Cross Language Information Retrieval for Accessing the English Web in Sinhala

By

Mohamed Hunais Mohamed Hisan
15000532

This dissertation is submitted to the University of Colombo School of Computing
in partial fulfilment of the requirements for the
Degree of Bachelor of Science Honours in Computer Science

University of Colombo School of Computing
35, Reid Avenue, Colombo 07,
Sri Lanka
July 2020

Declaration

I, M.H.M. Hisan (15000532) hereby certify that this dissertation entitled “Cross Language Information Retrieval for Accessing the English Web in Sinhala” is entirely my own work and it has never been submitted nor is currently been submitted for any other degree.

…….. ……..

Date Signature of the Student

I, Dr. A.R. Weerasinghe, certify that I supervised this dissertation entitled “Cross Language Information Retrieval for Accessing the English Web in Sinhala” conducted by M.H.M. Hisan in partial fulfilment of the requirements for the degree of Bachelor of Science Honours in Computer Science.

…….. ……..

Date Signature of the Supervisor

I, Dr. B.H.R. Pushpananda, certify that I supervised this dissertation entitled “Cross Language Information Retrieval for Accessing the English Web in Sinhala” conducted by M.H.M. Hisan in partial fulfilment of the requirements for the degree of Bachelor of Science Honours in Computer Science.

…….. ……..

Date Signature of the Co-Supervisor

Abstract

The Internet is a place where people tend to access in search of knowledge. An immense amount of information is available in many different languages and they can be accessed by people irrespective of the location and time. But it has been observed that search engines do not always provide relevant answers when searching using a less popular language including Sinhala which is one of the native languages of Sri Lanka. Although relevant documents are available for the given query, search engines are not able to link the queries to the appropriate documents since the query and documents are in two different languages. This study focuses on performing Cross Language Information Retrieval (CLIR) from Sinhala to English to retrieve relevant web documents. This includes determining whether a proper system can be built which could perform such a task effectively. To the best of my knowledge, there have been no efforts taken to perform CLIR involving Sinhala Language. In addition to the normal procedure of retrieving documents, this study checks whether there is a different order of importance of the documents when they are translated back to the language of the query.

A word embedding based approach was considered to represent words since they have shown to be effective in representing text data. Several translation models were employed to obtain the equivalent English query for a given Sinhala query and the Linear Transformation combined with the Standard Nearest Neighbour Retrieval method has performed well. Among the Re-ranking models used in this study, the LSI based re-ranking model was performed well. But re-ranking the documents did not show a positive impact.

A brief user-based evaluation was performed and the results showed that it is possible to perform Sinhala to English CLIR using a word embedding based approach.

Preface

Translation Models and Re-ranking models have been developed and evaluated to determine the best performing models to build a system that could assist people to retrieve relevant Sinhala documents from the web. The data to train the word embedding model was obtained online and it was preprocessed completely by me. This preprocessed data was fed into the FastText model provided by Gensim to obtain the Sinhala word embedding model. The English word embedding model used was the model provided by FastText. Gensim provides implementations to train a Linear Transformation model with both the Standard Nearest Neighbour and Globally Corrected Neighbour Retrieval methods. The procedure which uses the orthogonal transformation based on Singular Value Decomposition (SVD) was obtained through GitHub. The dictionary required to train both these transformations were created by me with the help of Google Translate to obtain the translations of Sinhala words.

The required documents for a particular query was obtained using the Google Search API. The algorithms to read the content of the retrieved documents and obtain the translated documents were created by me where the translations were obtained using Google Translate. Once the translated documents are obtained, they were subjected to a Re-ranking process. The basic re-rank model was developed by me while the LSI based re-rank model was obtained through GitHub.

The queries required for evaluations were obtained by distributing a Google form among several people as well as the evaluation process which was conducted that happened through Google forms. A rigorous process included analyzing the data obtained which was done by me and finally, the decisions were obtained.

Acknowledgement

My sincere gratitude to my research supervisor, Dr. A.R Weerasinghe, Senior Lecturer of the University of Colombo School of Computing, and my research co-supervisor, Dr. B.H.R. Pushpananda, Senior Lecturer of University of Colombo School of Computing for the guidance, immense support and valuable advice throughout the research.

I would also like to thank Dr. G.P Seneviratne, Senior Lecturer of the University of Colombo School of Computing and Mr. W.V. Welgama, Lecturer of University of Colombo School of Computing for the valuable feedback on my research proposal and interim evaluation which helped me going forward conducting my research. I would like to extend my thanks to all the Lecturers who were present in my proposal defense and interim presentations and providing me with their valuable opinions. I also extend my gratitude to Dr. H.E.M.H.B. Ekanayake for the assistance provided as the final year computer science project coordinator.

A Special thanks go to all my friends who helped me by conducting brainstorming sessions and sharing their knowledge with me which were very useful when carrying out my research. I would also like to acknowledge the assistance and patience of family and being a great support to me. Finally, I would like to thank all the people who helped me to successfully complete my research.

Table of Contents

List of Figures

List of Tables

List of Acronyms

API	Application Programming Interface
BWESG	Bilingual Word Embedding Skip Gram
CBOW	Continuous Bag of Words
CLIR	Cross Language/Lingual Information Retrieval
HITS	Hyper-Link Induced Topic Search
HTML	HyperText Markup Language
IR	Information Retrieval
LSI	Latent Semantic Indexing
LT-GC	Linear Transformation Globally Corrected
LT-NN	Linear Transformation Nearest Neighbour
MAP	Mean Average Precision
NER	Named Entity Recognition
OT-IS	Orthogonal Transformation Inverted Softmax
OT-NN	Orthogonal Transformation Nearest Neighbour
POS	Part-Of-Speech
SMT	Statistical Machine Translation
SVD	Singular Value Decomposition
WE	Word Embeddings

Chapter 1 - Introduction

1.1 Background to the Research

A massive amount of knowledge is embedded in the world wide web and people use search engines to search for knowledge from the world wide web. It has been reported that Google processes 3.5 billion searches per day where 16%-20% of the queries have not been searched before [1]. This shows that people are curious about finding knowledge and they are satisfied with the knowledge received.

A search engine uses the concept of Information Retrieval (IR) where a query is matched against a large number of documents and the relevant results are retrieved. Most IR systems are implemented such that the query is matched with the documents written in the same language as the query.

There are around 7000 different languages in the world but among them, 23 languages account for half the population of the world [2]. Hence, most of the documents available on the internet will be in these popular languages. This becomes a problem for people who are not familiar with those languages. As a result, they are not able to search the internet for knowledge since IR systems, although contains a vast number of documents will match the query given in their native language with the limited number of documents available in that language. This procedure will not produce appropriate and valuable results due to the limitation of resources. This is a problem faced by the people of Sri Lanka as well whose native language is Sinhala.

The ample knowledge available online in many different languages should be accessible by anyone regardless of their native language or the languages they are familiar with. This causes IR systems with the challenge of matching a query given in one language with documents in different languages. This gives rise to Cross Language Information Retrieval (CLIR) where the query given in one language is matched with documents in another language.

An issue in CLIR is the ambiguity of queries since they are short and do not provide much context for interpreting the query. This causes a reduction in the accuracy of the results obtained. This is a problem that should be handled carefully since this is crucial in matching the query against the documents. Several techniques have been used to tackle the task of matching queries and documents in different languages by translating the query to the language of the document, translating the document to the language of the query, or translating both the query and document to an intermediary language. The pros and cons of each technique will be mentioned in the Literature review section and it could be seen that the query translation approach is the popular method due to its advantages. Several approaches have been used for query translation such as token to token translation using a machine-readable dictionary [3]–[5] employing a Statistical Machine Translation [6]–[8] or using corpus-based techniques [9].

CLIR is a field with a very wide scope as there are numerous languages around the world. It has been reported that insufficient attention has been given by the CLIR community to solve the Cross Language issue of the world wide web [10].

1.2 Research Problem and Research Questions

1.2.1 Research Problem

The knowledge embedded in the web is mostly inaccessible to those not fluent in English. Non-English users need to be able to search for knowledge in their mother tongue (in this case Sinhala) and be able to retrieve the relevant information in that language itself. But in the current situation, although people are able to query the web in Sinhala Language, it will not provide accurate or relevant results. As a result of this, people do not tend to browse the web as well as they are unable to search for what they want.

1.2.2 Research Questions

1. **How to convert the Sinhala search query into the equivalent query in English?**
   The most common language in browsing the web is English. Querying the web in English often produces the best results. So, in CLIR, when another language is used to search the web, the best approach is to find the equivalent English query of the source query. Several approaches have been used to translate the source query to English such as dictionary-based methods, machine translation methods, parallel corpora-based methods, and so on. A simple translation is merely not enough to extract the meaning of the query. The challenge here is to extract the meaning of the source query and find the equivalent English query using a suitable approach. Once the equivalent English query is obtained, a simple search would provide good results.

2. **How to evaluate and 're-rank' the results returned in order to present the user with the most relevant documents?**
   The results retrieved using the equivalent English query will be the best results for the English query. These results will be translated back to Sinhala before presenting it to the user. Once translated, these results might have a different order of importance with respect to the Sinhala search query. So, the results should be re-ranked in order to obtain the most relevant results.

1.3 Research Aim and Objectives

1.3.1 Research Aim

The main aim of the project is to provide the capability for people who are not fluent in English to search the web in Sinhala and retrieve the results in that language itself. This capability should replicate the behavior of searching the web in English as closely as possible.

1.3.2 Research Objectives

- **Convert the Sinhala search query into the equivalent query in English**
  An approach based on word embeddings will be taken as those have been proven to be effective. Using the source word embeddings, a mapping mechanism from the source language vector space to the target language vector space should be found out to obtain the equivalent target word embeddings and obtain the equivalent target query for the given source query.

- **Search the web using the equivalent English query**
  After obtaining the equivalent target query, the results will be retrieved using the Google Search API.

- **Translate the results to Sinhala and present it to the user**
  The results should be translated back to the language of the source query since the users are not fluent in English.

- **Re-rank the results returned in order to present the user with the most relevant documents**
  A Ranking mechanism should be decided in order to Re-Rank the results returned to be relevant to the source query.

- **Evaluate the usefulness of the process**
  The performance of the proposed approaches should be measured by comparing it to other well-known models in order to evaluate the usefulness of the process.

1.4 Justification for the Research

Extensive research has been conducted in the field of CLIR and Translation. A language possesses certain characteristics and each language is different from one another. For example, the English language has a “Subject, Verb, Object” structure for sentences while the Sinhala language follows a “Subject, Object, Verb” structure. Hence, a well-performed method for a particular language pair, might not necessarily be good for another language pair. To the best of my knowledge, no study has been conducted for

the Sinhala to English CLIR. Sinhala is a low resource language. So, finding suitable tools and technologies is another challenging task when performing such a study in the Sinhala language.

Google Translate is a powerful tool that can be used to achieve multilingual translation. It currently supports 104 languages and Sinhala is one of them. But there are certain instances where google translate does not provide accurate results. For example, the query "සංඛ්‍යාවක සාධක සොයාගන්නේ කෙසේද" when translated using Google translate returns "How to find the proof of a number" which is not correct. Also, some rare words like "දෑස" are transliterated and google translate returns "Desa" as the result. Google Translate has become a very powerful tool to obtain translations but due to the examples provided above, it is clear that Google Translate needs improvements as well. But the source code for Google translate is not publicly available. As a result, a separate model was decided to be created targeting the Sinhala Language which can be improved regularly.

Word Embeddings has been a concept that has been used for CLIR and translation tasks that have shown good performance for low resource languages as well. This concept can be inherited to improve the translation and provide accurate results in CLIR for the Sinhala language.

1.5 Methodology

To deal with the first research question of obtaining an equivalent English query for the given Sinhala query, a translation model will be created. This translation model will consist of two major components namely the Projection Component and Retrieval Component as shown in Figure 1.1.

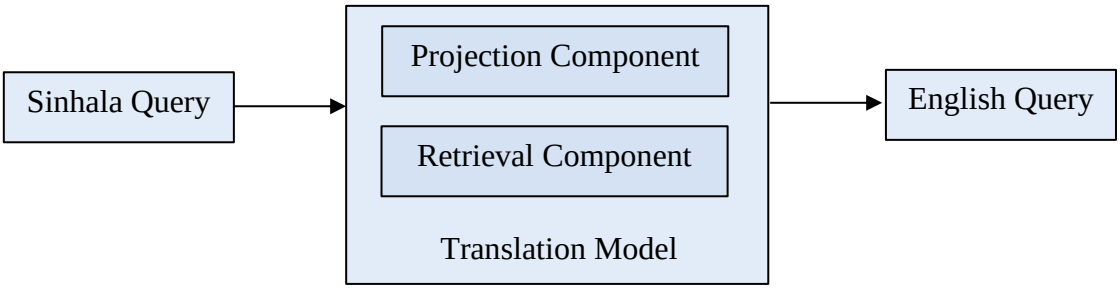


Figure 1.1: Architecture of the translation model

Several projection mechanisms, as well as several retrieval mechanisms, are available. For the projection mechanism, the Linear Transformation proposed by [11] and the Orthogonal Transformation using SVD proposed by [12] will be tried. The Linear Transformation mechanism will be combined with two Retrieval methods namely the Standard Nearest Neighbour Retrieval method and the Globally Corrected Neighbour Retrieval method as the retrieval mechanism. The Orthogonal Transformation mechanism will be combined with the Standard Nearest Neighbour Retrieval method and the Inverted SoftMax method as the retrieval mechanism. Hence, four different models will be experimented with, as the Translation model as summarized in Table 1.1.

Table 1.1: Translation Models

Translation Models		**Retrieval Component**	
		Standard Nearest Neighbour Retrieval	**Globally Corrected Neighbour Retrieval**
Projection Component	**Linear Transformation**	LT-NN Model	LT-GC Model
	Orthogonal Transformation	OT-NN Model	OT-GC Model

Once the English query is obtained, the Google Search API will be used to retrieve the documents. These documents will be translated back to Sinhala using the Google Translate API.

To deal with the second research question of determining an effect of Re-ranking documents before presenting it to the user, a Re-Ranking model will be created. Two types of re-ranking models will be considered namely the Basic Model and the Latent Semantic Indexing (LSI) based Model. The task of the Re-ranking model is to take the set of translated documents and rank them according to the model's criteria. The functionality of the Re-Ranking model indicating the Input and Output is shown in Figure 1.2.

Figure 1.2: Functionality of the re-ranking model

A user-based evaluation system will be carried out to evaluate the different models.

1.6 Delimitations of Scope

1.6.1 In Scope

The following will be covered under the scope of the project.

- **Only Sinhala language queries will be considered**
  Cross Language Information Retrieval mainly involves two languages. Since this project is done in a Sri Lankan context and the main language of Sri Lanka is Sinhala, it was decided to provide the ability to search the web using Sinhala queries.

- **Only existing web documents will be considered**
  Multimedia such as images and videos will not be considered when retrieving results. This is because the re-ranking models are based on the text content of the result.

1.6.2 Out Scope

The following will not be covered under the scope of the project.

- **Information extraction aspect will not be considered**
  When the query contains a question statement, normally google provides a summarized answer to the query using information extraction. This project will not try to emulate this step but will only focus on retrieving the relevant documents.

- **Queries with Named Entities will not be handled**
  Named Entity Recognition is a problem in NLP application and since there are not NER for Sinhala language, queries containing them won't be used.

- **Queries requesting location-specific information will not be considered**
  When browsing Google, Google is able to capture a lot of information including location about the person browsing the web and will return results specific to that location. In this study, since we will be using the Google Translate API, it is difficult to replicate the browser behavior.

- **Queries related to Technical Domain will not be considered**
  The technical domain consists of specialized terms that are difficult to interpret and translate. Hence, this study won't handle such queries.

1.7 Outline of the Dissertation

The Introduction chapter comprises of an introduction to the research domain which started by explaining the background to the research. Then the research problem was explained and research questions were presented. The research aim and objectives, the justification for the research, methodology, and delimitations of the scope were also discussed in the Introduction chapter. The Literature Review chapter explains the related work and the background theories associated with this research mainly discusses topics such as Query Translation, WE, and Document Re-Ranking.

The justification to use the chosen methodology followed by the design overview has been discussed in the Design chapter which concludes by presenting the sources that were used to obtain data for the research. The implementation of the models will be discussed in the Implementation chapter mainly as two subsections which are obtaining the equivalent English query for a given Sinhala query and re-ranking the documents with respect to the original query given in Sinhala.

The evaluation which has been conducted in two phases will be discussed in the Results and Evaluation chapter, along with the results and analysis of each phase. Finally, in the Conclusion chapter, the conclusions about the research questions, conclusions about the research problem, research contribution, limitations and future work to be carried out will be discussed extensively.

1.8 Summary

This chapter began by discussing the background to the research domain followed by the introduction to the research problem and listed two research questions associated with it. The motivation and the need for this research were discussed in the justification section thus conforming to the requirement of this research in the field of NLP. The research methodology was discussed in the next section followed by the section titled outline of the dissertation which has given a brief overview of this thesis. The chapter concluded by explaining the delimitations of the scope.

Chapter 2 - Literature Review

2.1 Query Translation

The main challenge in CLIR is to adopt a mechanism to cross the language barrier to match the user query in one language to a set of documents in another language. One such mechanism is to transform the query into the language of the documents. This process is easy since the size of the query is small but having a small size query introduces ambiguity since it does not provide enough context to interpret the query. Another approach adopted is to convert the documents to the language of the query. This approach reduces ambiguity but translating whole documents is a complicated task [13]. A summary of the advantages and disadvantages are shown in Table 2.1.

