# Automated Title Generation in Sinhala Language

T. H. Batawalaarachchi

# Automated Title Generation in Sinhala Language

**T. H. Batawalaarachchi**
**Index No. : 14000105**

**Supervisor: Dr. M.I.E. Wickramasinghe**
**Co-Supervisor: Mr. W.V. Welgama**

**December 2018**

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

# Declaration

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

Candidate: T. H. Batawalaarachchi

Signature of Candidate

May 30, 2019

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

Supervisor: Dr. M.I.E. Wickramasinghe

Signature of Supervisor

May 30, 2019

Supervisor: Mr. W.V. Welgama

Signature of Supervisor

May 30, 2019

# Abstract

With the recent advancements in the area of information technology, information has become available in large quantities. Though the availability has increased, the value of correct and meaningful information has not gone down and considered as one of most valuable resources. But, due to the high number of resources and the increase in content, time taken for accessing the required information has also increased. Hence, it is more important to access the desired information within the required time. In this case, the value of the concepts of summary and more importantly title reaches a great value. When considering a document, a title presents a compact representation of what is included in the document. Hence title is important in capturing the main idea of a document quickly, without spending time on reading the whole article. Then a reader can make the decision whether the document is useful for the purpose he/she intended to use it.

In this work, two approaches in selecting words to be included in the title for a given Sinhala document is discussed. Both the approaches use statistical features from a selected corpus to include words in the title. First approach considers the words included in the titles and the structure of the corresponding document, while the second approach focuses on translation from words in the document to the words contained in the title.

Two approaches were evaluated using human evaluation and automatic (averaged F1 score) evaluation. Though similar approaches have obtained acceptable results with the work done on other languages, by the results of this work, it is clear that statistical approaches are not the go to method for the title word selection task in Sinhala Language. This can be mainly due to the high complexity of the language organization and also the structural distance of Sinhala Language from the languages which these approaches have shown better results.

# Preface

Two approaches are described in this work to provide a solution to the problem of title generation for Sinhala language texts. First approach is based on a statistical method for title word selection while the second employs statistical translation. Both the approaches have been proposed earlier for similar work done on other languages (Hindi and Telugu).

However, these approaches have not been attempted on any work in the domain of title generation for Sinhala language texts. The evaluation model introduced in this dissertation contains automatic and manual evaluation methods. While the automatic measures have been already employed in the domain, the manual evaluation process was proposed by myself with the input of research supervisors to measure the relevance of the suggested title words from each model.

# Acknowledgement

I would like to express my sincere gratitude to my research supervisor, Dr. M.I.E. Wickramasinghe, lecturer of University of Colombo School of Computing and my research co-supervisor, Mr. W.V. Welgama, lecturer of University of Colombo School of Computing for providing me continuous guidance and supervision throughout the research.

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

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

# Contents

# List of Figures

# List of Acronyms

| | |
|---|---|
| EM | Expectation Maximization |
| NLP | Natural Language Processing |
| IBM | International Business Machines |
| TF | Term Frequency |

# Chapter 1

# Introduction

## 1.1 Background to the Research

With the improvement of the technology, methods of generating and making available of information has been increasing exponentially. On the other hand, high value of correct information has reached its peak. With the busy schedules of modern day people, it has become difficult to go through every bit of information available to someone to check whether it is really relevant for a particular person or not. But with the rapid growth of world wide web, most of the information made available has become unstructured, and hence it has become a problem of accessing correct information at the correct time. Therefore, in identifying the relevance of available information for a person without spending much time, for a body of information concepts of summary and title has gained a great importance.

A summary of a text provides most important contents of the text in a condensed manner providing the general idea of what the text is about. In general, many forms of summaries can be found in day to day usage such as abstracts, titles, headlines ,table of contents, outlines, minutes, previews, synopses, reviews, digests, biographies, abridgments, bulletins, sound bites, histories [1]. As mentioned above, generation of titles comes under the task of summarization, providing a more shorter length sentence or a phrase denoting the main theme of the text.

With the introduction of the Unicode standard for Sinhala language, Sinhala related e-content generation, storing and publishing has been increased which now has even led to the problems concerning quality and the quantity of the content. There are instances where titles assigned to text are misleading in a way that the title is not indicating the exact idea conveyed in the text. This can be done on

purpose for marketing gains of the writer or can be due to the writer's inability to assign a meaningful title. Either way this removes the basic value of a title. If there is a predefined representation to a title, this ambiguity can be overcome by providing a well formed word group related to the text content as a title.

The motivation towards this research has been driven with the idea of introducing an acceptable method for selecting words for a representation which can overcome above mentioned ambiguity issues and providing a model which assigns a titles for documents in Sinhala language using that representation.

