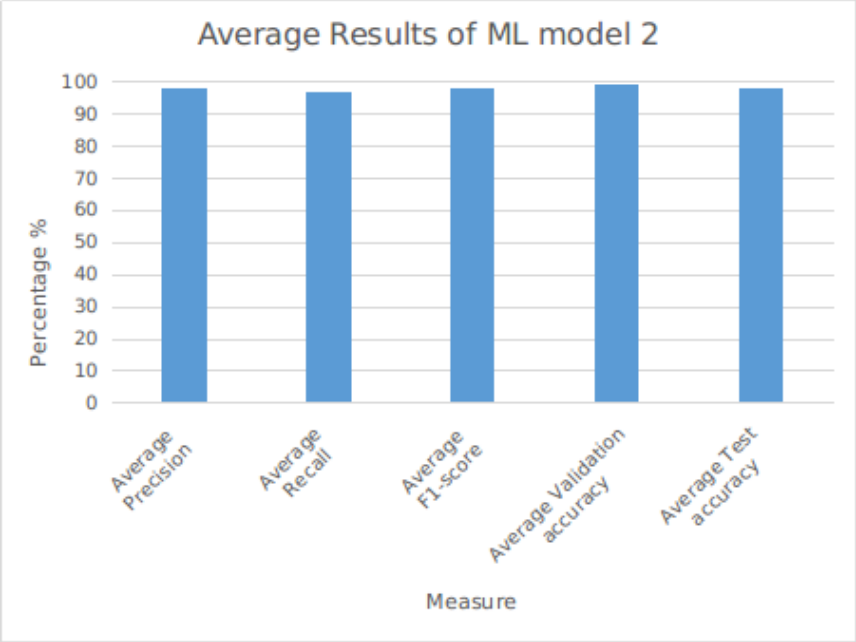

Figure 5.5: Average results of machine learning model 2

Table 5.3: Results of Machine learning model 2

Class	Precision	Recall	F1- Score	Cross validation accuracy	Test Accuracy
person	98%	99%	99%	99%	98%
location	98%	96%	97%		
Average	98%	97%	98%		

Chapter 6

Conclusion

This chapter includes a review of the research aims and objectives, research problem, limitations of the current work and implications for further research.

6.1 Conclusions about research problem and objectives

This research reviews how the idea of bootstrapping came into the linguistic area and how was it evolved with relation extraction. What were the initial bootstrapping methods and how were they changed to categorize semantic words. When considering the semantic word categorization, Basilisk was one of initials which begins with an un-annotated corpus and a list of seed words for few semantic categories and then bootstrapped to learn new words for each category. The same concept was used and achieved accurate results for the Tamil language named entities by few Indian researchers at Anna University. The proposed solution is also the same which uses Sinhala named entities as semantic categories. But like Tamil named entity categorization study, we don't have accurate POS taggers or well defined semantic constraints to use in feature extraction. Because of that, word level features like n-grams, suffixes and context word features are used as features.

Bootstrapping module uses two seed lists and pre-processed pattern corpus to give the output of person and location category words. Evaluating this module alone can't get an idea about how successful the word extraction process is. But it will give an idea about the accuracy of extracted words. It shows 93% precision for person names and 89% precision for location names. So overall precision is about 91%. That is actually a good intermediate result if we consider only about the extracted words. So we have to identify what is the real purpose of this process and what are we going to with the results of this. There are two main things that we can use the results of bootstrapping. First one is to increase the number of words that belong to relevant semantic word categories. The second one is to build a corpus by collecting sentences which contain intended named entity categories. Considering both aspects, we can understand that knowing only the precision of the result is not a big issue here. Because there is no big concern of extracting all the named entity words from the corpus.

The machine learning models were trained under the assumption that the results of the bootstrapping module are 100% accurate and tested under the assumption all human labeled data is 100% accurate. So before analyzing the results of ML models, we have to understand these two errors are always there. In ML model 1, words are separated into the name and non-name categories. Normally in a text, non-name entities are higher than named entities. Even though positive and negative training sets are equally given, the model has trained more bias to the non-name category. The high percentage of precision, recall, and f1- score values of the non-name category is explained that. The precision of named entity category is 83% and that is sufficient for the solution. But the recall of named entity category is 13% and it is very low percentage compare to other results. There can be a few possible reasons behind that result. Three main reasons that I observe are given below.

First one is, most of the time Sinhala person names existed with two names and when system identify named entities by looking their pre-word and post-word (not suffix) features, the ones that identified by post word "මහතා" are only final names. As an example consider the sentence segment "සරත් අමුණුගම මහතා". Name "අමුණුගම" will be identified by the system because post-word "මහතා" is located after "අමුණුගම" and named entity "සරත්" will be ignored.

The second one is, there are some places that can't identify by any word features that we used. As an example, consider sentence segment "උපුල් තරංගගේ ක්‍රීඩා විලාසය". There are two named entity words in this sentence segment and none of them will be identified by the system. Solution to this is using POS feature or morphological parser to consider language grammar.

Since words are not stemmed or lemmatized, Sinhala word ending suffixes are still there. Because of that, suffix word feature will not always catch all the named entities. That is the third reason. Ex :- "මහරගමදී", "ඉන්දියාවේ", "මාතරින්".

Other than that some foreign names might not be extracted by the system due to their uncommonness compare with Sinhala names.