Table 2.1: Comparison of Query and Document Translation

Parameter	**Query Translation Approach**	**Document Translation Approach**
Query Length	Small	Large
Cost	Inexpensive	Expensive
Ambiguity	High	Low
Overhead	Low	High

As a result of the advantages of the Query Translation approach compared to the Document Translation Approach, the query translation approach is mostly used when performing CLIR. The main techniques in performing CLIR using the query translation approach are the Dictionary based approach as well the employing a Statistical Machine Translation based approach.

Dictionary-based approaches have been taken during the early days of handling CLIR tasks. This approach translates each source query word to the target query using the corresponding dictionary entry. Ambiguity and handling out-of-vocabulary words have been shown to cause problems and result in poor retrieval performance. Several approaches have been taken to tackle these situations. Using “local feedback” to expand

the target query has been shown to improve retrieval performance by reducing ambiguity [5]. A transliteration-based approach to handle Out-of-Vocabulary terms has been used in [3]. This study tackled the problem of ambiguity by using an iterative PageRank style algorithm to disambiguate the translated query words. [4] conducts experiments to identify the basic components required to handle the CLIR task using the dictionary-based method and identifies and presents the issues which could arise when performing such a task.

Statistical Machine Translation (SMT) is another approach used widely for CLIR. The important component which handles the translation of an SMT system is the statistical model whose parameters are learned by analysing a parallel bilingual corpus. translations. [6] translates a source query to the target query using a word alignment table which was learned using an SMT system with the use of aligned parallel sentences. [7] explores several statistical translation models such as context-independent token translation, token translation using phrase-dependent contexts, and token translation using sentence-dependent contexts.

2.2 Word Embeddings

2.2.1 Algorithms

The introduction of Word2Vec [14] created a massive breakthrough in the area of text representation. Word2Vec uses a neural network and is based on computing word embedding using a word's context. There are two approaches used by Word2Vec. The continuous-bag-of-words approach predicts the target word given its context while the skip-gram approach predicts the context using the target word.

While learning embeddings, Word2Vec does not consider frequent co-occurrence of words. For Word2Vec, having some context words appear more often than others carry no additional information. Glove [15], an unsupervised algorithm for learning word representations considers this frequency of co-occurrences as important and uses this data while learning word embeddings. Embeddings created using glove relates directly to the probability of word's co-occurrences in the corpus.

A problem in Word2Vec is the inability of generalizing to unknown words. FastText [16], [17] was able to overcome this issue. FastText is very similar to Word2Vec but it uses part of words and characters when learning word embeddings instead of using whole words. As a result of this, FastText is able to provide word embeddings for new words if the new word comprises the same characters of known ones. Another advantage of FastText is that it requires less training data since much information can be captured from a small piece of text. Hence FastText has been able to create word embedding models for more languages than other algorithms and these models are publicly available [18].

2.2.2 Features and Applications

Word Embeddings transforms a given word to a vector of real numbers. When a set of given words are visualized in a 2-Dimensional space, it can be seen that words with similar meanings are grouped together and close to each other. Also, when a set of words in two different languages is visualized, it could be observed that they would depict a similar arrangement in the space they are embedded in [11]. This feature has been exploited when developing Cross Language Word Embedding models to transform a word from one language to another [19].

Word Embedding has been shown to poses the advantage of handling words that are either not found in the training corpus or found less frequently [20]. This has caused the word embedding to be seen as a powerful tool in handling text data.

Word Embedding has been proven to be useful in tasks such as synonym detection, word analogy, and semantic word similarity tasks [15], [21]–[23]. Apart from these, word embeddings have been effectively used to tackle some of the main problems in NLP such as Named Entity Recognition (NER), Part-Of-Speech (POS) tagging, chunking and semantic role labeling [24], [25].

2.2.3 Mapping-Based Cross Language Word Embedding Models

The mapping-based approaches learn word embeddings using the large monolingual corpus. Using the learned embeddings, this attempts to produce a transformation matrix with the help of bilingual dictionaries in order to map the embedding of words in the source language space to the target language space. The mapping-based approaches require word-level aligned parallel data which are available in the form of bilingual dictionaries. This is the most popular method in Cross-Lingual Word Embedding models due to its simplicity and ease of use as well as since Document aligned comparable corpora are not always available especially for low resource languages [26].

Four different types of mapping methods have been proposed in [26].

1. Regression methods adapt a Linear Transformation and attempt to map source language words to target language words by maximizing the similarity between them.
2. Orthogonal methods employ a similar approach to the Regression method with the constraint that the transformation should be orthogonal.
3. Canonical methods project the source and language word embeddings to a new common space and try to maximize the similarity between them
4. Margin methods find correct translations and other candidates and try to maximize the margin between them.

A very common regression method approach is the Linear Transformation introduced in [11]. The geometric arrangements of words and their corresponding translations were similar when each of them was visualized in their respective 2-Dimensional space. This was the fact which backed up the method of projecting a word in one embedding space to another embedding space in order to find the relevant translation. This has used a dictionary of 5000 most common words of the source language to learn this projection.

Several improvements of this Linear Transformation have been proposed. [27] proposes that the monolingual embeddings should be of unit length. [28] suggests that in order to have all the training instances contribute equally to the objective, a length normalization step has to be performed.

Orthogonal Mapping methods have been proposed to improve on the Regression method by suggesting that the transformation should be orthogonal. [12] proposed that this Transformation can be learned by employing Singular Value Decomposition (SVD). This study has improved the Precision at 1 of the Linear Transformation model introduced in [11] from 34% to 43%. Also, it has been shown that among the four types of mapping models, the Orthogonal Model is the most commonly adopted method [26].

2.2.4 Word Embedding Based CLIR

A novel architecture called BWESG (Bilingual Word Embedding Skip Gram) is introduced in [20] which uses a corpus of aligned documents in order to learn word embeddings. In the BWESG model, each aligned document pair are merged and the words are randomly shuffled to obtain a 'pseudo-bilingual' document which is a document consisting of words from both source and target languages. Once the pseudo-bilingual documents for all the document pairs are obtained, they are fed to a skip-gram model in order to learn embeddings for words from both the languages which would result in embeddings that are shared in a common vector space. Query and document vectors are obtained by combining the vectors of individual words comprising the query or document. For the Information Retrieval process, the vector for each issued query is obtained and the similarity score between the query vector and all the document vectors are calculated using the standard cosine similarity measure and the documents are ranked in the descending order of the similarity score.

A CLIR approach using the Linear Transformation to translate source queries to target queries is discussed in [19]. A monolingual corpus is trained using the Word2Vec tool to obtain word embeddings and a linear projection is learned from the source vector space to the target vector space. In order to obtain the target query words from the projections, several methods have been proposed. These methods include picking the k best translations after computing the cosine similarity between the projected word and the target words, following the same procedure but assigning weights to query words and computing the similarity vector for each query. Dictionary and Google Translate in different combinations have also been used with these methods to create hybrid models. Named entities are handled using transliterations. The hybrid model that uses both

Google Translate and dictionary has been shown to outperform the English monolingual baseline by 15%. The approach discussed in this paper paves a good way to perform CLIR tasks effectively using the Linear Transformation method.

Another approach for CLIR using Word Embedding has been performed by adopting multilingual word clusters to perform query translations [29]. Once multilingual word embeddings are obtained in a common space, cosine similarities between word vectors are calculated. Then a Graph is created where the vertices represent the words while an edge exists between two vertices only if the cosine similarity between the two words represented by the vertices is greater than a threshold value of 0.5. Once the Graph is constructed, a community detection algorithm called the Louvain algorithm is applied to the graph clusters the vertices. The Query translation process occurs in such a way that for each word in a query, the cluster it belongs is obtained and the top t English words which are most similar to the source query word are extracted. This study also has used different combinations of Google Translate and a dictionary along with the proposed cluster-based approach to creating hybrid models where the hybrid model which consists of the cluster-based approach, dictionary, and Google translate has shown to perform well.

2.3 Document Re-Ranking

In a search engine, once the relevant documents for a given query are obtained, a re-ranking mechanism is applied to the documents before presenting them to the user. The reason for this is since users very rarely or almost never check the second or beyond pages of the results, it is important to present the user with the most relevant and important documents as the first few results. Several approaches have been taken to address the re-ranking of documents where some approaches look at how important a web page is whereas some tries to measure relevance looking at the content of the document. Approaches that combine both these methods also have been tried out [30].

PageRank [31] is the approach taken by Google to re-rank its search results. The importance of a page is measured by the importance of incoming links to a page. The core idea is that if a web page has higher importance incoming links, then the links

outgoing from this page will also be important. So, it uses a mechanism of backlinks to rank documents such that if a sum of ranks of backlinks of a document is high, then the rank of the document is high. This method uses a graph-based structure to calculate the ranks. Along with PageRank, Google uses many other factors to present users with the most relevant documents.

PageRank divides the rank of a document equally when assigning scores to outgoing links. A slight variation to this called the Weighted PageRank where important pages are given a higher score is proposed in [32]. The importance of a page is measured by the weighted values of incoming and outgoing links. Experiments have been performed to compare the PageRank and Weighted PageRank methods based on a relevancy rule where the Weighted PageRank method has shown to produce larger relevancy values indicating that it performs well than the PageRank method.

Page Content Rank algorithm [33] looks at the content of the document to determine its importance. It considers that the importance of a document depends on the importance of the terms contained in the documents where the term importance is measured using factors such as term frequency and term position in a document.

Another approach that uses the link structure of web pages is the Hyper-Link Induced Topic Search (HITS) algorithm [34], [35]. This considers two categories of pages called Hubs and Authorities. The algorithm is based on the relationship between the Hubs and Authorities. Authorities are the pages that contain relevant information to a query while Hubs are the pages that point to many other pages including Authority pages. This algorithm assigns two values to a page namely the Hub value which is the value of the links to other pages and the Authority value which is the value given depending on the content of the page. So, this algorithm does not only consider the link structure but also consider the content of documents as well.

A comparison of the Re-Ranking algorithm is shown in Table 2.2 [30].

Table 2.2: Comparison of Page Ranking Algorithms [30]

Algorithm	**PageRank**	**Weighted PageRank**	**Page Content Rank**	**HITS**
Input Parameters	Backlinks	Backlinks, Forward Links	Content	Backlinks, Forward Links, Content
Complexity	O(log N)	< O(log N)	O(m) (m - Total Number of occurrences of query terms)	O(log N) (Higher than WPR)
Relevancy	Less	Less (Higher than PR)	More	More (Less than PCR)
Importance	More	More	Less	Less
Quality of Result	Medium	Higher than PR	Approximately equal to WPR	Less than PR

2.4 Summary

This chapter mainly focused on giving a thorough review of the literature associated with this research. The literature review started by discussing the concept CLIR by elaborating on the main challenges and the techniques that were used to overcome those challenges. The next section discussed WE by introducing the algorithms used, features of WE with their advantages and disadvantages along with the applications of WE, Cross Language WE models, and WE based CLIR tasks. The next section concluded the chapter by elaborating Document Re-Ranking.

Chapter 3 - Design

3.1 Justification for the Methodology

Word embedding has shown to perform well on many Natural Language Processing tasks including CLIR as mentioned in Section 2.2. Also looking at the various advantages of word embeddings, it could be used to extract the meaning of the query in order to perform a translation.

The Linear Transformation proposed in [11] is the basic approach that has been used to project an embedding to obtain translations. Since there have been no embedding-based translations from Sinhala to English language, it would be better to try out a basic model to check its performance to handle this translation. The Orthogonal transformation method which was decided to use to learn the projection function has a similar approach and it has shown to perform well as discussed in the Literature review section. Hence a basic model and a well-performed model would be used to handle the Sinhala to English translation.

For the Re-ranking process also, a basic model is considered to check the initial effect of re-ranking in order to determine any change in the order of importance. Along with the basic model, an LSI based model which is a popular topic modeling approach was decided to use since it could find the relationship between the original query and the translated document and produce relevant results.

Most of the work that address the CLIR task, retrieve documents for a user query from a set of available documents. Those work label the available documents as relevant or not for user queries prior to retrieving documents. Then they use the Mean Average Precision (MAP) as the evaluation metric to measure the performance of the proposed methods [19],[29].

In this approach, since the results are retrieved from the web, it is not possible to label the documents. So, a user-based evaluation purpose will be used as suggested in [36].

3.2 Design Overview

Figure 3.1 shows the High-Level Design of the System.

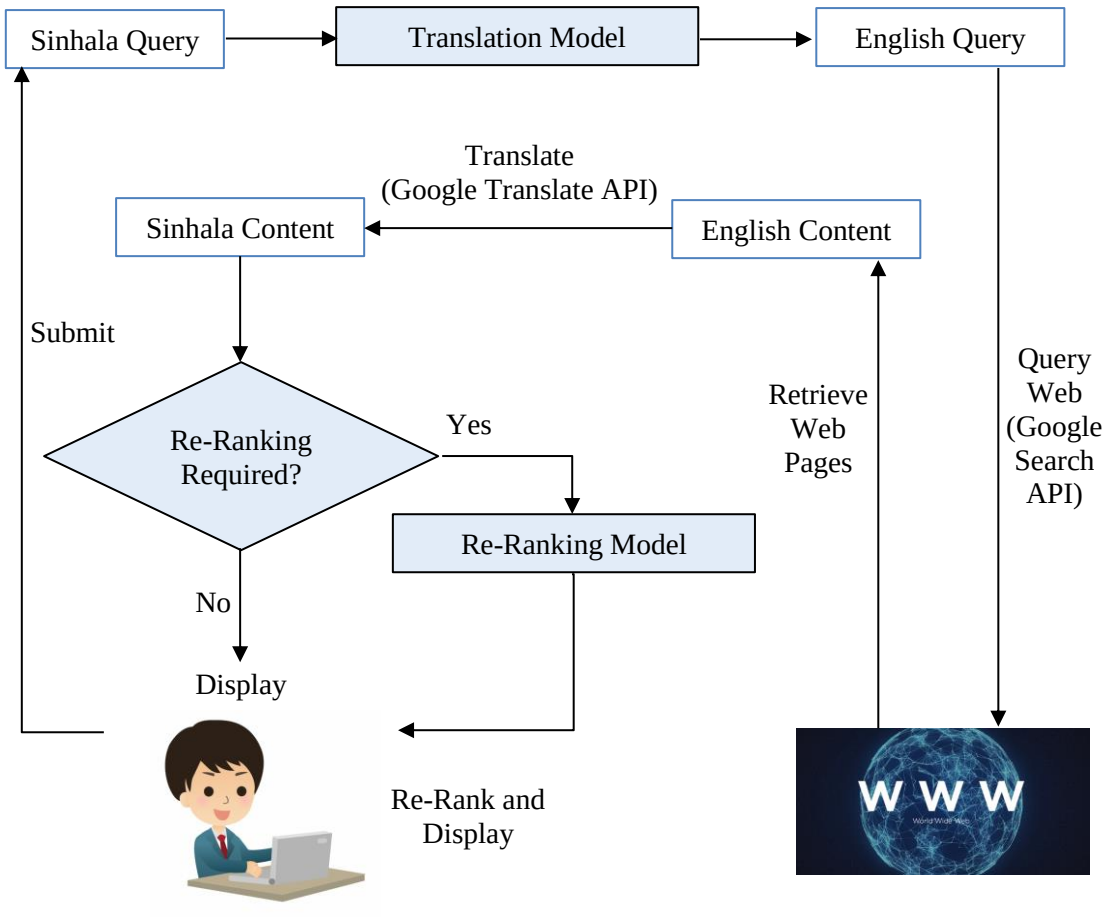


Figure 3.1: High-level design of the system

The research design consists of two main components.

1. Translation Model
2. Re-Ranking Model

Several translation models will be used which was discussed in Section 1.5. Using the results of the user evaluation, the best Translation model will be used in the Final System.

Two different Re-Ranking mechanisms will be adapted to handle the Re-Ranking of documents and the best model will be decided based on user evaluation. Based on the final evaluation, it will be decided whether or not to embed the Re-ranking model in the Final System.

The research design consists of the following processes.

1. Obtaining Word Embeddings for the Source and Target Languages
2. Obtaining the Equivalent English query for a given Sinhala query. This is the task of the Translation Model
3. Retrieving the documents corresponding to the Query
4. Translating the Documents to the source language.
5. Re-ranking the documents with respect to the Original query given in Sinhala.
6. Evaluating the Models using a user-based evaluation process

3.2.1 Obtaining Word Embeddings for the Source and Target Languages

Since the search queries are given in Sinhala, it will be the source language while English will be the target language.

A large monolingual corpus is needed to train a good word embedding model. The source of the Sinhala corpus which was used to train the Sinhala Word Embedding model including its details is given in Section 3.3. This will be the Source embeddings. The training procedure is mentioned in Section 4.

The pre-trained English embedding model given by FastText was used as the target embeddings. The details of the target embeddings, as well as their specifications, are mentioned in Section 3.3.

3.2.2 Obtaining the Equivalent English Query for a Given Sinhala Query

Learning the Projection of Word Embeddings from the Source to the Target Language Space

As mentioned in Section 2, a projection can be learned which maps embeddings from the source language space to the target language space. The learning procedure of the projection function is mentioned in Section 4.

Once the projection is learned, a source word can be mapped to the target embedding space as follows.

Step 1: Obtain the word embedding corresponding to the given source word
Step 2: Apply the projection function to the obtained source embedding

This procedure is shown in Figure 3.2.