## 1.2 Research Problem and Research Questions

With the flooding of Sinhala texts with large number of resources, Sinhala readers in the cyber world has increased. But there is no clear accepted representation of a title for documents written in Sinhala language, resulting poor representation of document content and misleading readers. Hence there is a need for a method of generating titles for Sinhala documents in Sinhala language automatically, a problem which is yet to attempt to solve.

Considering this research problem, the generated research question is as follows:

- What is the most suitable method to select title words from a text written in Sinhala Language?

## 1.3 Justification for the research

Having a well-defined model to assign titles holds a great significance in many practical scenarios. One of the main usages is that the ability to use such model to title articles or documents in order to avoid readers from being misled based on the writer assigned title. This can be achieved by incorporating such a model and providing the generated title with the writer assigned title. Also, as already mentioned, since title generation task can be viewed as a highly condensed sum-

marization problem, findings of this research can be applied in the summarization tasks up to a certain extent.

In text containing large number of paragraphs or sections such as legal documents (acts, case documents, etc.), there is no easy way to access a required part of the document easily. If there is a way of assigning titles for those sections, it can be used as an indexing mechanism. For devices with limited display or bandwidth, summarizing web pages or email content can be done with the use of a proper title generator.

Though this research concerns on objective aspects, after defining a proper framework, using the outcomes of the research, based on a social study, it can be incorporated to develop a model to generate subjective titles based on a requirement in future. For example, generating politically biased titles for news articles can be attempted.

Furthermore, usage of automatic title generation is not limited to written articles. It can be used to create titles for machine-generated texts, such as speech recognition transcripts and machine translated documents. Providing compressed descriptions for search results in search engines and augmenting those descriptions for with a user search query to re-rank and improve the search results can be performed with the automatically generated titles [1]. The later application of improving search results is used in contextual text search [2].

Although there has been attempts on automatic text summarization for Sinhala Language, no recorded attempts are to be found for the task of title generation for Sinhala Language texts.

## 1.4 Methodology

Due to the unavailability of a data set suitable to be used in the approaches selected, first a sufficient amount of articles is collected. Along with the already available evaluation methods used in the literature, a ranking system is suggested to be used in the manual evaluation as the next step of the proposed approach. Then the

preprocessing of the obtained data. Natural Language Processing tasks required by each approach is performed. Figure 1.1 depicts the overview of the methodology.

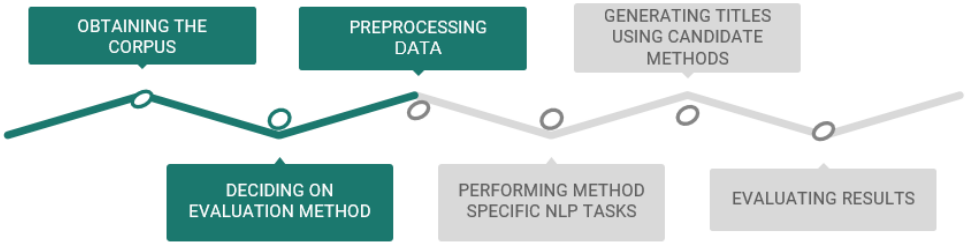


Figure 1.1: Methodology

Two main approaches considered are described in the upcoming chapters in detail.

## 1.5 Outline of the Dissertation

The dissertation is structured as follows. Chapter two explores the existing approaches related to the domain of summarization and title generation for English language and similar Indic languages. Chapter three describes the proposed research design and methodology. Potential ways of addressing the research problem is discussed in this chapter. Chapter four demonstrates the implementation details of the proposed methodologies. Chapter five presents the evaluation model and the evaluation results of the proposed approaches. The last chapter, chapter six demonstrates the conclusion of the thesis and outlines the future work.

## 1.6 Delimitations of Scope

This research will be done to find a suitable model for selecting words to be included in the title in a document written in Sinhala language. Due to unavailability of suitable dataset and requirement of pre assigned titles for the articles, dataset used in the research will be a collection of feature articles collected from a website of Sinhala science magazine assuming the documents are written and titles are

assigned by people who carry acceptable language usage of Sinhala.

Since linguistic tools and resources employed in the work performed on English and similar languages are yet to achieve acceptable quality in Sinhala language, statistical approaches will be considered in this work which is more suitable for less resourced languages.

# Chapter 2

# Literature Review

Title Generation for Sinhala Language is not a previously focused topic. Therefore background of the research is mainly limited to the work done on Title Generation for English and other Indic languages. Requirement of natural language understanding techniques and natural language synthesis has distinguished automatic title generation from other similar tasks such as key phrase extraction ([3] ,[4], [5]), text summarization ([6]), information retrieval ([7]) and information extraction ([8]), where the only task is to identify the important content of documents. Yet the task title generation can be considered as an extremely shortened summarization to some extent. Hence, when referring to the earlier work done on title generation, it can be thought as a summarization problem as well.