The machine learning model 2 shows more than 95% precision, recall, and f1-score values for both person and location categories. Reason for that much percentage is because ML model 2 get only patterns that predicted as names by ML model 1. So this method can be very useful to categorize person and location names from named entity list that have only person names and location names.

Finally, we can argue that the complete system was performed better than we expect. One of the initial goals of this research was to increase categorized data with very less effort compare to fully supervised or statistical learning methods. Another one was to test the bootstrapping method with Sinhala language corpus and identify and analyze the problems that will be encountered. Accordingly, both goals were achieved successfully.

6.2 Implications for further research

One of the main limitations of this research is names that can be recognized by considering only itself are categorized as a person or location names. So this limitation can be eliminated in the future by using BIO encoding. As mentioned in chapter1 other than person and location names there are more named entity categories like organization, time expressions, numeric expressions. So future studies can also include these entity categories. Scoring algorithms that used to score pre-words, post-words and suffixes can be modified to getting better results. Finally, if the accuracy of stemming, lemmatization, morphological parsing or POS tagging for the Sinhala language will come to an acceptable level, using those features to build pattern will definitely increase the accuracy of results.

References

[1] R.Grishman and B.Sundheim, "Message understanding conference-6: a brief history," 16th conference on Computational linguistics, vol. 1, p. 466–471, 1996.

[2] Chinchor and N. A., "Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition," p. 21 pages, April 1998. [Online]. Available: http://acl.ldc.upenn.edu/muc7/ne_task.html

[3] D. S. Batista., "Semi-supervised bootstrapping relationship extractors with distributional semantics," July 2017. [Online]. Available: http://www.davidsbatista.net/assets/documents/talks/PyData2017-Berlin-presentation.pdf

[4] Z. Kozareva and E. Hovy, "Not all seeds are equal: Measuring the quality of text mining seeds," p, pp. 618–626, 2010.

[5] S. Brin, "Extracting patterns and relations from the world wide web," Lecture Notes in Computer Science, pp, pp. 172–183, 1999.

[6] E. Agichtein and L. Gravano, "Snowball," in Proceedings of the fifth ACM conference on Digital libraries - DL '00, 2000.

[7] E. Riloff and R. Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," 1999.

[8] W.Lin, R.Yangarber, and R.Grishman, "Bootstrapped learning of semantic classes from positive and negative examples," The Continuum from Labeled to Unlabeled Data, vol. 4, no. 4, 2003.

[9] E.Riloff, J.Wiebe, and T.Wilson, "Learning subjective nouns using extraction pattern bootstrapping," Natural language learning at HLTNAACL, vol. 4, pp. 25–32, 2003.

[10] E.Riloff and R.Jones, "Learning dictionaries for information extraction by multi-level bootstrapping," AAAI/IAAI, pp. 474–479, 1999.

[11] D. Waegel, "A survey of bootstrapping techniques in natural language processing," 2009.

[12] M. Thelen and E. Riloff, “A bootstrapping method for learning semantic lexicons using extraction pattern contexts,” in Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP ’02, 2002.

[13] T.S, B.J, and G.T.V, “Semi-supervised bootstrapping approach for named entity recognition,” International Journal on Natural Language Computing, vol. 4, no. 5, pp. 1–14, 2015.

[14] R. Weerasinghe, D. Herath, and V. Welgama, “Corpus-based sinhala lexicon.”

[15] D. Upeksha, C. Wijayarathna, M. Siriwardena, L. Lasandun, C. Wimalasuriya, N. D. Silva, and G. Dias, “Implementing a corpus for sinhala language,” 2015.

[16] J. Dahanayaka and A. Weerasinghe, “Named entity recognition for sinhala language,” in 2014 14th International Conference on Advances in ICT for Emerging Regions (ICTer), 2014.

[17] K. Senevirathne, N. Attanayake, A. Dhananjanie, W. Weragoda, A. Nugaliyadde, and S. Thelijjagoda, “Conditional random fields based named entity recognition for sinhala,” in 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS, 2015.

[18] S. Manamini, A. Ahamed, R. Rajapakshe, G. Reemal, S. Jayasena, G. Dias, and S. Ranathunga, “Ananya - a named-entity-recognition (ner) system for sinhala language,” vol. 2016, 2016.

[19] S. S. David Nadeau, “A survey of named entity recognition and classification,” 2007. [Online]. Available: https://nlp.cs.nyu.edu/sekine/papers/li07.pdf

[20] M. Asahara and Y. Matsumoto, “Japanese named entity extraction with redundant morphological analysis,” in Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - NAACL ’03, 2003.

[21] E. Bick, “A named entity recognizer for danish,” 2004.

[22] S. Boutsis, I. Demiros, V. Giouli, M. Liakata, H. Papageorgiou, and S. Piperidis, “A system for recognition of named entities in greek,” Lecture Notes in Computer Science, pp, pp. 424–435, 2000.

[23] A. Cucchiarelli and P. Velardi, “Unsupervised named entity recognition using syntactic and semantic contextual evidence,” Computational Linguistics, vol. 27, no. 1, pp. 123–131, 2001.

[24] T. Poibeau, “The multilingual named entity recognition framework,” in Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - EACL ’03, 2003.

[25] D.Salazar, J.Vélez, and J.Salazar, “Comparison between svm and logistic regression: Which one is better to discriminate?” Revista Colombiana de Estadística, vol. 35, no. 2, pp. 223–237, 2012.