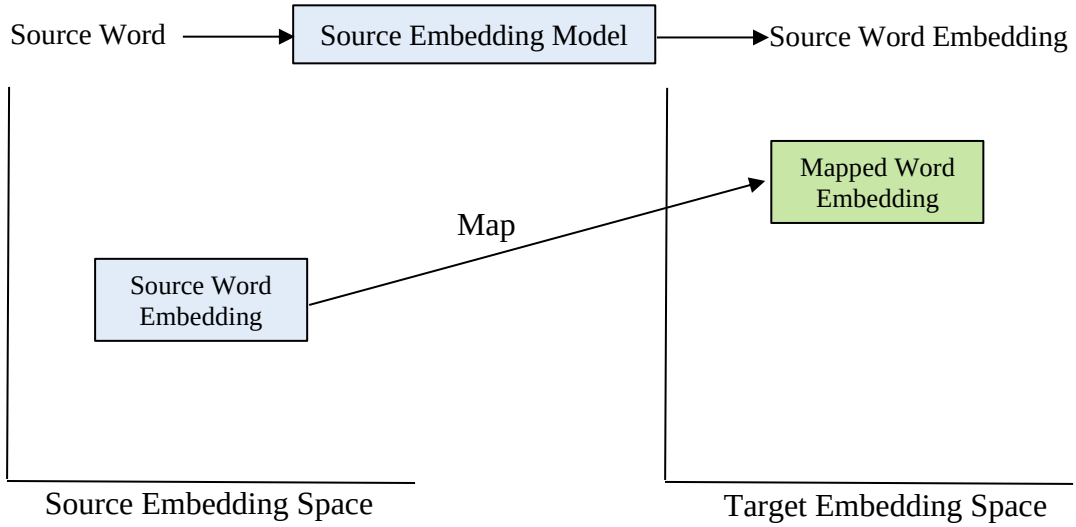


Figure 3.2: Process involved in the projection function

Applying the projection function is the task of the Projection component of the Translation model. Its task is to take a source word and project its embedding to the target embedding space.

Retrieving the Translation in the Target Language

Projecting a source word embedding to the target embedding space will not result in a translation. It will be just an embedding in the target embedding space. The embeddings of the target language will also be in this embedding space around the mapped embedding as shown in Figure 3.3. In order to obtain a translation, this projected embedding should be linked to an embedding in the target embedding space since they are the embedding that represents words in the target language. Once a target embedding is linked to the mapped source embedding, the word corresponding to the linked target

embedding can be obtained using the English word embedding model. This will be the translation for the given source word. This linking mechanism is known as the Retrieval method and as mentioned in Section 1.5, several retrieval methods will be used to retrieve translations. Each retrieval method will have a specific criterion to link the mapped embedding to a target embedding as described in Section 2.

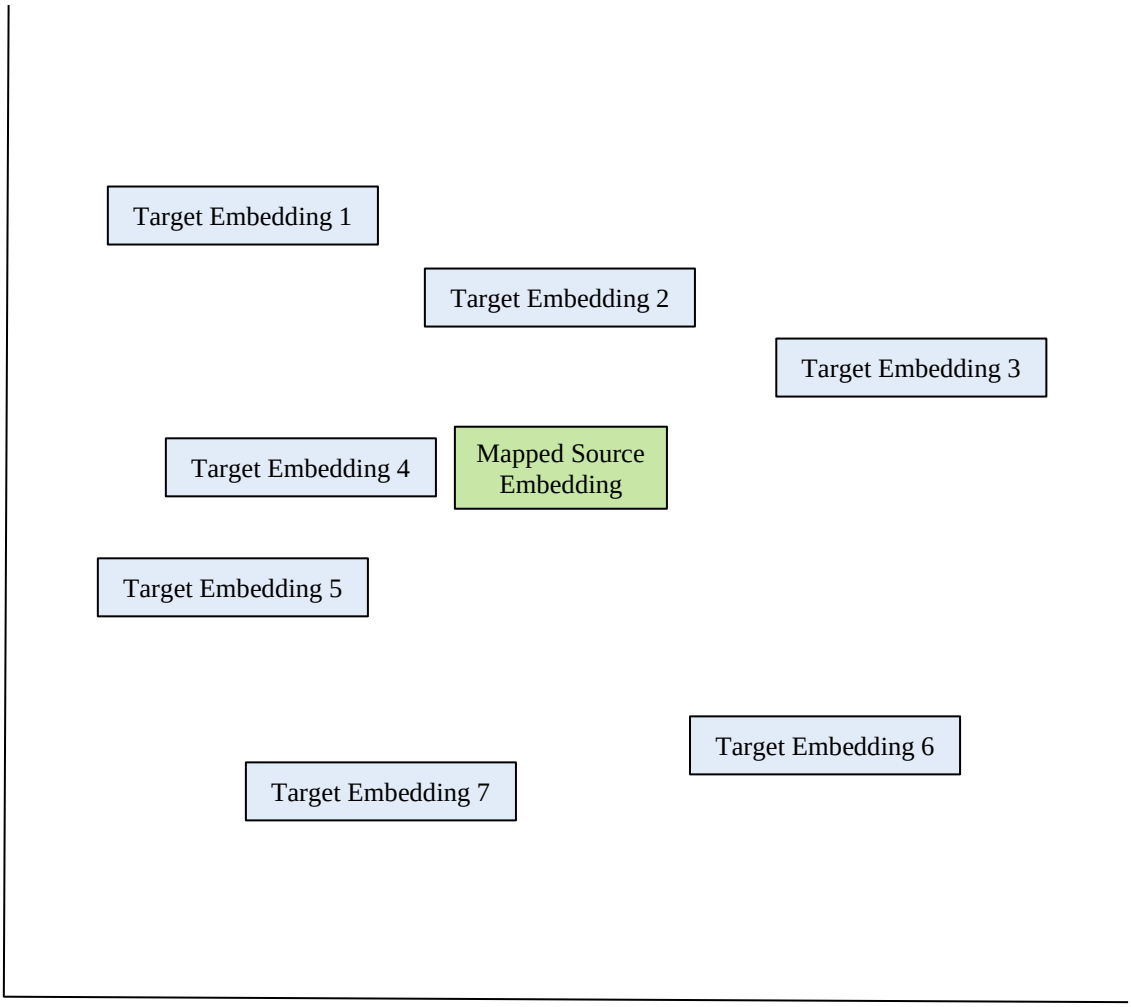


Figure 3.3: Word Embeddings in the target language space

Finding the translation using a retrieval method is the task of the Retrieval component of the Translation model. This component receives the embedding of the source word projected to the target space and it returns a translated word corresponding to the source word.

The overall architecture of the Translation Model is shown in Figure 3.4.

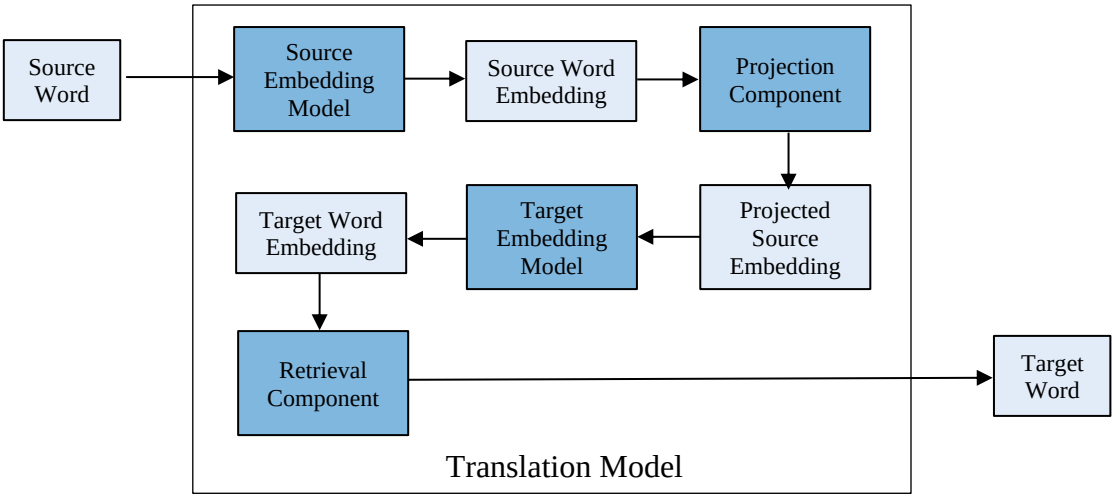


Figure 3.4: Overall architecture of the translation model

Query Translation Process

The query translation process takes place according to the following procedure.

Step 1: The source query is split into a list of words.
Step 2: Each word is fed into the Translation model and the Target word is obtained.
Step 3: The set of target words are combined to obtain the target query.

3.2.3 Document Retrieval Process

Once the source query is translated to the corresponding target query, Google Search API will be used to retrieve the relevant documents for the target query. (Refer Appendix D.3 for implementation details)

3.2.4 Re-Ranking the Documents with respect to the Original Query Given in Sinhala

The retrieved documents will be in English. In order to perform a re-ranking of the documents with respect to the original Sinhala query, these documents should be translated to Sinhala. This translation is done using Google Translate. Two models will be considered to Re-rank the documents.

The first model is a basic model that calculates the number of times the terms in a search query appear in the document. Then the documents will be ordered in the descending order of the value.

The second model will employ the Latent Semantic Indexing (LSI) which is a Topic Modelling approach to Re-rank the documents. Latent semantic indexing (LSI) is an indexing and retrieval method that uses a mathematical technique called singular value decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. The idea behind using the LSI model is to find the relationship between the documents and the search query.

The document translation process and the re-ranking are explained further in Section 4.

3.2.5 User Evaluation Process

As mentioned in Section 1.5, two types of evaluations will be conducted.

Evaluation 1 – Evaluating the performance of the re-ranking models

The focus of Evaluation 1 is to determine which of the two re-ranking mechanisms is performing well on a set of queries as well as determining the effect of re-ranking of documents. Hence, three types of models will be considered during this evaluation. The Sinhala queries will be translated to English using Google Translate.

Evaluation 1 – Evaluating the performance of the overall models

The focus of Evaluation 2 will be to determine which overall model performs well. There will be 4 or 5 models depending on the results of Evaluation 1.

A detailed explanation of the Evaluation process will be discussed in Section 5.

3.3 Data

Pre-Trained Word Vectors are available for English Languages that were trained on Common Crawl and Wikipedia using FastText. These models were trained using CBOW with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives.

The data set used to create word vectors for Sinhala words were obtained from [37]. The filtered Sinhala common crawl data set contains 5,178,491 sentences with 110,270,445 words.

3.4 Summary

This chapter mainly explained the design aspects of the research. The justification to choose the methodology stated in Section 1.5 was explained in the first section of this chapter. Then Design Overview has been explained which gives an idea of the whole process which takes place starting from submitting the user query to retrieving the documents in the source language. The chapter concluded by discussing the sources which were used to obtain data for the research.

Chapter 4 - Implementation

4.1 Obtaining the Equivalent English Query for a Given Sinhala Query

4.1.1 Obtaining Word Embeddings for the Source and Target Languages

Gensim provides the functionality to train word embedding models using the FastText algorithm. In order to obtain word embeddings for the Sinhala language, a large dataset should be trained using the FastText algorithm. The 100-million-word Sinhala corpus mentioned in Section 3.3 was used to create word embeddings for the Sinhala language.

Data Preparation

The corpus was initially subjected to a mild preprocessing where English characters, special characters and numbers were removed. Then it was transformed into a list of sentences where each sentence is a list of words. This resulted in a two-dimensional list with each element being a word. This is the form that should be fed to the FastText algorithm defined in Gensim. It was observed that there were words with some leading and trailing punctuation (Eg: බල්ලා “,'බල්ලා). These words had to be subjected to a transformation which eliminates these punctuations and results in the pure Sinhala word. The following procedure was used to obtain pure Sinhala words.

Step 1: Read the string from the beginning replacing each character by “” (empty) until the first Sinhala character was obtained.

Step 2: Once Step 1 was completed (i.e. first Sinhala character at the beginning was obtained), read the string from the end replacing each character by “” (empty) until the first Sinhala character was obtained.

Step 3: Once Step 2 was completed (i.e. first Sinhala character at the end was obtained), return the result.

Refer Appendix D.1 for implementation details.

Sinhala Word Embedding Model Creation

Once this preprocessing and tokenization tasks were completed, the two-dimensional list was used to train the Word Embedding model. Only one line of code is required to train the model as shown in Figure 4.1.

```
from gensim.models import FastText
si_we_model = FastText(sentences=si_sentences, size=300, window=5, min_count=5, workers=4, sg=0)
```

Figure 4.1: Code segment for learning the Word Embedding model

Several parameters are required to be given to the model

sentences: The list of split sentences.

size: The dimensionality of the embedding vector

window: The number of context words the model should look at

min_count: Model ignores the words with a total count less than this number.

workers: The number of threads being used

sg: Whether to use skip-gram or CBOW

The functioning of WE Models

The functioning of the Sinhala WE model was checked using the following tasks and the results are shown in Table 4.1.

1. Finding Similar Words
2. The similarity between word pairs
3. Performing vector operations
4. Finding a word that doesn't relate when a set of words are given

Table 4.1: Functioning of the Sinhala WE Model

Task	Example Input	Output
1	දෙක	දෙකණ, දෙකහ, දෙකලව
2	දෙක, තුන	0.66485494
3	කාන්තාව + රජු - පිරිමියා	රජ
4	බළලා බල්ලා දෙක සිංහයා	දෙක

As mentioned in Section 3.3, the Pre-trained model available from FastText was used as the English WE model and the Functioning of the English WE model was also checked using the tasks mentioned in the above section and the results are shown in Table 4.2.

Table 4.2: Functioning of the English WE Model

Task	Example Input	Output
1	dog	dogs, puppy, pup
2	two three	0.94294167
3	woman + king – man	queen
4	cat dog two lion	two

Since both the Word Embedding models showed decent results, these models were used for further processing tasks.

4.1.2 Learning the Projection of Word Embeddings from the Source to the Target Language Space

Linear Transformation proposed in [15] and a method that uses the Orthogonal Transformation proposed in [16] was used to learn the Projections.

Dictionary Generation

In order to learn the projection, a Sinhala-English dictionary is required. To obtain the dictionary, the most frequent 6500 words of the 100-million-word corpus was obtained. These words were translated into English using Google Translate. The 6500 entries and their translations were one word each. (Eg: බල්ලා - dog). If a frequent word or its translation contained more than one word, they were removed. The reason was that each word is mapped to a vector and if an entry has more than one word, a vector calculation has to be performed to map that entry to a single vector. For simplicity, entries with more than two words were removed. Then it was split with 5000 pairs as the training dictionary and 1500 words as the testing dictionary. Both these dictionaries were saved as text files. The format of the dictionary should be a list where each entry is a word and its corresponding translation. The text file was converted to the list using the Algorithm 1. (Refer Appendix D.2 for implementation details)

Algorithm 1: Converting Dictionary into a List

```
1: procedure dictionary_to_list( dictionary_file )
2:     si_words ← [ ]
3:     en_words ← [ ]
4:     file_content ← read ( dictionary_file )
5:     for word_pair in file_content do
6:         split_pair ← split ( word_pair )
7:         if ( length ( split_pair) < 2 ) then
8:             si_words.add ( split_pair[0] )
9:             en_words.add ( split_pair[1] )
10:        end if
11:    end for
12:    si_en_dict ← [ ]
13:    for si_word, en_word in si_words, en_words do
14:        si_en_dict.add ( ( si_word, en_word ) )
15:    end for
16:    return si_en_dict
17: end procedure
```

Gensim library provides a function that performs the Linear Transformation. It requires the source and target language word embeddings and a dictionary. The code segment is shown in Figure 4.2.

```
from gensim.models import TranslationMatrix
trans_model = TranslationMatrix(si_we_model, en_we_model, word_pairs=training_dict)
```

Figure 4.2: Code segment for learning the translation model

Once the projection model is trained, it can project embeddings from the Sinhala language space to the English language space. Two types of methods were adapted to retrieve the correct English translations from the projections with the Linear Transformation method.

- Method 1 (LT-NN) - Standard Nearest Neighbor Retrieval method which retrieves translations based on cosine similarity.
- Method 2 - Globally corrected Neighbor Retrieval method which tries to reduce the hubness problem. (LT-GC)

Both of these methods were tested using the testing dictionary of 1500 words and their performance is reported in Section 5.1.

The implementation of the Orthogonal Transformation using SVD, proposed in [12] is also publicly available and it provides a guide to train a translation model using any source and target language pairs. The training consists of the steps as shown in Figure 4.3.

```
# form the training matrices
source_matrix, target_matrix = make_training_matrices(
    si_dictionary, en_dictionary, output)

# learn the transformation
transform = learn_transformation(source_matrix, target_matrix)

#apply the transformation
si_dictionary.apply_transform(transform)
```

Figure 4.3: Steps involved in training the translation model

This method also proposed two types of translation retrieval methods.

- Method 1 (OT-NN) - Standard Nearest Neighbor Retrieval method which is similar to the previous.
- Method 2 (OT-IS) - Inverted SoftMax function where at test time, the SoftMax used for finding the translation of a word is inverted and probability over the source words are normalized instead of the target words. Method 2 is also proposed as a solution to the Hubness problem.

The performance of these methods is reported in Section 5.

4.2 Re-Ranking the Documents with respect to the Original Query Given in Sinhala

Once the search query is translated to the equivalent English query, web documents were retrieved using the Google Search API. The retrieved results are in JSON format. The following steps were performed to Re-rank the documents.

4.2.1 Web Content Extraction

The JSON object does not contain the content of the web pages. As a result, the web content should be extracted using the URL of the web pages.

The URL should be passed to a request object and using the get() method of the request object, the structure of the web pages can be obtained. This result should be passed through an HTML parser to remove the HTML content. Beautiful Soup is a python library that is able to parse HTML. From the resulting content, all the text elements should be extracted. Then the text elements are passed to a function that removes unwanted elements such as scripts, document, head, etc. This function returns a list which should be converted to a string. Finally, common escape sequences are removed and the resulting content is taken as the web document content.