In the literature, types of titles are mainly categorized into three groups.

- Indicative – indicates what topics are covered in the text
- Informative – convey the particular concept, event or theme covered in the text
- Eye-catchers – designed to grab the attention and tempt a person to read the text

Most of the related work in the literature on title generation can be categorized to be either an Extractive, a Rule-based or a Statistical approach [1]. From the above three, based on the selected approach, type of the generated title may differ.

## 2.1 Extractive Approaches

By identifying titles as summaries of a very short length, automatic text summarization methods can be used for the task of automated title generation. Three groups can be identified among the extractive approaches namely surface-level, entity-level and combination of the two groups[1].

Surface-level approaches find important sentences which are suitable to be included in the summary using surface-level features. Baxendale [9] pointed out sentence position as a feature to find the most important parts of documents. For this, he used 200 paragraphs and presented the results as the topic sentence came as the first sentence in 85% and as the last sentence in 7%. But since this approach is not using the entire document to provide a summary, this was considered not suitable for summarization tasks. Some other features used for this purpose were word frequency, cue phrases, and number of key words present in a sentence. Using 4 features namely, word frequency, positional importance (incorporated from earlier work), presence of cue words and skeleton of the document individually, Edmundson [10] proposed a system to generate document extracts.

Later researches have focused on combining these features with the use of machine learning algorithms. A method which was a derivation of Edmundson [10] was suggested by Kupiec et al [11]. Using a naïve Bayes classifier, the classification function categorized sentences if they are worthy of extracting or not. In addition to the features used in [10], sentence length and the use of uppercase letters were also included.

Along with the naïve Bayes classifier, Lin [12] considered the dependency of the features and modeled the problem of sentence extraction using decision trees. Here he examined a lot of features and their effect on sentence extraction. Some of the novel features he used were presence of numerical data, proper name, pronoun, adjective, weekday or month, quotation. In the evaluation, it was seen that out of the whole data set, decision tree method outperforms the using of naïve Bayes classifier. While above methods were mostly feature based and non- sequential,

Conroy and O'leary [13] modeled extracting a sentence from a document using Hidden Markov Model. Usage of this sequential model was chosen taking the local dependencies between sentences in to account. Additionally, they used a joint distribution for the features set, unlike the independence-of-features assumption used by naïve Bayesian methods. In the experiment, position of the sentence in the document (built into the state structure of the Hidden Markov Model), number of terms in the sentence and likeliness of the sentence terms given the document terms were used as features. Osborne [14] used log linear model to show the assumption of most of the previous approaches to have feature independence is not accurate on every instance. In this work, he managed to produce results in favor of his assumption.

Entity-level approaches for the task are syntactic analysis, discourse analysis and semantic analysis. These methods depend on linguistic analysis of text in obtaining linguistic structures such as discourse structure, syntactic structure and rhetorical structure to create a summary.

Some work has been done combining the surface-level approaches and entity-level approaches as well. Aone et al [15] used a naïve Bayes classifier. Their system, DimSum used features such as term frequency and inverse document frequency to derive signature words. In their work, an entity tagger was used and discourse analysis was done to reference the same entities in the text.

Main advantage of using extractive approaches is that there is no need to consider the title generation as a special problem, and can use an existing summarization technique to generate a highly compressed summary which can be used as a title. But the main issue of using these techniques for title generation is when the compression rate is well below 10%, the quality of the generated summaries is poor. Since most of the articles contain titles between 10-15 words, the required compression rate becomes lower than 10% resulting poor quality titles. Another problem with this approach is that that the smallest unit of summarization may become longer than the required length of the title. Another drawback of this approach is that the generated title is limited to the phrases and words present in

the original text.

## 2.2 Statistical Approaches

This approach depends on the availability of a large training corpus and works in a supervised learning setting. The model is trained to learn the correlation between articles and the corresponding titles. Then, the trained model is applied to generate the titles of novel documents. In general, compared to the other approaches, this method is considered more robust to the noise in the articles given that a large training corpus is present. Hence, this approach is suitable for machine generated texts. This approach can be used to produce titles containing words and phrases which are not included in the article as well.

A Naïve Bayesian approach for learning the document word and title word correlation was suggested by Witbrock and Mittal[16]. Their research was limited to a special case of the document word and the title word are same surface string.

Restriction used in [8] to use a special case of the document word and the title word are same surface string was relaxed by Jin and Hauptmann [17]. They used K Nearest Neighbor Method for generating titles.

To address the problem as a variant of Machine Translation Problem, Kennedy and Hauptmann [18] have used iterative Expectation-Maximization Algorithm. A large corpus of documents with human-assigned titles were required for training title "translation" models.

Relying in the availability of large set of training data itself becomes a major disadvantage of the statistical approaches. Furthermore, since statistical methods compute the correlation between every article word and every title word throughout the training data, it is more computationally expensive than other methods.

## 2.3 Rule Based Approaches

Rule based approaches are somewhat similar to extractive approaches in using linguistically motivated heuristics to create a title that guide the choice of a potential title. Hedge Trimmer [19] is an example of this category which uses a parse and trim scheme. The approach in Hedge Trimmer is very similar to the sentence compression work of Knight and Marcu [20], where a single sentence is shortened using statistical compression.

The system creates a headline for a news article by removing constituents from the parse tree of the lead sentence of the article until a certain length threshold is reached. Linguistically motivated techniques guide the choice of what constituents should be removed and retained.

Main advantage of Rule based techniques is that there is no requirement of prior training on a large corpus of headline-story pairs since there is no model to be learnt. But deciding which single sentence best reflects the contents of the entire news article is a difficult task. Often, news stories have important pieces of information scattered throughout the article and the approach of trimming the lead or a single important sentence can be unsuccessful in practice.

## 2.4 Attempts on Non-English Languages

Since Sinhala is a language very distant from English language, in this section, some of the previous works performed on Indic Languages, which are more closely related to Sinhala Language than English Language is discussed in this section.

In the attempts on automatic summarization of Tamil language texts, an extractive approach using a sub graph was employed in [21] with the use of a Language Neutral Syntax and logical form of semantics in each document. A Suppport Vector Machine was used as the training mechanism. [22] has used a graph theoretic scoring technique for ranking sentences based on word frequency, position of word in a sentence and a sentence weight using string pattern. The system has achieved

a Rouge score of 0.4723. Similar sentence weighting mechanism was employed in [23], which was done for Hindi Language achieving 85% average accuracy.

In an attempt to generate gist for Hindi news articles [24], authors have achieved 0.602 F1 score using a combination of 3 models. Employing statistical translation for Telugu [25], which was originally introduced for English language in [18], authors have achieved 3.52 score from human evaluation on 150 articles, out of 5 (5 - very good, 1 - extremely bad). The above two approaches are explained in detail in next two chapters.

Among the above approaches, statistical approaches can be identified as the most suitable for generating indicative titles since they scan whole article to select title words. For generating informative headlines, Rule based linguistic techniques like hedge trimmer is more suitable since they occupy a method of trimming the most important sentence to an acceptable length for a title. Extractive approaches tend to provide either an informative title or an indicative one.

# Chapter 3

# Design

This chapter explicates the proposed solutions to the research problem. It consists of four major sections namely introduction, data set, preprocessing and Title word selection. Introduction section provides an idea on why the selected approaches are chosen. Data set section explains why and how the unit of analysis was chosen. Preprocessing section describes what operations were done on data before it was analyzed. Under title word selection, each approach used in the work is described in detail.

## 3.1 Introduction

As described in the previous chapter, title generation task has been attempted in three major approaches. Among them, rule based approaches require accurate linguistic resources if they are to be employed. On the other hand, while requiring linguistically motivated techniques, extractive approaches suffer from the mismatch between the length of the unit of extraction and the length of the title. Due to Sinhala being a low resourced language, models selected in this work is limited to statistical context.

## 3.2 Data set

Similar to the general statistical approaches, selected models for the work requires a corpus of title-document pairs. It also requires the documents in the data set to be of similar context. Therefore, feature articles of the weekly magazine "Vidusara" was chosen to be used as the data set for this work.

Scrapy, a python web-crawling framework is used to extract data from the magazine's web site. A pattern is observed in the URL of web page for each article. Extracted articles are then re-analyzed before including into the final data-set.

## 3.3 Preprocessing

During the preprocessing stage each article is stripped off of unnecessary content such as writer's name and details. Then functional words are removed. Resulting articles are then used to create a word list in order to perform stemming on the corpus. This process is explained further in next chapter.

## 3.4 Title World Selection

Approaches chosen for selecting words to be included in the title are previously employed for Hindi c, Telugu [25] and English [18]

### 3.4.1 Statistical Approach

This approach is adopted from [24] which was done for gist generation for news articles written on Hindi Language. The main concern here in [24] was to select informative words based on three models. This was adopted to select words which are important to be included in the title in this work. Three models used in the work are,

1. Sentence Position Model
2. Informative Word Position Model
3. Text Model

Sentence position model captures the sentence position information. Informative word position model identifies the information regarding the likeliness of the first

appearance of a document word while the text model basically captures the correlation between title words and document words. A detailed explanation of the above three models is provided in the next chapter.