4.2.2 Web Content Translation

The content extracted in the previous step is in English. This has to be translated into Sinhala in order to perform the re-ranking. The extracted content is translated to Sinhala using Google Translate.

Google Translate has a restriction on the maximum number of bytes. Only a string of 5000 bytes can be translated at once using Google translate. Some of the documents extracted contained more than 5000 bytes. As a result, the extracted contents should be broken down into chunks, translated separately and combine the translated chunks to obtain the final translated documents.

4.2.3 Model Application

Once the translated contents are obtained, the following models will be applied to re-rank the documents.

Basic Model

The idea of the basic model is to count the number of times the terms in the search query appears in the document. To achieve this, the frequency of the unique terms appearing in the document should be obtained. Nltk provides the FreqDist() function which performs this calculation. FreqDist() uses a tokenized list of words as the input. As a result, the translated documents should be tokenized. Then the tokenized content is fed to the FreqDist() function which returns the frequency of the terms in the documents. Then it is possible to calculate the number of times the search terms appear in the documents. Once this calculation is performed, the result should be ordered in the descending order of the value. The final result obtained after ordering will be the ranks of the documents with respect to the search query in Sinhala using the basic model.

LSI based Model

An implementation of the LSI model to rank the documents can be found in [38]. This implementation was used to rank the translated documents using the LSI model. It takes the translated documents as a list and the search query as a string in order to create the model which returns the ranked results. The result obtained from the process() function of the implementation will be the ranks of the documents with respect to the search query in Sinhala using the LSI model.

Refer Appendix D.4 for implementation details.

4.3 Summary

This chapter explains the code segments relevant to each step involved in the design process. The main two steps are obtaining the equivalent English query for a given Sinhala query and Re-ranking the documents with respect to the original Sinhala query. Each sub-step involved in these two main steps are explained with relevant code segments.

Chapter 5 - Results and Evaluation

5.1 Results of the Translation Models

The testing dictionary of 1500 words created as described in Section 4.1.2 was used to evaluate the Translation models. The performance of the models is shown in Table 5.1. During a translation, since the best possible results are needed, the Precision at 1 (P1) measure was considered. The P1 measure was calculated by finding the percentage of Sinhala words in the testing dictionary which gave the correct English word according to the testing dictionary.

Table 5.1: Performance of the Translation Models

Model	**P1 measure**
LT-NN	35%
LT-GC	28.4%
OT-NN	31.467%
OT-IS	32.8%

Since the LT-NN model has the highest P1 measure, that model will be used for subsequent processes of Re-ranking and user evaluation.

5.2 Evaluation Process

5.2.1 Query Collection

In order to evaluate the different models, a diverse set of Sinhala queries is required. Hence, a Google form was created and distributed among university students.

The students were asked to provide 2 queries with a length restriction of 2-5. It was also mentioned to avoid providing queries containing Named Entities as well as queries specific for a country or region. The reason for posing these restrictions are mentioned in Section 1.7.

20 queries were collected for the evaluation process and the selected queries are shown in Appendix C.

5.2.2 Evaluation 1 - Evaluating the Performance of the Re-Ranking Models

Procedure

The focus of evaluation 1 was to determine the best ranking model from the basic model and the LSI based model.

The 20 queries were translated using Google Translate and 50 documents were retrieved using the Google Search API. These are the documents obtained without any re-ranking process. The top document was extracted to be used for the evaluation process. Next, these documents were separately re-ranked using both the re-ranking models and the top-ranked documents were extracted.

The three documents were viewed through the Google Chrome browser and translated to Sinhala using the Google Translate plugin for Google Chrome. Then PDF documents of these were obtained using the print option of Google Chrome.

Three models were considered for this evaluation and the top document of these three models were used for the evaluation.

Model 1: Model without Re-ranking
Model 2: Model using the basic Re-ranking model
Model 3: Model using the LSI based Re-ranking model

Four separate Google Forms were created each having five queries to evaluate. These forms are shown in Appendix A. Each query had three documents corresponding to the three models where document 1 corresponds to Model 1, document 2 corresponds to Model 2 and document 3 corresponds to Model 3. The evaluator's task was to read the three documents and rank them according to the opinion of the evaluator. Each query was ranked by 10 evaluators.

Results and Analysis

The analysis was carried out by calculating a score for each model using eq. (1) where $Score_{rank}$ is the Score obtained for individual ranks. $Score_{rank}$ was calculated using eq. (2) where $Count_{rank}$ is the number of times the document corresponding to a model obtained that particular rank and $Weight_{rank}$ is the weight assigned to that particular rank. The weight assigned to Rank 1, Rank 2, Rank 3 and Rank 4 is 1, 2, 3 and 4 respectively.

$$Model\ Score = \sum Score_{rank} \quad (1)$$

$$Score_{rank} = Count_{rank} * Weight_{rank} \quad (2)$$

The model which gets the lowest score will be considered as the best since most of the participants have given a top rank to the document corresponding to that model which would result in a low score. An example of a set of ranks given to documents is shown in Table 5.2 where Ranks given by five evaluators are included.

Table 5.2: Example Set of Ranks for Documents

Document	Ranks	Score
Document 1	1, 1, 2, 1, 1	6
Document 2	2, 2, 1, 2, 2	9

Assume Document 1 corresponds to model A and Document 2 corresponds to model B. Then the score of model A would be calculated as follows. Using eq. (2), Score for Rank 1 will be (4 * 1) and Score for Rank 2 will be (1 * 2). Using eq. (1) the final score is calculated as (4 * 1) + (1 * 2) which results in a value of 6. Similarly, the model B will obtain a score of 9. Among these two models, model A will be considered the best performing model since it has the lowest score compared to model B.

The results obtained for the four question forms are given in Appendix A. The model scores for each form as well as the final model scores are given in Table 5.3.

Table 5.3: Results of Evaluation 1

Model	Form 1 Score	Form 2 Score	Form 3 Score	Form 4 Score	Final Score
Model 1	83	82	92	81	338
Model 2	118	114	113	116	461
Model 3	99	104	95	103	401

From Table 5.3, it can be seen that Model 1 has obtained the lowest score in each of the four forms and hence has the lowest final score. This shows that Model 1 performs consistently and is the best Model compared to the other two. Model 1 corresponds to the model where the query was translated using Google Translate and No Re-ranking of documents is performed. Hence, these results show that queries translated using Google Translate performs well when no re-ranking is applied.

From the remaining two models which were subjected to Re-ranking, Model 3 has shown to perform well since it has the 2nd lowest rank and it is consistent across the four forms. Model 3 corresponds to the model where queries were translated using Google Translate and the LSI based Re-ranking model was applied. Hence, the LSI-based re-ranking model can be determined as performing well than the basic re-ranking model.

5.2.3 Evaluation 2 - Evaluating the Performance of the Overall Models

Procedure

The focus of Evaluation 2 is to determine the best performing overall model which once identified, could be used to implement the final system. This evaluation follows a similar procedure as Evaluation 1 but considers four models and the top document of these four models was used for the evaluation.

Since this study tries to provide a solution to the current Google Sinhala search, this is used as a model in this evaluation to compare the performance of other models against Google Search. LT-NN model showed to perform well among the other translation models, and hence, a model with LT-NN as the Translation Model without re-ranking is considered here. Since the results of evaluation 1 showed that Google Translate performs well without re-ranking, Google Translate Model without Re-ranking is considered a model in this evaluation. The LSI based re-ranking model showed to perform better than the basic re-rank model. So, this is combined with the LT-NN model to determine the performance of the LT-NN model with re-ranking.

The four models used in evaluation 2 are summarized as follows.

Model 1: Current Google Search
Model 2: Google Translate Model without Re-ranking
Model 3: LT-NN Translation Model without Re-ranking
Model 4: LT-NN Translation Model with LSI based Re-Ranking

10 queries were randomly selected from the initial list of 20 queries. Two separate Google Forms were created each having five queries to evaluate. These forms are shown in Appendix B. Each query had four documents corresponding to the four models where document 1 corresponds to Model 1, document 2 corresponds to Model 2, document 3 corresponds to Model 3 and document 4 corresponds to Model 4. The evaluator's task was to read the four documents and rank them according to the opinion of the evaluator. Each query was ranked by 10 evaluators.

Results and Analysis

A similar approach as discussed in Section 5.2.2 was taken to calculate the score of each model. The results obtained for the four question forms are given in Appendix B. The model scores for each form as well as the final model scores are given in Table 5.4.

Table 5.4: Results of Evaluation 2

Model	Form 1 Score	Form 2 Score	Final Score
Model 1	74	85	159
Model 2	81	86	167
Model 3	71	78	149
Model 4	79	84	163

From Table 5.4, it can be seen that Model 3 has obtained the lowest score in each of the four forms and hence has the lowest final score. This shows that Model 3 performs consistently and is the best Model compared to the other three. Model 3 corresponds to the model where the query was translated using the LT-NN model and no Re-ranking of documents is performed. Since Model 3 has performed well than Model 4, it can be concluded that LT-NN performs well when no Re-ranking is applied.

5.3 Summary

The results and analysis of the models are given in this Chapter. It starts by showing the results of the Translation Models and selecting the best translation model. Next, the evaluation process is discussed which explains the query collection process and procedure, results and analysis of the two user-based evaluation process.

Chapter 6 - Conclusions

6.1 Conclusions about the Research Questions (Aims and Objectives)

The aim of this research was to provide a way for people who are not fluent in English to be able to search the web in Sinhala. This involves finding the best equivalent English query for a given Sinhala query. Hence, this study included checking the performance of various translation models in order to find the best performing model to build a system that could solve the problem of obtaining relevant documents when browsing the web in Sinhala.

The translation model developed using the Linear Transformation model suggested in [11] with the Standard Nearest Neighbour Retrieval method performed the best when evaluated on a test set of 1500 Sinhala words along with their translations in English obtained through Google Translate. This had obtained a Precision at 1 score of 35%. Hence this model could be used to translate a Sinhala query, word by word to convert to an approximately equivalent English query which could provide the correct translations of the main keywords in the Sinhala query which is the vital component in retrieving relevant documents. Hence, this answers the first research question of "How to convert the Sinhala search query into the equivalent English one?". This model is a basic model which has could be improved further.

This model has outperformed the Linear Translation with the Globally Corrected Neighbour Retrieval method suggested by [39] as well as the Orthogonal Transformation method suggested by [12]. [39] shows that the Globally Corrected Neighbour Retrieval method outperforms the Standard Nearest Neighbour Retrieval method for English to Italian Translation while [12] shows that the Orthogonal Translations employing SVD outperforms both these methods for English to Italian Translation. The results obtained through this study do not align with these results and hence it can be shown that Sinhala to English Translation behaves in a different way compared to the English to Italian Translation. Since there has not been any study that

includes a Sinhala to English Translation model built employing the concept of Word Embeddings, the best model in this study cannot be compared to check whether it aligns with other studies as well.

The second research question focuses on determining the effect of re-ranking the translated documents. Results of Evaluation 1 shows that documents obtained after re-ranking them using an LSI based model give good results compared to the basic model but the documents obtained from the queries translated using Google Translate without applying re-ranking are shown to be the most relevant. Evaluation 2 also shows that the LT-NN translation model performs well without re-ranking. Hence, it can be concluded that the effect of re-ranking the translated documents do not show a positive impact and the final system should present the documents to the user in the order they are obtained using the Google Search API.

6.2 Conclusions about the Research Problem

This study focuses on determining the best method to solve the problem of browsing the web in Sinhala. Two user-based evaluations were conducted and the LT-NN model without Re-ranking performed well for a limited set of queries. But due to some limitations of the LT-NN model developed which will be discussed in Section 6.3, this model is not optimized to work in a production environment. Hence, this model should be optimized and also improved as described in the Future work section in order to be an effective model that could be used to build a working system.

The word embedding model created in this study for the Sinhala language has been shown to perform well as discussed in Section 4. This embedding model has been used to create a translation model as well which performs decently which also shows the performance of the Sinhala word embedding model. Hence, the Sinhala word embedding model created here can be shown as a contribution to the research community which could be utilized to perform various other NLP tasks.

This study has compared several translation models that perform Sinhala to English translation. Hence, subsequent studies can refer to this study and check whether their

finding matches this study or could use this as a guide for experimenting with various other models which could be used to train a Sinhala to English Translation model. This study also provides a guide on two re-ranking models that are applied to Sinhala documents and provides the effects of those models.

6.3 Limitations

The translation models created in this study takes about 30 seconds to give the output for a particular word. As a result, translating a query is time-consuming and not suitable to be implemented in a system. But these models can be optimized and hence necessary methods should be considered for optimizing these models.

The Google Search API provides only 100 results for a particular query and allows up to 100 queries per day. In the evaluation process, only 50 documents were considered due to this query limit but considering more documents might have an impact on the re-ranking process. The results obtained do not include the content of the documents. Hence, the content should be retrieved by scraping the web page using the URL but it would be difficult to extract the content alone since different web pages have different structures. This also has an impact when translating the documents back to Sinhala since the document to be translated would not contain the content alone but some other unnecessary texts as well which would affect the Re-ranking process too.

6.4 Implications for Further Research

The translation models created in this study performs a word by word translation when given a query. This procedure can be extended to handle phrase translations in order to obtain more accurate answers. Also, there can be words whose translations might not be the first result given by the translation model but it may be the second or third result. In such cases, an English Language model could be trained and used to identify the correct translation. This study has focused on two projection mechanisms as well as two translation retrieval mechanisms. Other available mechanisms also could be tried out to determine their performance.

A basic re-rank model and an LSI-based re-ranking model have been used in this research. This could be extended to employ several other re-ranking models such as LDA based re-ranking.

References

[1] “Google Search Statistics - Internet Live Stats.” https://www.internetlivestats.com/google-search-statistics/ (accessed Feb. 18, 2020).

[2] “How many languages are there in the world? | Ethnologue.” https://www.ethnologue.com/guides/how-many-languages (accessed Feb. 18, 2020).

[3] M. K. Chinnakotla, S. Ranadive, P. Bhattacharyya, and O. P. Damani, “Hindi and Marathi to English cross language information retrieval at CLEF 2007,” *CEUR Workshop Proc.*, vol. 1173, 2007.

[4] D. A. Hull and G. Grefenstette, “Querying across languages,” no. May, pp. 49–57, 1996, doi: 10.1145/243199.243212.

[5] L. Ballesteros and B. Croft, “Dictionary methods for cross-lingual information retrieval,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 1134 LNCS, pp. 791–801, 1996, doi: 10.1007/bfb0034731.

[6] J. Jagarlamudi and A. Kumaran, “Cross-lingual information retrieval system for Indian languages,” *CEUR Workshop Proc.*, vol. 1173, 2007, doi: 10.1007/978-3-540-85760-0_10.

[7] F. Ture, J. Lin, and D. W. Oard, “Combining statistical translation techniques for cross-language information retrieval,” *24th Int. Conf. Comput. Linguist. - Proc. COLING 2012 Tech. Pap.*, vol. 3, no. December 2012, pp. 2685–2702, 2012.

[8] F. Ture, J. Lin, and D. W. Oard, “Looking inside the box: Context-sensitive translation for cross-language information retrieval,” *SIGIR’12 - Proc. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, no. 5, pp. 1105–1106, 2012, doi: 10.1145/2348283.2348491.

[9] R. Prasath, S. Sarkar, and P. O’Reilly, “Improving Cross Language Information Retrieval Using Corpus Based Query Suggestion Approach,” 2015.

[10] F. C. Gey, N. Kando, and C. Peters, “Cross-language information retrieval: The way ahead,” *Inf. Process. Manag.*, vol. 41, no. 3, pp. 415–431, 2005, doi: 10.1016/j.ipm.2004.06.006.

[11] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting Similarities among

Languages for Machine Translation," 2013, [Online]. Available: http://arxiv.org/abs/1309.4168.

[12] S. L. Smith, D. H. P. Turban, S. Hamblin, and N. Y. Hammerla, "Offline bilingual word vectors, orthogonal transformations and the inverted softmax," *5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc.*, pp. 1–10, 2019.

[13] A. H. Vahid, P. Arora, Q. Liu, and G. J. F. Jones, "A comparative study of online translation services for cross language information retrieval," *WWW 2015 Companion - Proc. 24th Int. Conf. World Wide Web*, no. May, pp. 859–864, 2015, doi: 10.1145/2740908.2743008.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global Vectors for Word Representation," 2014.

[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching Word Vectors with Subword Information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.

[17] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *15th Conf. Eur. Chapter Assoc. Comput. Linguist. EACL 2017 - Proc. Conf.*, vol. 2, pp. 427–431, 2017, doi: 10.18653/v1/e17-2068.

[18] "Word vectors for 157 languages · fastText." https://fasttext.cc/docs/en/crawl-vectors.html (accessed Feb. 19, 2020).

[19] P. Bhattacharya, P. Goyal, and S. Sarkar, "Using Word Embeddings for Query Translation for Hindi to English Cross Language Information Retrieval," *Comput. y Sist.*, vol. 20, no. 3, pp. 435–447, 2016, doi: 10.13053/CyS-20-3-2462.

[20] I. Vulić and M. F. Moens, "Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings," *SIGIR 2015 - Proc. 38th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 363–372, 2015, doi: 10.1145/2766462.2767752.

[21] M. Baroni, G. Dinu, and G. Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," *52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf.*, vol. 1, pp. 238–247, 2014, doi: 10.3115/v1/p14-1023.

[22] O. Levy, Y. Goldberg, and I. Dagan, “Improving Distributional Similarity with Lessons Learned from Word Embeddings,” *Trans. Assoc. Comput. Linguist.*, vol. 3, pp. 211–225, 2015, doi: 10.1162/tacl_a_00134.