### 3.4.2 Statistical Translation Approach

This approach is adopted from [18] which was based on IBM machine translation approach. In IBM Machine Translation Approach, given a source language string **s**, finding the target language text string **t** most likely to represent the translation that produced **s**, i.e., find

$$Argmax_t p(t|s) = Argmax_t p(s|t) \cdot p(t)[Bayes]$$

This definition is used in the title word selection context as for a given document, finding the title most likely to represent the translation that produced the document.

$$Argmax_{title} p(title|document) = Argmax_{title} p(document|title) \cdot p(title)$$

In machine translation tasks [26] has used an English language model to estimate the prior probabilities p(t). Similarly, to estimate p(title), [18] a standard trigram language model to define a space of possible titles and their prior probabilities is used. In this work, a unigram model of title words is used for that purpose. Estimating the $\text{Argmax}_{\text{title}}$(document|title) is explained in the implementation chapter.

# Chapter 4

# Implementation

## 4.1 Preprocessing

After obtaining the documents and cleaning the documents, Functional Words were removed from the documents in the data-set. Since both the approaches were based on statistical methods, it is required to perform stemming on the articles. To this end, to identify stems a special procedure was used.

All Sinhala words are collected to prepare a word list as the first step. This list is then sorted in the alphabetical order. Based on a pre-identified Sinhala suffix list, starting from the first word, following actions are performed in the sorted order. Steps followed are given in the Algorithm 1. First word of the list is taken as the current stem.

Figure 4.1 shows some sample stems generated using the above method.

| Word | Stem | | Word | Stem | | Word | Stem |
|---|---|---|---|---|---|---|---|
| අංකන | අංකන | | අංශික | අංශික | | අකැමැත්ත | අකැමැත්ත |
| අංකනය | අංකන | | අංශු | අංශු | | අකැමැත්තකින් | අකැමැත්ත |
| අංකය | අංකය | | අංශුක | අංශු | | අකැමැත්තයි | අකැමැත්ත |
| අංකයකට | අංකය | | අංශුන් | අංශු | | අකැමැත්තෙන් | අකැමැත්ත |
| අංකයකි | අංකය | | අංශුන්වල | අංශුන්වල | | අකැමැති | අකැමැති |

Figure 4.1: : Sample of results of the stemming process

Based on the generated stems, the corpus is stemmed.

```
input  : Alphabetically sorted word list
         Suffix list
output: Words with their corresponding stems
cur_word ← First word in word list;
cur_stem ← curword;
while world list has more words do
    cur_word ← next word from the word list;
    if cur_word starts with cur_stem then
        if cur_word ends with a suffix in the suffix list then
            stem[cur_word] ← cur_stem;
        else
            stem[cur_word] ← curword;
            cur_stem ← curword;
        end
    else
        stem[cur_word] ← curword;
        cur_stem ← curword;
    end
end
```

**Algorithm 1:** Algorithm for assigning stems to words

## 4.2 Title Word Selection

### 4.2.1 Statistical Approach

As described in the previous chapter, statistical approach consists of three models to assign a score to each article word in order to select title words.

#### Sentence Position Model

From the early approaches of summarization, sentence position is considered as a key feature when selecting words or phrases which represents the salient theme of a text [9]. This idea is taken into consideration here as a step in selecting suitable title words.

$$CountPos_i = \sum_{k=1}^{M} \sum_{j=1}^{N} P(G_k/W_j)$$

For each sentence position **i** over all **M** texts in the collection and over all the words in the **M** titles (each containing up to **N** words), **Countpos** records the number of times where sentence position **i** has the first appearance of any informative word. $\mathbf{P(G_k|W_j)}$ is a binary feature. This value is calculated for all sentence positions from 1 to **P**.

$$P(G|Pos_i) = \frac{CountPos_i}{\sum_{p=1}^{P} CountPos_p}$$

Resulting $\mathbf{P(G|Pos_i)}$ represents each sentence position containing one or more title words.

#### Informative Word Position Model

This model is intended to identify information related to each document word.

$$P(Pos_i|W_g) = \frac{Count(Pos_i, W_g)}{\sum_{j=1}^{P} Count(Pos_j, W_g)}$$

For each document word $W_g$ likeliness of the word $W_g$ appearing at position **i** is calculated using this model.

**Text Model**

Text model is designed to capture the correlation between words in text and words in key phrases.

$$P(G_w|T_w) = \frac{\sum_{j=1}^{M}(docTF(w,j) \cdot titleTF(w,j))}{\sum_{j=1}^{M} docTF(w,j)}$$

Here,

- docTF(w, j) is the TF of word **w** in the $j^{th}$ document among all documents in the train set.
- titleTF(w,j) is the TF of word **w** in the $j^{th}$ title.
- $G_w$ and $T_w$ are words that appear in both the theme and body of the text. For each instance of $G_w$ and $T_w$ pair, $G_w$=$T_w$.