[23] T. Mikolov, W. Yih, and G. Zweig, “Linguistic Regularities in Continuous Space Word Representations,” 2013.

[24] J. Turian, L. Ratinov, and Y. Bengio, “Word representations: A simple and general method for semi-supervised learning,” *ACL 2010 - 48th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, no. July, pp. 384–394, 2010.

[25] R. Collobert and J. Weston, “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning,” in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 160–167, doi: 10.1145/1390156.1390177.

[26] S. Ruder, I. Vulić, and A. Søgaard, “A Survey of Cross-lingual Word Embedding Models,” *J. Artif. Intell. Res.*, vol. 65, pp. 569–631, 2019, doi: 10.1613/jair.1.11640.

[27] C. Xing, D. Wang, C. Liu, and Y. Lin, “Normalized word embedding and orthogonal transform for bilingual word translation,” *NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf.*, pp. 1006–1011, 2015, doi: 10.3115/v1/n15-1104.

[28] M. Artetxe, G. Labaka, and E. Agirre, “Learning principled bilingual mappings of word embeddings while preserving monolingual invariance,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 2289–2294, 2016, doi: 10.18653/v1/d16-1250.

[29] P. Bhattacharya, P. Goyal, and S. Sarkar, “Query Translation for Cross-Language Information Retrieval using Multilingual Word Clusters,” *Proc. 6th Work. South Southeast {A}sian Nat. Lang. Process.*, pp. 152–162, 2016, [Online]. Available: https://www.aclweb.org/anthology/W16-3716.

[30] N. Duhan, A. K. Sharma, and K. K. Bhatia, “Page ranking algorithms: A survey,” *2009 IEEE Int. Adv. Comput. Conf. IACC 2009*, no. March, pp. 1530–1537, 2009, doi: 10.1109/IADCC.2009.4809246.

[31] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web.,” 1999.

[32] W. Xing and A. Ghorbani, “Weighted PageRank Algorithm,” 2004, pp. 305–314,

doi: 10.1109/DNSR.2004.1344743.

[33] J. Pokorný and J. Smizanský, “Page content rank: an approach to the web content mining,” 2005.

[34] J. M. Kleinberg, “Authoritative Sources in a Hyperlinked Environment,” *J. ACM*, vol. 46, no. 5, pp. 604–632, Sep. 1999, doi: 10.1145/324133.324140.

[35] C. Ding, H. Zha, X. He, and H. Simon, “Link Analysis: Hubs and Authorities on the World Wide Web,” *SIAM Rev.*, vol. 46, 2002, doi: 10.1137/S0036144501389218.

[36] S. Brin and L. Page, “The anatomy of a large-scale hypertextual Web search engine BT - Computer Networks and ISDN Systems,” *Comput. Networks ISDN Syst.*, vol. 30, no. 1–7, pp. 107–117, 1998, doi: 10.1016/S0169-7552(98)00110-X.

[37] “Parallel Corpus Filtering for Low-Resource Conditions Task - ACL 2019 Fourth Conference on Machine Translation.” http://www.statmt.org/wmt19/parallel-corpus-filtering.html (accessed Feb. 19, 2020).

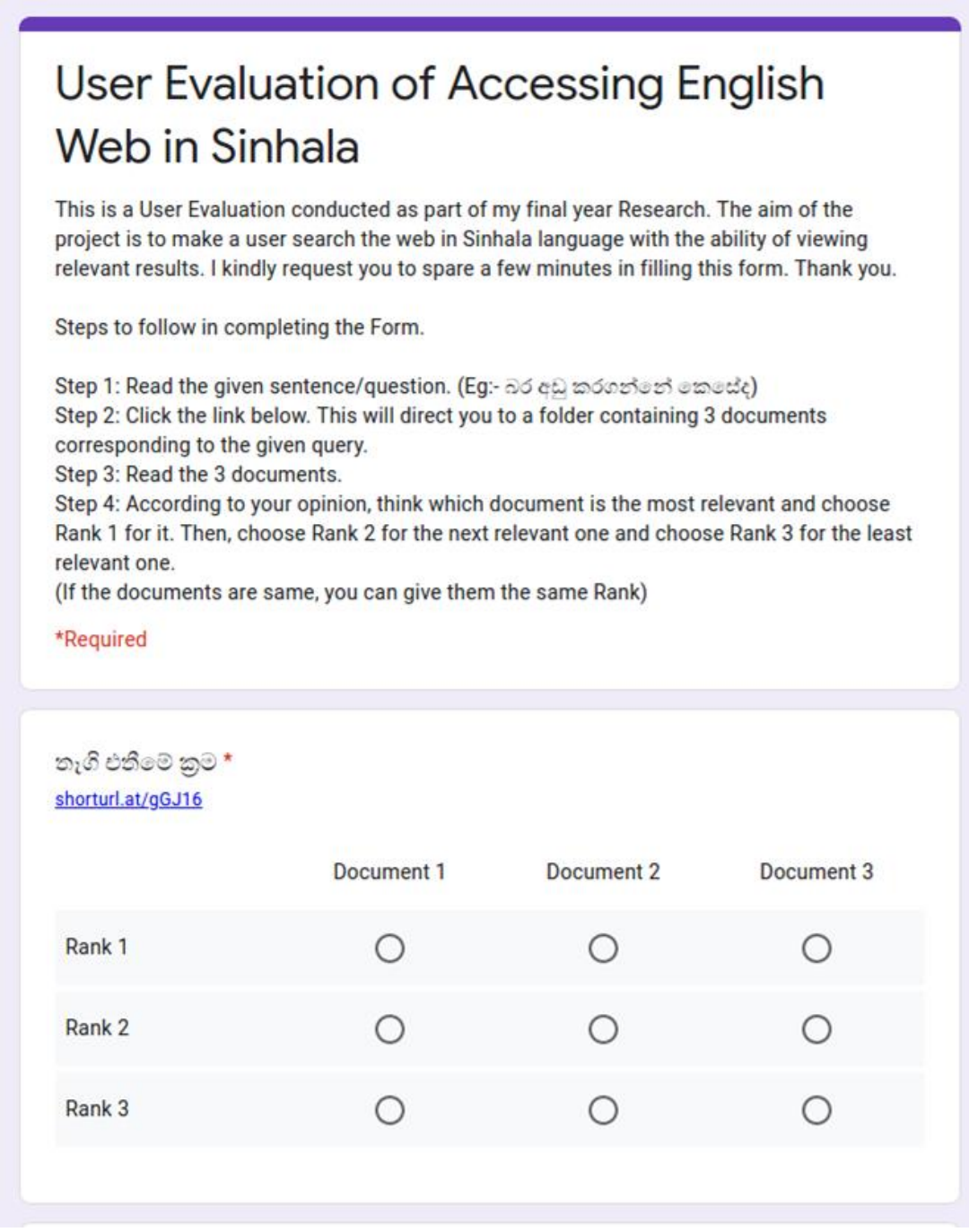
[38] “lzakharov/lsi: Latent Semantic Indexing.” https://github.com/lzakharov/lsi (accessed Feb. 19, 2020).

[39] G. Dinu, A. Lazaridou, and M. Baroni, “Improving zero-shot learning by mitigating the hubness problem,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Work. Track Proc.*, pp. 1–10, 2015.

Appendix A: Evaluation 1

A.1 Questionnaire 1 and Results

Following is the Google Form that was used as Questionnaire 1.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same Rank)

*Required

නැගී එනිමේ ක්‍රම *
shorturl.at/gGJ16

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

සම්මුඛ පරීක්ෂණයකට මුහුණ දෙන්නේ කෙසේද *
shorturl.at/ckwQ4

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

සෞඛ්‍ය සම්පන්න ආහාර *
shorturl.at/qFSX2

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ලෝකයේ හොඳම රැකියා *
shorturl.at/gVY02

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

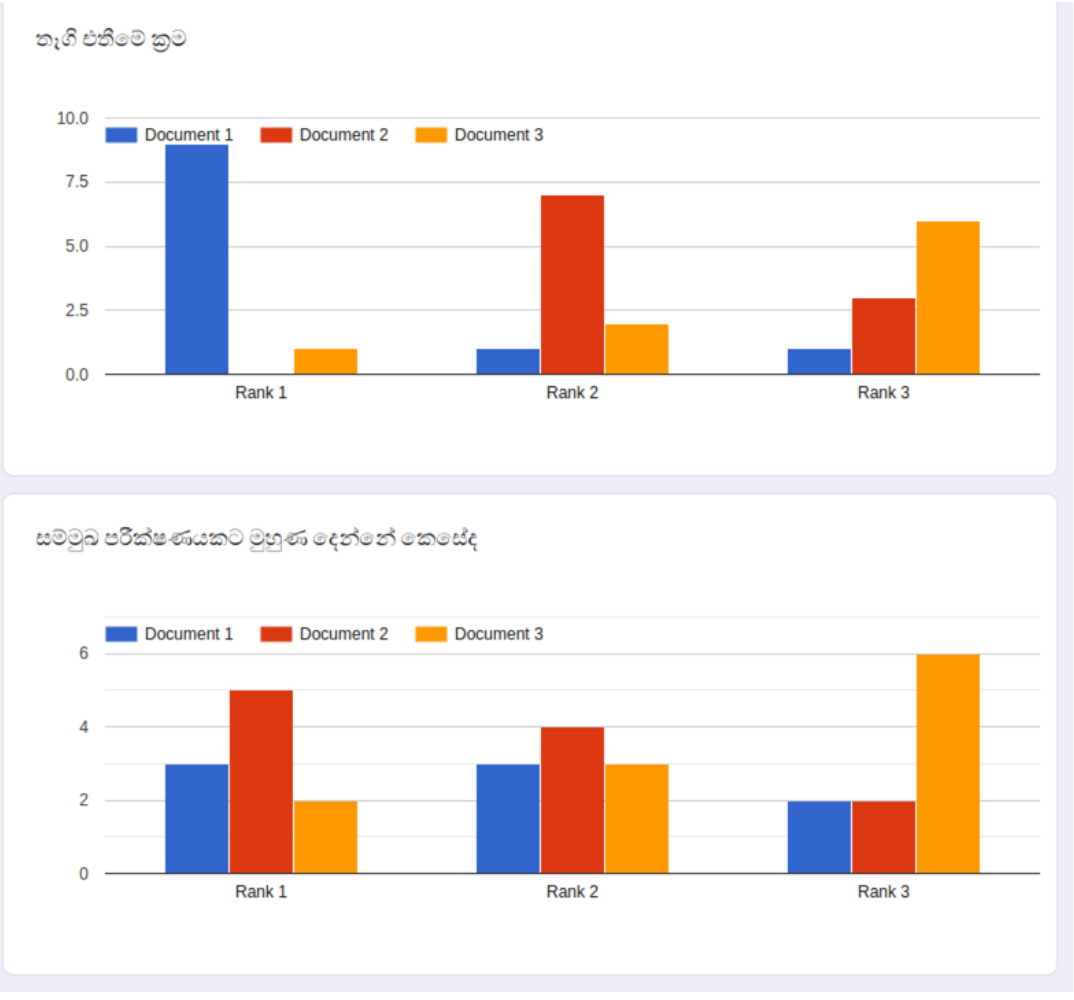
රචනයක් ලියන ආකාරය *

shorturl.at/pBGH9

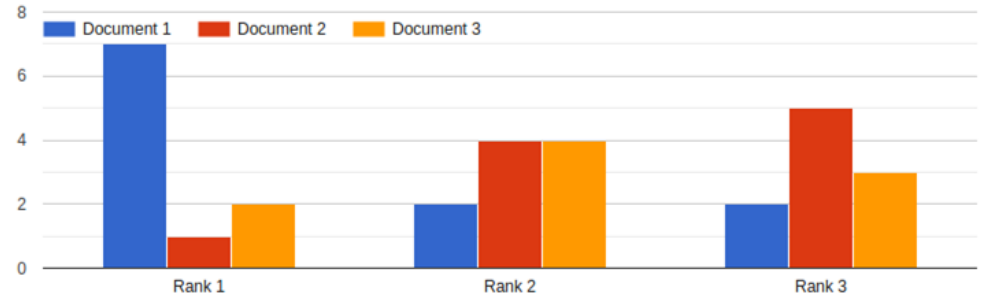
	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

Submit

Following are the results of the Questionnaire 1.



සෞඛ්‍ය සම්පන්න ආහාර

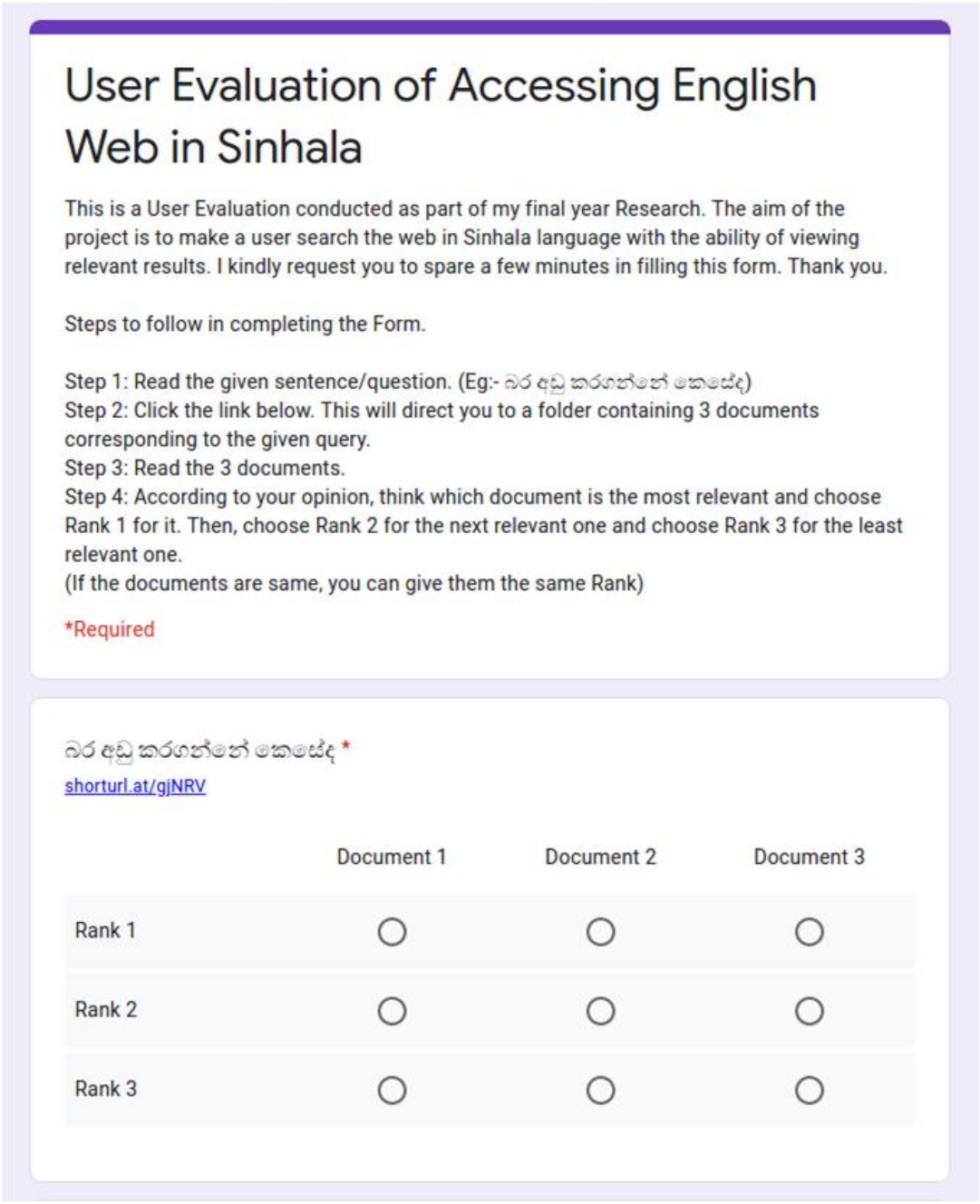


ලෝකයේ හොඳම රැකියා

රචනයක් ලියන ආකාරය

A.2 Questionnaire 2 and Results

Following is the Google Form that was used as Questionnaire 2.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same Rank)

*Required

බර අඩු කරගන්නේ කෙසේද *

shorturl.at/gjNRV

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ඔබෙ කම්මැලි විට කුමක් කළ යුතුද *
shorturl.at/blKOS

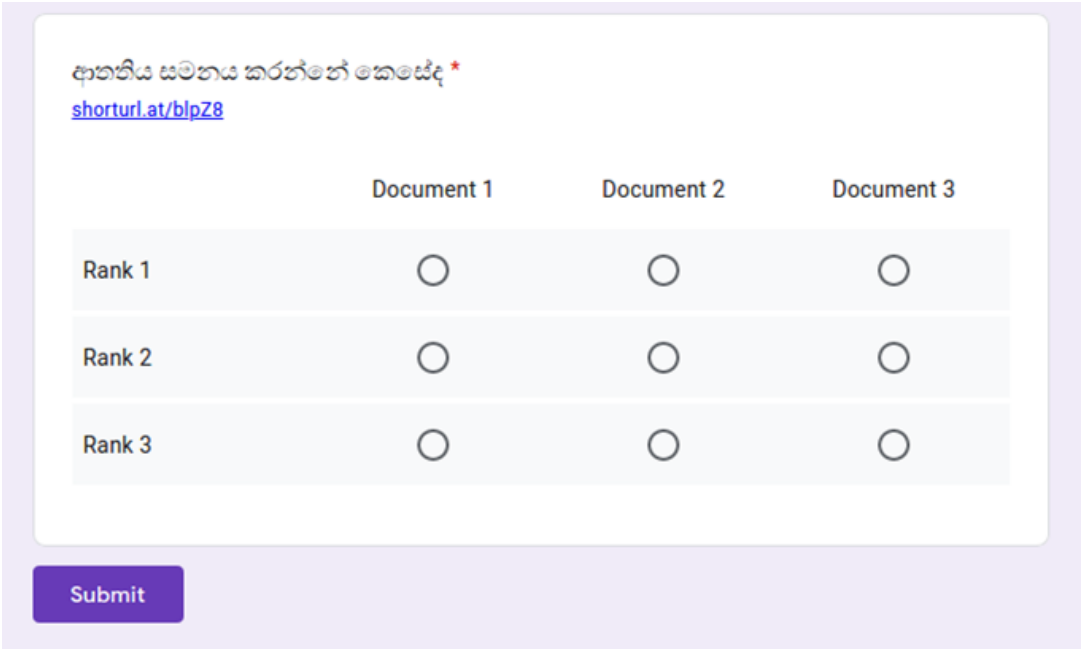
	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ඉංග්‍රීසි ඉගෙන ගන්නේ කොහොමද *
shorturl.at/cknuC

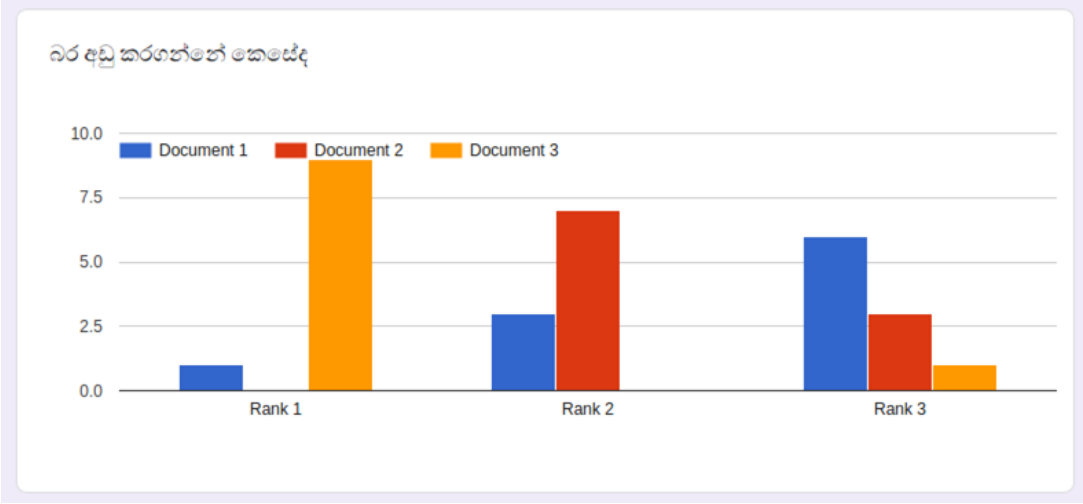
	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

මිතුරා සඳහා නැගි අදහස් *
shorturl.at/imxC1

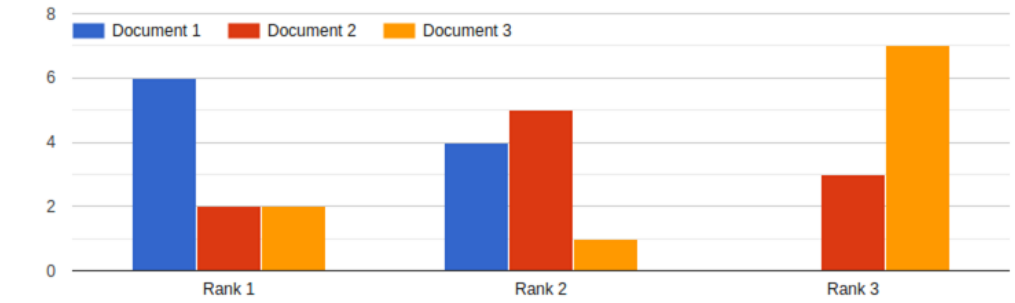
	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○



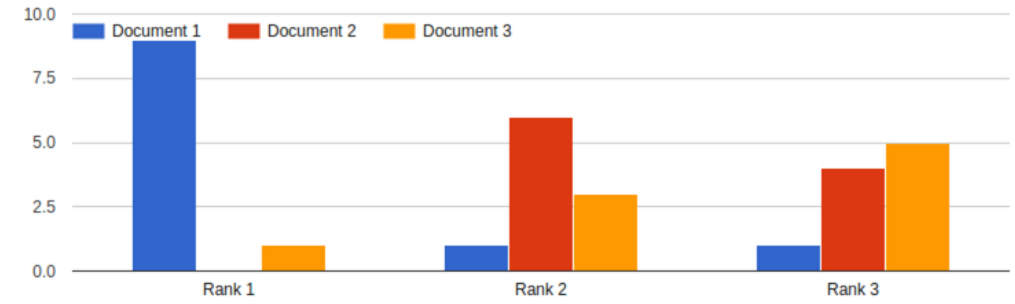
Following are the results of the Questionnaire 2.



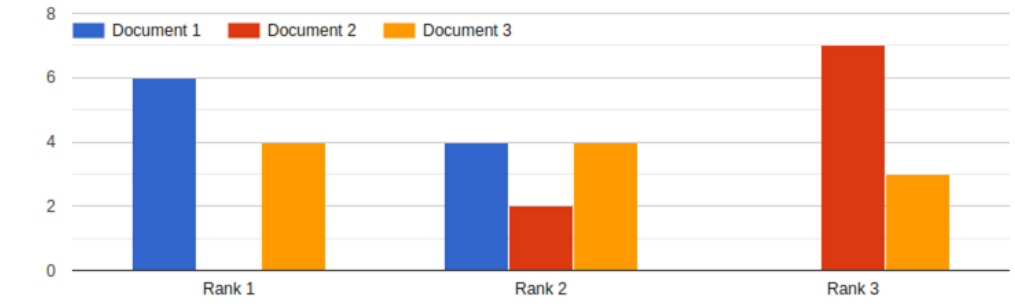
ඔබේ කම්මැලි විට කුමක් කළ යුතුද

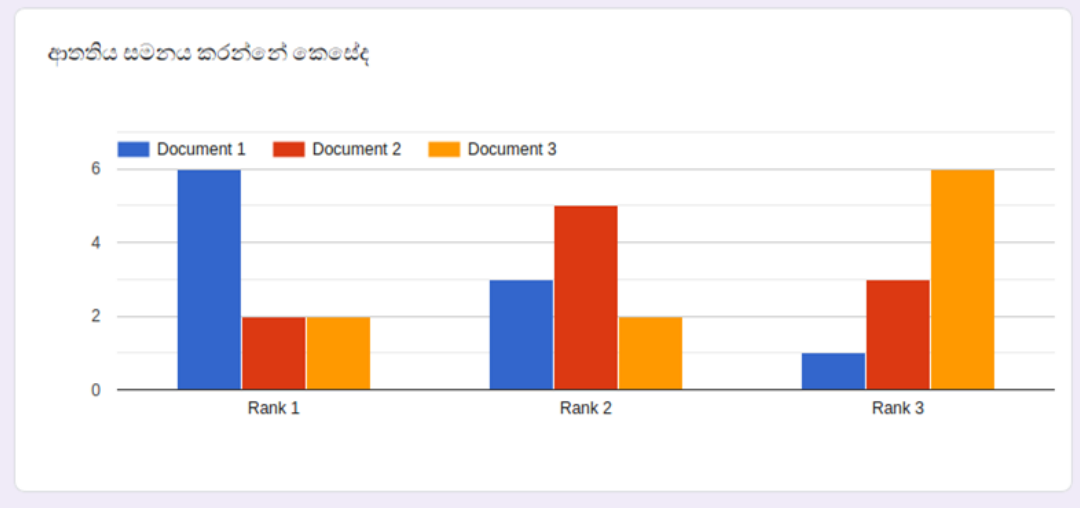


ඉංග්‍රීසි ඉගෙන ගන්නේ කොහොමද



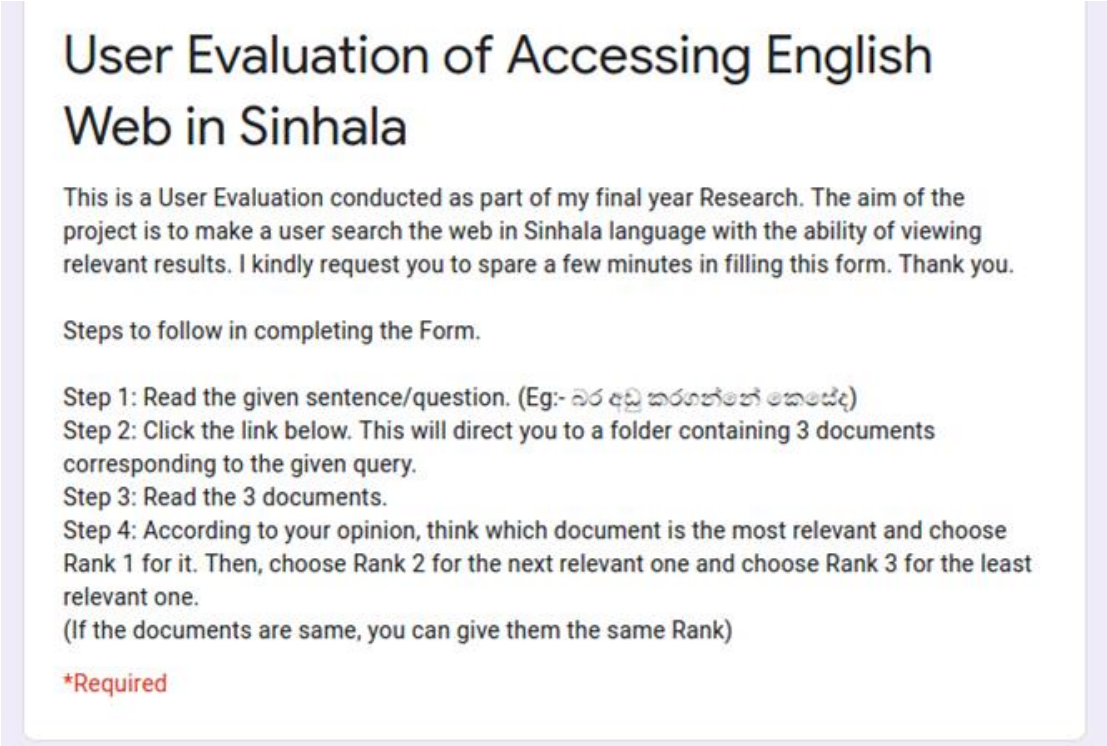
මිතුරා සඳහා තැගි අදහස්





A.3 Questionnaire 3 and Results

Following is the Google Form that was used as Questionnaire 3.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same Rank)

*Required

ලෝකය වෙනස් කළ විශිෂ්ට පුද්ගලයන් *

shorturl.at/exB47

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ක්‍රිකට් තරහ ප්‍රතිපල *

shorturl.at/jlAGJ

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

සමීකරණයක මූලයන් සොයා ගන්නේ කෙසේද *

shorturl.at/aeijQ

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

සෞරග්‍රහ මණ්ඩලය *

shorturl.at/arD67

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ලෝකයේ ධනවතුන් *

shorturl.at/sK067

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

Submit

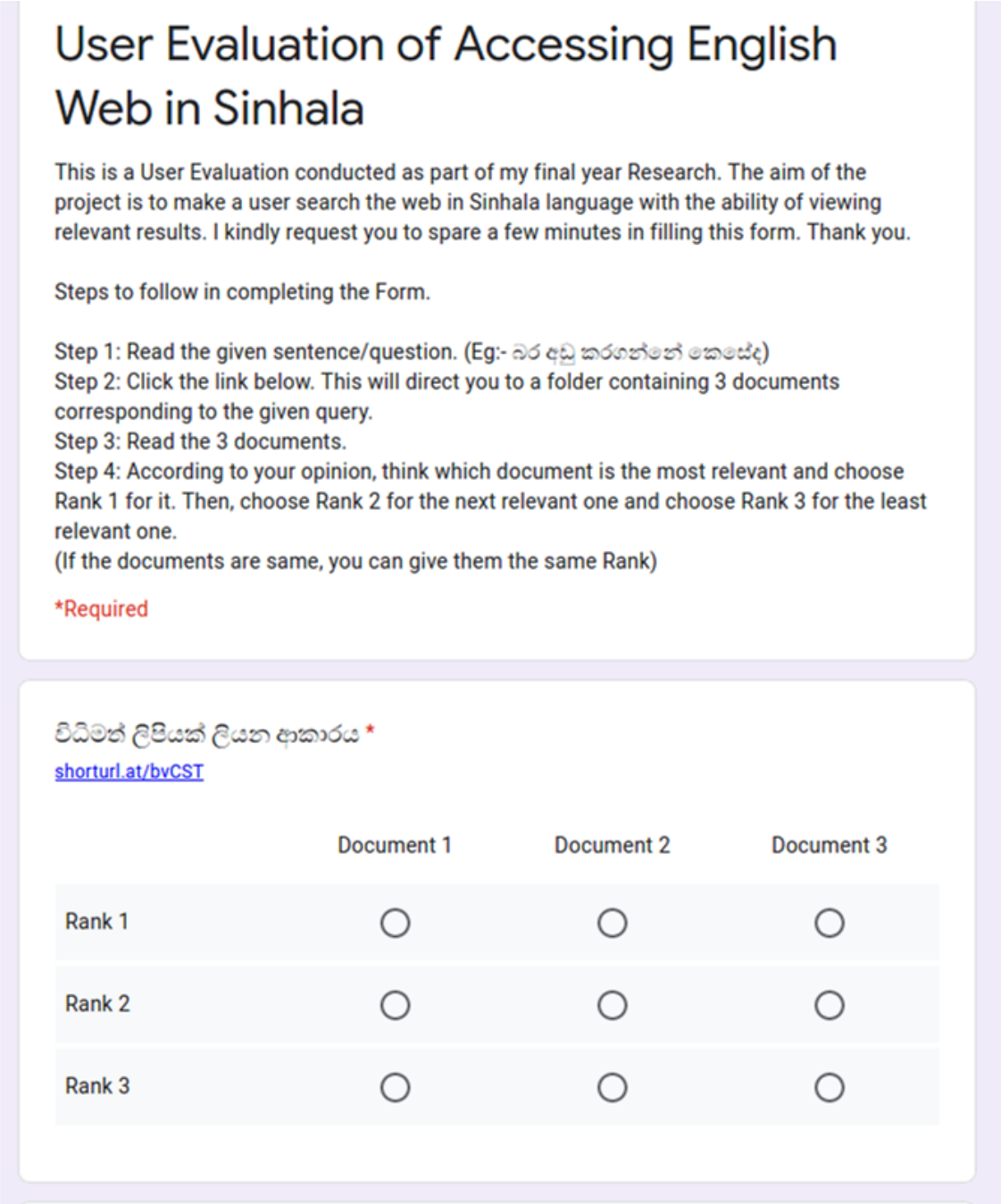
Following are the results of the Questionnaire 3.