Combined score of the three models is calculated as the product of the values obtained in three models.

$$P(G|W_i) = P(G|Pos_i) \cdot P(Pos_i|W_g) \cdot P(Gw_i|Tw_i)$$

### 4.2.2 Statistical Translation Approach

As explained in the previous chapter IBM translation model is adopted for the task of title word selection as following equation.

$$Argmax_{title}p(title|document) = Argmax_{title}p(document|title) \cdot p(title)$$

In the original statistical translation approach of IBM, they have introduced several statistical models of alignments based on how the target language sentence might translate into corresponding words or phrases in a source language sentence. Among those models, the simplest model(model 1) is used in this work for estimating **p(document|title)**. This model treats the title and document as a "bag of words". For a given pair of words, one from the title vocabulary and one from the document vocabulary, this model simply estimates the probability that the

document word appears in a document given that the title word appears in the corresponding title. Thus the model consists of a list of document-word/title-word pairs, with a probability assigned to each. For a pair to be in the list there must have been an actual document/title pair in the training corpus where the title word occurs in the title and the document word occurs in the document. The probabilities are estimated in multiple iterations using the EM algorithm [18]. Each title word in a document maps to one or more words in the document for a given alignment. Each document word maps to 0 or 1 title words. If the document word maps to 0 title words, then it is considered to be mapped to a "null" title word. All possible combinations of mappings between words of document and title are allowed and considered to be equally probable. Lengths of the title and document are considered to be independent. The model is estimating a fixed **translation probabilities** for each title-word, document-word pair. The EM algorithm converges to a global maximum in the model. Full implementation of the IBM model is in Appendix A: under code listing A.2.

# Chapter 5

# Results and Evaluation

## 5.1 Evaluation Criteria

### 5.1.1 Automatic Evaluation

For evaluating the selected title words for each approach, Precision and Recall values were calculated against the original titles of the articles. Experimental results are evaluated using the F1 score calculated based on Precision and Recall. These metrics are well established and widely used in the field of information retrieval.

In this work, precision denotes the fraction of the title words that are appearing in both human assigned original titles and machine selected title words from the machine selected title words. Recall is the fraction of the title words that are appearing in both human assigned original titles and machine selected title words from the words in the human assigned original title. Below equations show the mathematical definitions used.

$$Precision = \frac{|\{W_{human}\} \cap \{W_{machine}\}|}{|\{W_{machine}\}|}$$

$$Recall = \frac{|\{W_{human}\} \cap \{W_{machine}\}|}{|\{W_{human}\}|}$$

Among the variations of F measure, simplest value F1 score is used here which is the harmonic mean of precision and recall, for evaluations, calculated using below equation.

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

### 5.1.2 Manual Evaluation

Automatic methods cannot be used as the only measure for machine generated title evaluation due to the difficulty of judging readability and quality by a computer program because it lacks linguistic skills that a human posses. Human evaluation does possess drawbacks as,

1. Process of evaluation is tedious
2. Lack of consistency: two human evaluators may not agree with each others' judgements

Due to above weaknesses, using automatic methods has been popular in related work because of the quick evaluation process and the consistency due to adhering to fixed logic in evaluating process.

In Manual evaluation, 8 evaluators were selected. Each evaluator was provided 10 articles along with the selected title words from two approaches and were asked to assign a score for each title word set based on the relevance to the article text. Score scale is ranging from 1 to 5 where 5 denotes highly relevant and 1 denotes not relevant at all.

## 5.2 Results

Since the average length of the title in the data-set is 10 words, from each approach, 10 words with the highest score were selected as the set of title words for a given document. Averaged F1 scores obtained for the statistical translation approach is 0.190. For the statistical approach it is 0.152. In the manual evaluation, statistical model has obtained a score of 1.75 while Statistical Translation approach has obtained a score of 2.125.

# Chapter 6

# Conclusions

## 6.1 Introduction

In this dissertation two approaches for selecting words to be included in the title for a given document written in Sinhala Language were tried out and results were obtained. Based on the results, conclusions are presented in this chapter.

## 6.2 Conclusions about research question

Among the two approaches discussed, while statistical translation approach has obtained higher scores in both automatic(0.190 against 0.152) and manual(2.125 against 1.75) evaluations, it cannot be said that it is the best approach to be used for selecting title words for a given Sinhala language document. This is mainly due to the less distinction between the two values obtained in each approach.

## 6.3 Conclusions about research problem

To provide a robust representation of a title for a text, it is important to identify salient words in that text. Sinhala Language is structurally distant from languages like English. Therefore it is understandable that these approaches give higher scores for English and other languages even without using any linguistically motivated techniques. Hence it should be noted that using pure statistical models for selection of title words does not provide acceptable results in the task of title generation.