සෞරග්‍රහ මණ්ඩලය

ලෝකයේ ධනවතුන්

A.4 Questionnaire 4 and Results

Following is the Google Form that was used as Questionnaire 4.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same Rank)

*Required

විධිමත් ලිපියක් ලියන ආකාරය *
shorturl.at/bvCST

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

අන්තර්ජාලය හරහා මුදල් උපයන්නේ කෙසේද *
shorturl.at/sFJ23

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

හිසකෙස් වැටීම වැළැක්වීමට උපදෙස් *
shorturl.at/jkqDH

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

ලෝකයේ හොඳම විශ්ව විද්‍යාල *
shorturl.at/iJ456

	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

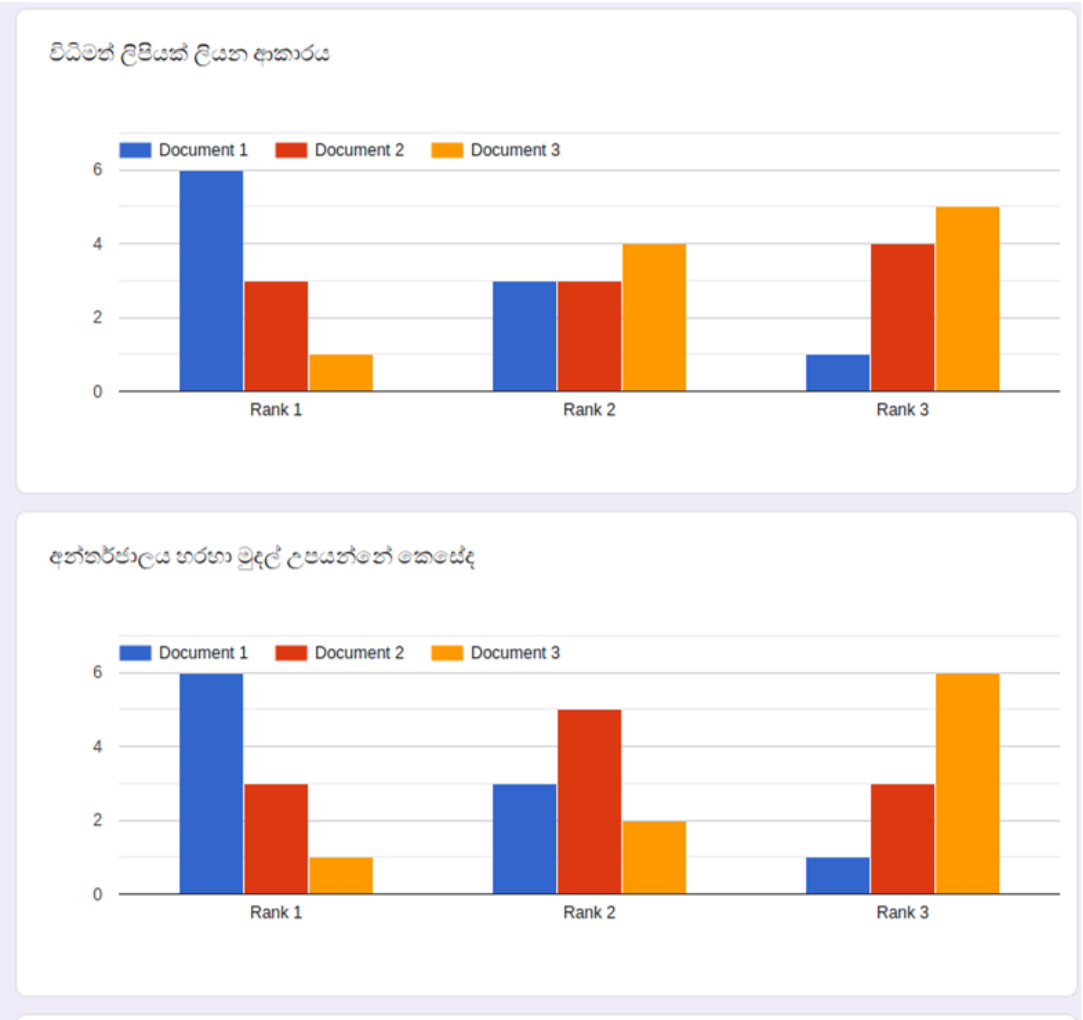
මිතුරෙකුට අලුත් අවුරුදු සුබ පැතුම් *

shorturl.at/fgJSY

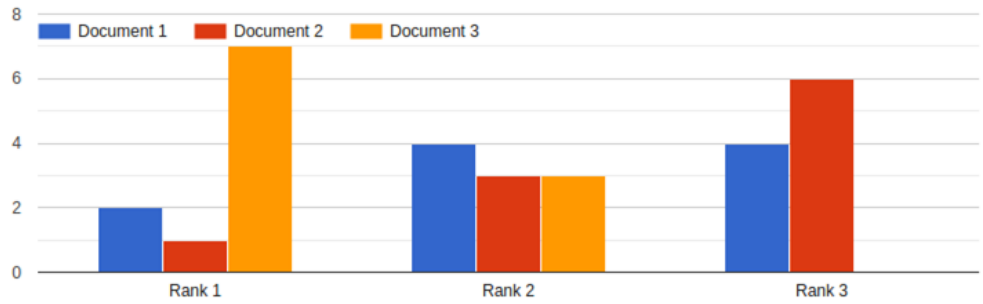
	Document 1	Document 2	Document 3
Rank 1	○	○	○
Rank 2	○	○	○
Rank 3	○	○	○

Submit

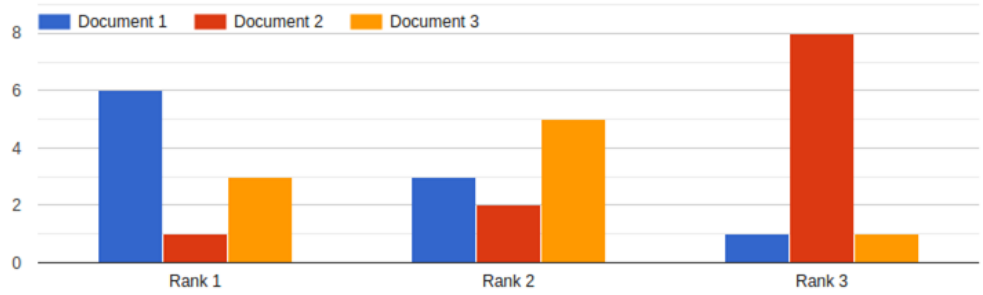
Following are the results of the Questionnaire 4.



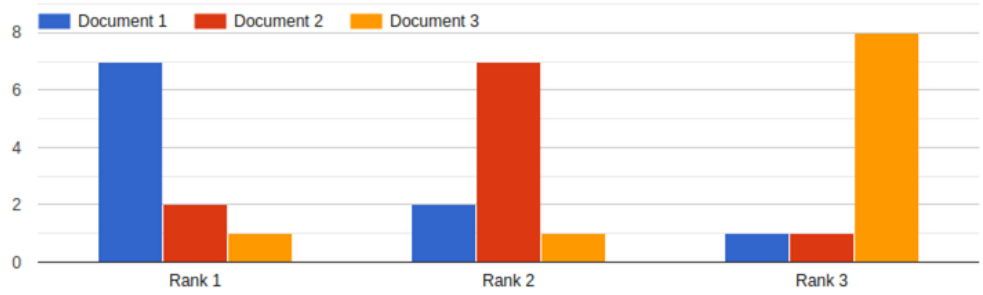
හිසකෙස් වැටීම වැළැක්වීමට උපදෙස්



ලෝකයේ හොඳම විශ්ව විද්‍යාල



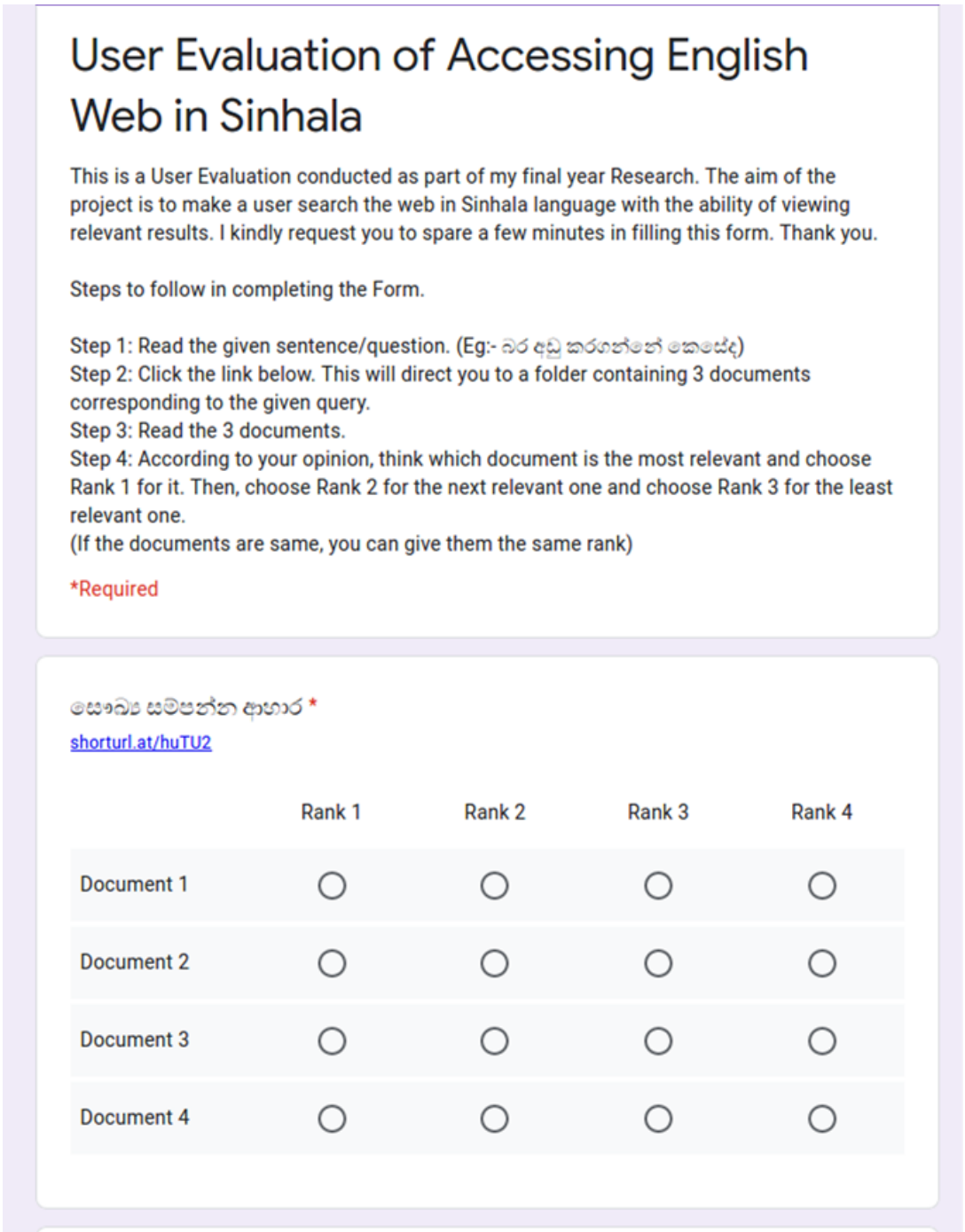
මිතුරෙකුට අලුත් අවුරුදු සුබ පැතුම්



Appendix B: Evaluation 2

B.1 Questionnaire 1 and Results

Following is the Google Form that was used as Questionnaire 1.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same rank)

*Required

සෞඛ්‍ය සම්පන්න ආහාර *

shorturl.at/huTU2

	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

ලෝකයේ හොඳම රැකියා *

shorturl.at/uzBGS

	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

ක්‍රිකට් තරඟ ප්‍රතිඵල *

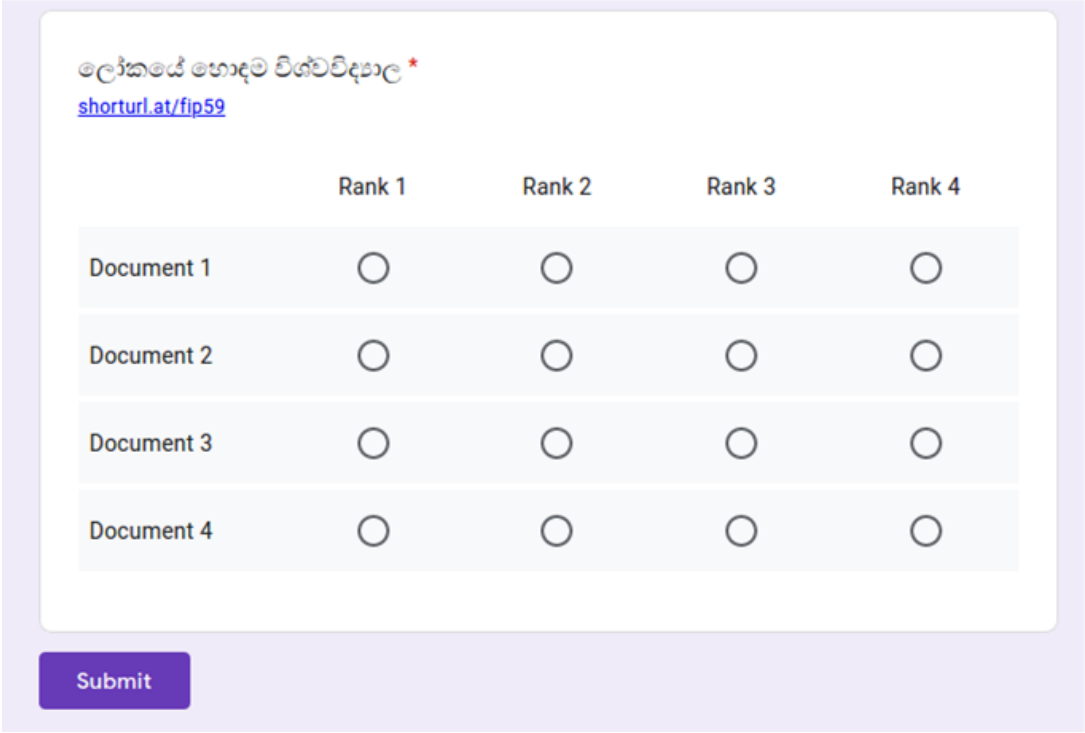
shorturl.at/nqOW5

	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

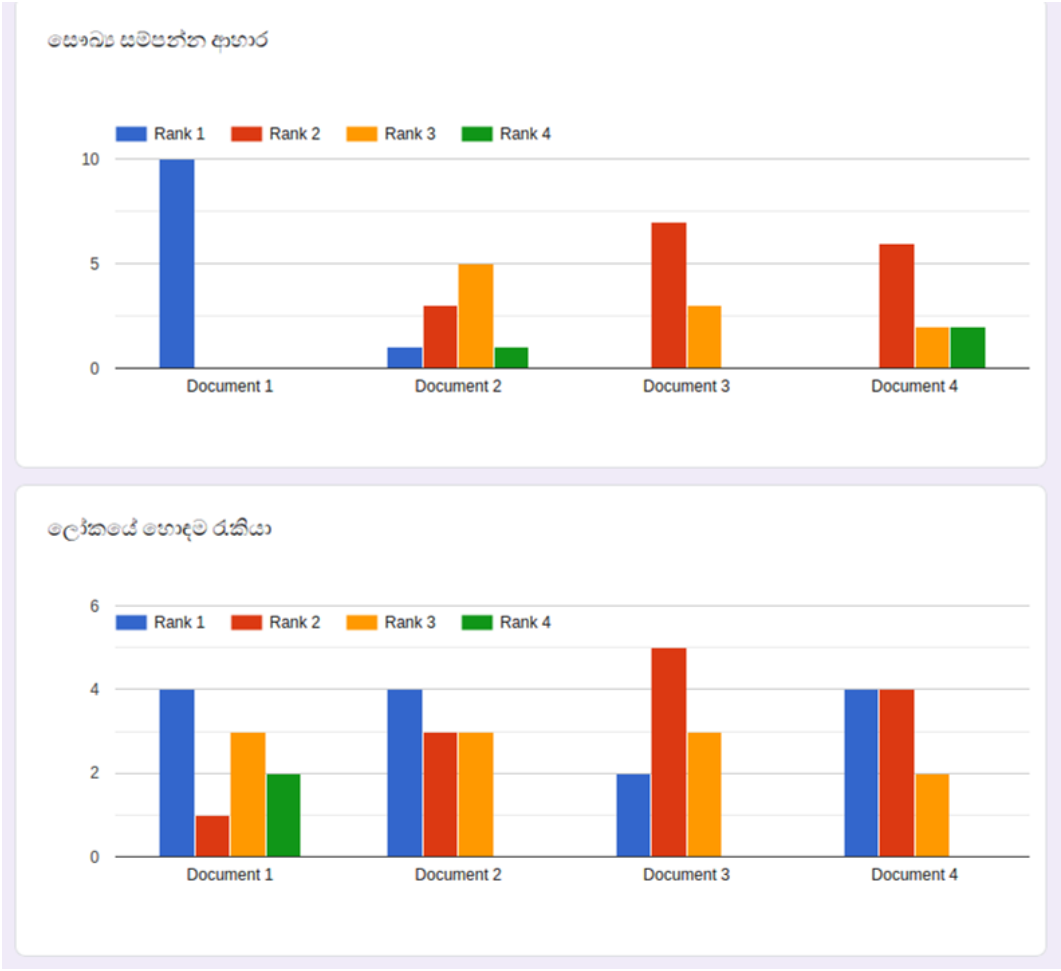
විධිමත් ලියුමක් ලියන ආකාරය *

shorturl.at/cdwDL

	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○



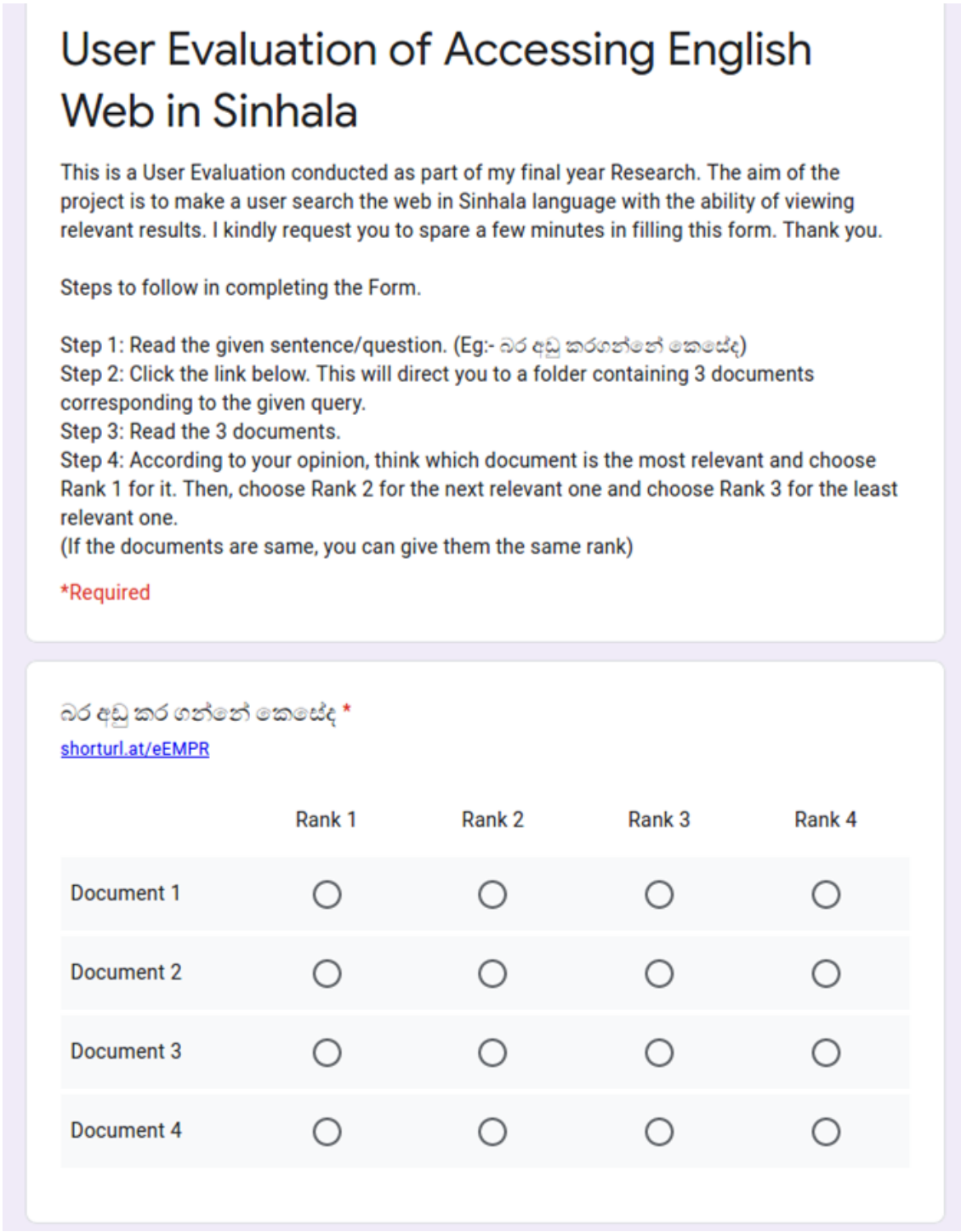
Following are the results of the Questionnaire 1.