## 6.4 Limitations

As explained in the previous chapters, data-set used in this work is a collection of feature articles. These articles are written by many distinct writers. Furthermore these articles are not limited to a certain area or field. These factors mainly affect the structure of the document, which is the most important aspect considered in statistical models. Furthermore, some titles in the data-set belonged to the category of eye catchers, which clearly deviates the original assumptions used in the statistical models of title providing a meaning representation of the document text.

## 6.5 Implications for further research

This research was carried out based on the statistical approaches of word selection used in automatic title generation . The main objective on selecting such approaches is to carry out the research with minimum available linguistic resources for Sinhala language. With the availability of a more robust data-set with a articles having a specific structure, these approaches may provide better results. Therefore future researchers are encouraged to employ these approaches based on obtainability of such a data-set.

Future research on automatic title generation can be enhanced with the development of Sinhala linguistic resources. Linguistically motivated rule based techniques along with the statistical features in a corpus can be used to train the features and that lead to give better performance for a title generator. Other linguistic resources such as sentence parsers, taggers, named entity recognizers and WordNet will be greatly helpful to identify the information in a sentence and then to extract them and represent in a machine understandable format. Finally, such resources can be used to regenerate Sinhala language texts which are essential to present the generated titles in a human readable, meaningful form. The ability of generating Sinhala language text is vital to create titles which are more closed to human

assigned titles.

# References


[1] A. K. Gattani, *Automated natural language headline generation using discriminative machine learning models*. PhD thesis, School of Computing Science-Simon Fraser University, 2007.

[2] R. Kraft, F. Maghoul, and C. C. Chang, "Y! q: contextual search at the point of inspiration," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, pp. 816–823, ACM, 2005.

[3] P. D. Turney, "Learning algorithms for keyphrase extraction," *Information retrieval*, vol. 2, no. 4, pp. 303–336, 2000.

[4] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning, "Domain-specific keyphrase extraction," in *16th International joint conference on artificial intelligence (IJCAI 99)*, vol. 2, pp. 668–673, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

[5] C.-H. Leung and W.-K. Kan, "A statistical learning approach to automatic indexing of controlled index terms," *Journal of the American Society for Information Science*, vol. 48, no. 1, pp. 55–66, 1997.

[6] I. Mani, *Advances in automatic text summarization*. MIT press, 1999.

[7] K. S. Jones, *Readings in information retrieval*. Morgan Kaufmann, 1997.

[8] *MUC6 '95: Proceedings of the 6th Conference on Message Understanding*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 1995.

[9] P. B. Baxendale, "Machine-made index for technical literature—an experiment," *IBM Journal of Research and Development*, vol. 2, no. 4, pp. 354–361, 1958.

[10] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.

[11] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 68–73, ACM, 1995.

[12] C.-Y. Lin, "Training a selection function for extraction," in *Proceedings of the eighth international conference on Information and knowledge management*, pp. 55–62, ACM, 1999.

[13] J. M. Conroy and D. P. O'leary, "Text summarization via hidden markov models," in *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 406–407, ACM, 2001.

[14] M. Osborne, "Using maximum entropy for sentence extraction," in *Proceedings of the ACL-02 Workshop on Automatic Summarization-Volume 4*, pp. 1–8, Association for Computational Linguistics, 2002.

[15] C. Aone, M. E. Okurowski, and J. Gorlinsky, "Trainable, scalable summarization using robust nlp and machine learning," in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pp. 62–66, Association for Computational Linguistics, 1998.

[16] M. J. Witbrock and V. O. Mittal, "Ultra-summarization (poster abstract): a statistical approach to generating highly condensed non-extractive summaries," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 315–316, ACM, 1999.

[17] R. Jin and A. G. Hauptmann, "Learning to select good title words: An new approach based on reverse information retrieval.," in *ICML*, vol. 1, pp. 242–249, 2001.

[18] P. E. Kennedy and A. G. Hauptmann, "Automatic title generation for em," in *Proceedings of the fifth ACM conference on Digital libraries*, pp. 230–231, ACM, 2000.

[19] B. Dorr, D. Zajic, and R. Schwartz, "Hedge trimmer: A parse-and-trim approach to headline generation," in *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*, pp. 1–8, Association for Computational Linguistics, 2003.

[20] K. Knight and D. Marcu, "Statistics-based summarization-step one: Sentence compression," *AAAI/IAAI*, vol. 2000, pp. 703–710, 2000.

[21] M. Banu, C. Karthika, P. Sudarmani, and T. Geetha, "Tamil document summarization using semantic graph method," in — *iccima*, pp. 128–134, IEEE, 2007.

[22] S. L. Devi *et al.*, "Text extraction for an agglutinative language.," *Language in India*, vol. 11, no. 5, 2011.

[23] V. Gupta and G. S. Lehal, "Automatic text summarization system for punjabi language," *Journal of Emerging Technologies in Web Intelligence*, vol. 5, no. 3, pp. 257–271, 2013.

[24] M. V. Rao, B. V. Vardhan, and P. V. P. Reddy, "A statistical model for gist generation: A case study on hindi news article," *International Journal of Data Mining & Knowledge Management Process*, vol. 3, no. 5, p. 15, 2013.

[25] P. V. Reddyl, B. V. Vardhan, A. Govardhan, and M. Y. Babufifi, "Statistical translation based headline generation for telugu," *International Journal of Computer Science and Network Security*, vol. 11, no. 6, 2011.

[26] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Computational linguistics*, vol. 19, no. 2, pp. 263–311, 1993.

# Appendices

# Appendix A

# Appendix A: Code Listings

Listing A.1: Implementation of Stem Generation Algorithm

```
def sinhala_word_match(strg, search=re.compile(r'[^\u0D80-\u0DFF.]').s
    return not bool(search(strg))

stop_words = []
suffixes = []
with open("Functional.txt", encoding = "utf8") as fun:
    stop_words = fun.read().splitlines()
with open("Suffixes.txt", encoding = "utf8") as suf:
    suffixes = suf.read().splitlines()
tokens = set()

files = glob.glob("../rawcorpus/*.txt")
for fle in files:
    with open(fle, encoding="utf8") as f:
        text = f.read()
        #creating token list
        tokens.update([token for token in word_tokenize(text) if sinha

sorted_tokens = sorted(tokens)
cur_stem = sorted_tokens[0]
stems = {}

for token in sorted_tokens:
```

```
        if token.startswith(cur_stem) and token[len(cur_stem):] in suffixe
                stems[token] = cur_stem
        else:
                stems[token] = token
                cur_stem = token

with open("stem_list.txt", 'w', encoding="utf8") as f:
    [f.write('{0} {1}\n'.format(token, stem)) for token, stem in stems
```

Listing A.2: Implementation of IBM Model

```
def get_corpus(corpusdir):
    files = glob.glob(corpusdir + "*.txt")
    corpus = []
    for fle in files:
        with open(fle, encoding="utf8") as f:
            text = f.read().split("</title>")
            if len(text[0])>7:
                corpus.append({"title": text[0][7:], "document": text[
    if VERBOSE:
        print(corpus, file=sys.stderr)
    return corpus

def get_words(corpus):
    def source_words(lang):
        for pair in corpus:
            for word in pair[lang].split():
                yield word
    return {lang: set(source_words(lang)) for lang in ('title', 'docum



def init_trans_probs(corpus):
    words = get_words(corpus)
    return {
        word_en: {word_fr: 1/len(words['title'])
                  for word_fr in words['document']}
        for word_en in words['title']}



def train_iteration(corpus, words, total_s, prev_trans_probs):
```

```
    trans_probs = deepcopy(prev_trans_probs)

    counts = {word_en: {word_fr: 0 for word_fr in words['document']}
              for word_en in words['title']}

    totals = {word_fr: 0 for word_fr in words['document']}

    for (es, fs) in [(pair['title'].split(), pair['document'].split())
                     for pair in corpus]:
        for e in es:
            total_s[e] = 0

            for f in fs:
                total_s[e] += trans_probs[e][f]

        for e in es:
            for f in fs:
                counts[e][f] += (trans_probs[e][f] /
                                 total_s[e])
                totals[f] += trans_probs[e][f] / total_s[e]

    for f in words['document']:
        for e in words['title']:
            trans_probs[e][f] = counts[e][f] / totals[f]

    return trans_probs


def is_converged(probabilties_prev, probabilties_curr, epsilon):
    delta = distance(probabilties_prev, probabilties_curr)
```

```
    if VERBOSE:
        print(delta, file=sys.stderr)

    return delta < epsilon


def train_model(corpus, epsilon):
    words = get_words(corpus)
    with open("title_words.txt", 'w', encoding="UTF-8") as f:
            for word in words['title']:
                f.write(word + '\n')
    with open("document_words.txt", 'w', encoding="UTF-8") as f:
            for word in words['document']:
                f.write(word + '\n')
    total_s = {word_en: 0 for word_en in words['title']}
    prev_trans_probs = init_trans_probs(corpus)

    converged = False
    iterations = 0
    while not converged:
        trans_probs = train_iteration(
                                        corpus, words, total_s,
                                        prev_trans_probs
                                    )

        converged = is_converged(prev_trans_probs,
                                 trans_probs, epsilon)
        prev_trans_probs = trans_probs
        iterations += 1
    return trans_probs, iterations
```