ක්‍රිකට් තරඟ ප්‍රතිඵල
10.0
7.5
5.0
2.5
0.0
Rank 1
Rank 2
Rank 3
Rank 4
Document 1
Document 2
Document 3
Document 4
විඩියෝ ලියුමක් ලියන ආකාරය
8
6
4
2
0
Rank 1
Rank 2
Rank 3
Rank 4
Document 1
Document 2
Document 3
Document 4
ලෝකයේ හොඳම විශ්වවිද්‍යාල
8
6
4
2
0
Rank 1
Rank 2
Rank 3
Rank 4
Document 1
Document 2
Document 3
Document 4

B.2 Questionnaire 2 and Results

Following is the Google Form that was used as Questionnaire 2.



User Evaluation of Accessing English Web in Sinhala

This is a User Evaluation conducted as part of my final year Research. The aim of the project is to make a user search the web in Sinhala language with the ability of viewing relevant results. I kindly request you to spare a few minutes in filling this form. Thank you.

Steps to follow in completing the Form.

Step 1: Read the given sentence/question. (Eg:- බර අඩු කරගන්නේ කෙසේද)
Step 2: Click the link below. This will direct you to a folder containing 3 documents corresponding to the given query.
Step 3: Read the 3 documents.
Step 4: According to your opinion, think which document is the most relevant and choose Rank 1 for it. Then, choose Rank 2 for the next relevant one and choose Rank 3 for the least relevant one.
(If the documents are same, you can give them the same rank)

*Required

බර අඩු කර ගන්නේ කෙසේද *

shorturl.at/eEMPR

	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

ඉංග්‍රීසි ඉගෙනගත හැකි ක්‍රම *
shorturl.at/fP148

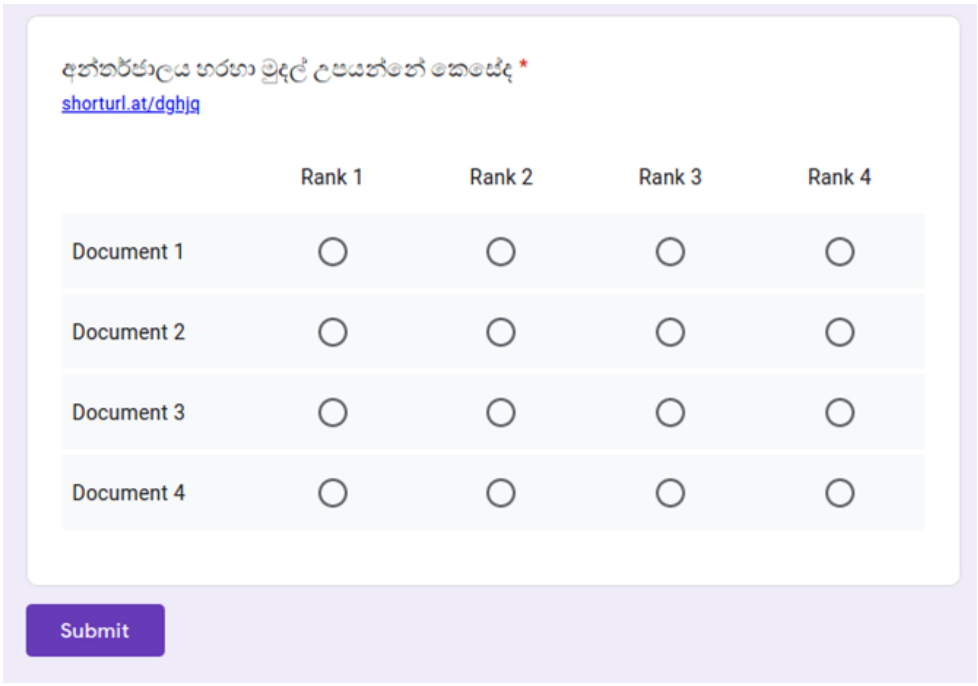
	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

මිතුරා සඳහා තෑගි අදහස් *
shorturl.at/gtuv7

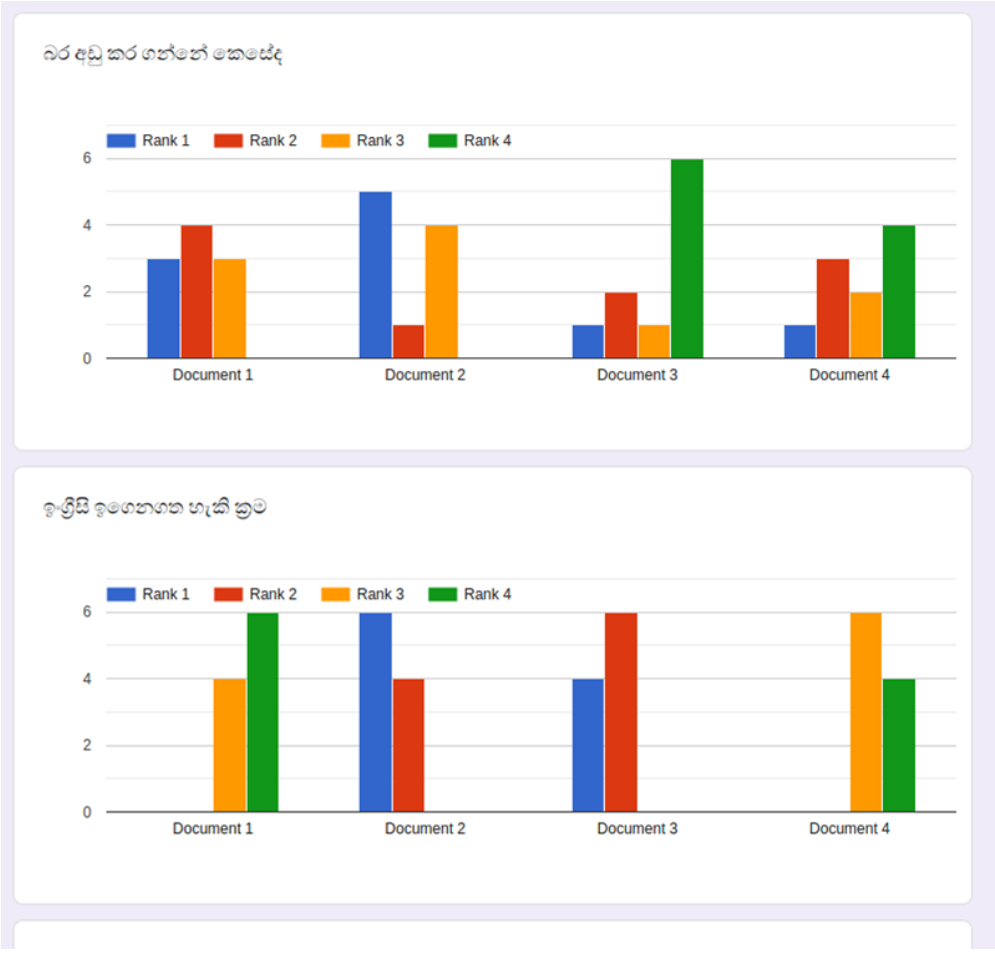
	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○

ආනතිය සමනය කරන්නේ කෙසේද *
shorturl.at/svN24

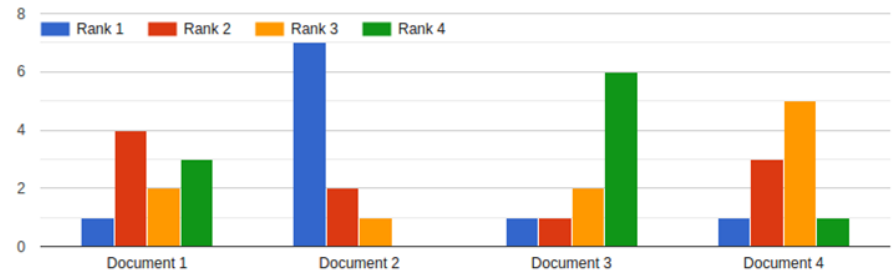
	Rank 1	Rank 2	Rank 3	Rank 4
Document 1	○	○	○	○
Document 2	○	○	○	○
Document 3	○	○	○	○
Document 4	○	○	○	○



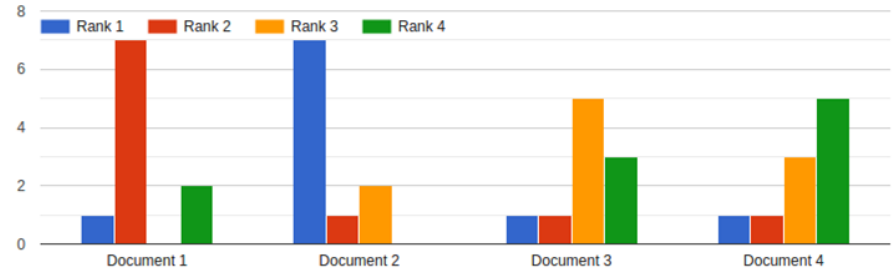
Following are the results of the Questionnaire 2.



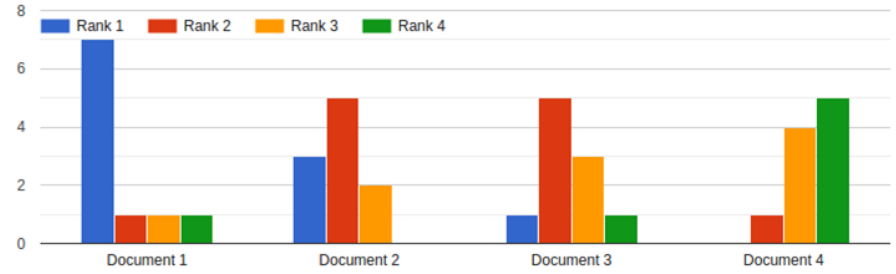
මිතුරා සඳහා තෑගි අදහස්



ආනතිය සමනය කරන්නේ කෙසේද



අන්තර්ජාලය හරහා මුදල් උපයන්නේ කෙසේද



Appendix C: Sinhala Queries

1. බර අඩු කර ගන්නේ කෙසේද
2. ඔබේ කම්මැලි විට කුමක් කළ යුතුද
3. ඉංග්‍රීසි ඉගෙන ගන්නේ කොහොමද
4. මිතුරා සඳහා තෑගි අදහස්
5. ආතතිය සමනය කරන්නේ කෙසේද
6. තෑගි එතීමේ ක්‍රම
7. සම්මුඛ පරීක්ෂණයකට මුහුණ දෙන්නේ කෙසේද
8. සෞඛ්‍ය සම්පන්න ආහාර
9. ලෝකයේ හොඳම රැකියා
10. රචනයක් ලියන ආකාරය
11. ලෝකය වෙනස් කළ විශිෂ්ට පුද්ගලයන්
12. ක්‍රිකට් තරඟ ප්‍රතිඵල
13. සමීකරණයක මූලයන් සොයා ගන්නේ කෙසේද
14. සෞරග්‍රහ මණ්ඩලය
15. ලෝකයේ ධනවතුන්
16. විධිමත් ලිපියක් ලියන ආකාරය
17. අන්තර්ජාලය හරහා මුදල් උපයන්නේ කෙසේද
18. හිසකෙස් වැටීම වැළැක්වීමට උපදෙස්
19. ලෝකයේ හොඳම විශ්ව විද්‍යාල
20. මිතුරෙකුට අලුත් අවුරුදු සුබ පැතුම්

Appendix D: Code Listings

D.1 Code Listings of Data Pre-Processing

```
import re

path = '/content/drive/My Drive/Data Sets/commoncrawl.deduped.si'

with open(path, 'r') as f:
  content = f.read()

input_text = re.sub(r'\([^)]*\)', '', content)

sentences_strings = []
for line in input_text.split('\n'):
    m = re.match(r'^(?:(?P<precolon>[^:]{,20}):)?(?P<postcolon>.*)
$', line)
    sentences_strings.extend(sent for sent in m.groupdict()['postc
olon'].split('.') if sent)
# store as list of lists of words
sentences = []
for sent_str in sentences_strings:
    tokens = re.sub(r"[0-9]+", " ", sent_str.lower()).split()
    sentences.append(tokens)


import re

non = re.compile('[\u0D80-\u0DFF]').search
non2 = re.compile(r'[^\u0D80-\u0DFF\u200d]').search

sentence_list2 = []

for sentence in sentences:
  word_list = []

  for word in sentence:
    word_copy = word

    for i in word:
      if(bool(non(i))):
        break
      else:
        word_copy2 = word_copy.replace(i,"",1)
        word_copy = word_copy2
```

```
    word_copy3 = word_copy

    for j in word_copy3[::-1]:
      if(bool(non(j))):
        break
      else:
        word_copy2 = word_copy[::-1].replace(j,"",1)
        word_copy = word_copy2[::-1]

    if(word_copy != ''):
      if(not bool(non2(word_copy))):
        word_list.append(word_copy)

  sentence_list2.append(word_list)
```

D.2 Code Listings of Training Dictionary Generation

```
si_words1 = []
en_words1 = []
with open("/content/drive/My Drive/Dictionary/traindict.txt", "r")
 as f:
  for item in f:
    words = item.split()
    if(len(words) == 0 or len(words) > 2):
      pass
    else:
      if(words[1] in outofvocabcombined):
        pass
      else:
        si_words1.append(words[0])
        en_words1.append(words[1].lower())

output1 = []
count = 0
for a,b in zip(si_words1, en_words1):
    output1.append((a,b))
    count = count + 1
```

D.3 Code Listings of Document Retrieval

```
resultSet = []
for i in range(5):
  startIndex = i * 10 + 1
  response = requests.get("https://www.googleapis.com/customsearch
/v1?key=AIzaSyCGDzvV_1kxkgos74hTSlkFxPhdI9k1Se4&cx=009597651223148
113194:9p08gtfdixe&q="+translatedSearchQuery+"&start="+str(startIn
dex)+"&siteSearch=www.youtube.com&siteSearchFilter=e&gl=lk")
  results = response.json()
  for item in results['items']:
    resultSet.append(item)

raw_documents = []
translated_documents = []

import time

translateDocuments(resultSet[0:10])
time.sleep(150)
translateDocuments(resultSet[10:20])
time.sleep(150)
translateDocuments(resultSet[20:30])
time.sleep(150)
translateDocuments(resultSet[30:40])
time.sleep(150)
translateDocuments(resultSet[40:50])

print("Number of documents:",len(raw_documents))
print("Number of translated documents:",len(translated_documents))
```

D.4 Code Listings of Re-Ranking Models

The following code segment depicts the basic re-rank model.

```
tokens_all = []
for translated_document in translated_documents:
  input_text = re.sub(r'\([^)]*\)', '', translated_document)
  sentences_strings = []
  for line in input_text.split('\n'):
      m = re.match(r'^(?:(?P<precolon>[^:]{,20}):)?(?P<postcolon>.
*)$', line)
```

```
      sentences_strings.extend(sent for sent in m.groupdict()['pos
tcolon'].split('.') if sent)
  # store as list of lists of words
  sentences = []
  for sent_str in sentences_strings:
      tokens = re.sub(r"[0-9]+", " ", sent_str.lower()).split()
      sentences.append(tokens)

  tokens = []
  for sentence in sentences:
      for word in sentence:
        tokens.append(word)
  tokens_all.append(tokens)

from nltk import FreqDist

translateTokens = searchQuery.split(' ')

freq_dist_list = []

total_count_list = []
for one_token in tokens_all:
  total_count = 0
  freq_dist = FreqDist(one_token)
  for translateToken in translateTokens:
    total_count = total_count + freq_dist[translateToken]
  total_count_list.append(total_count)

normalized_count_list = []
i = 0
for count in total_count_list:
  normalize_count = count / (len(translated_documents[i]) + 1)
  normalized_count_list.append(normalize_count)
  i = i + 1

import numpy as np

arr = np.array(normalized_count_list)

re_rank = arr.argsort()[-3:][::-1]

print(re_rank)

for i in re_rank:
  doc = resultSet[i]
  print("\n" + "Title: " + doc['title'] + "\nURL: " + doc['link'])
```

The following code segment depicts the LSI based re-rank model.

```
lsi = LSI(translated_documents, translatedSearchQuery)
ranking = lsi.process()

print(ranking)

import numpy as np

arr = np.array(ranking)

re_rank = arr.argsort()[:3][::1]

print(re_rank)

for i in re_rank:
  doc = resultSet[i]
  print("\n" + "Title: " + doc['title'] + "\nURL: " + doc['link'])
```