Use of interpretable evolved search query classifiers for sinhala documents

KANKANAMALAGE, Prasanna Haddela, HIRSCH, Laurence <http://orcid.org/0000-0002-3589-9816>, BRUNSDON, Teresa and GAUDOIN, Jotham

Published version

KANKANAMALAGE, Prasanna Haddela, HIRSCH, Laurence, BRUNSDON, Teresa and GAUDOIN, Jotham (2020). Use of interpretable evolved search query classifiers for sinhala documents. In: ARAI, K., KAPOOR, S. and BHATIA, R., (eds.) Proceedings of the Future Technologies Conference (FTC) 2020,. Advances in Intelligent Systems and Computing, 1 . Springer International Publishing, 790-804.

Copyright and re-use policy

Use of Interpretable Evolved Search Query Classifiers for Sinhala Documents

Prasanna Haddela[1,3], Laurence Hirsch[1], Teresa Brunsdon[2] and Jotham Gaudoin[1]

[1] Sheffield Hallam University, Sheffield S1 1WB, United Kingdom
[2] University of Warwick , Coventry, CV4 7AL, United Kingdom
[3] Sri Lanka Institute of Information Technology, Colombo, Sri Lanka
prasanna.s@sliit.lk

Abstract. Document analysis is a well matured yet still active research field, partly as a result of the intricate nature of building computational tools but also due to the inherent problems arising from the variety and complexity of human languages. Breaking down language barriers is vital in enabling access to a number of recent technologies. This paper investigates the application of document classification methods to new Sinhalese datasets. This language is geographically isolated and rich with many of its own unique features. We will examine the interpretability of the classification models with a particular focus on the use of evolved Lucene search queries generated using a Genetic Algorithm (GA) as a method of document classification. We will compare the accuracy and interpretability of these search queries with other popular classifiers. The results are promising and are roughly in line with previous work on English language datasets.

Keywords: Evolved Search Queries, Genetic Algorithm, Interpretable text Classification, Lucene Sinhala Analyzer, Sinhala Document Classification.

1 Introduction

Document analysis is a mature research field that has continued to grow over the past few decades. It remains an active field of research and development not only because of the complex nature of building computing tools and techniques and the ever-evolving nature of technology but also because of inherent problems due to the variety of languages spread all over the world. For example, languages that are isolated (whether geographically or in their evolution) have their own unique features. Therefore, language and computing experts from these communities need to fine tune any technological tools, which are often developed with more widely spoken languages in mind. However, due to a lack of experts, some languages do not have even the basic tools to analyze documents. In terms of modern technologies, this makes some communities disconnected from others in the world as they do not have access to recent technological NLP tools. This research aims to contribute to the development of such tools for the Sinhalese language (Sinhala), which at present is one of the underresourced and underrespresented languages in the world in terms of available technological tools.

Sinhala, which belongs to the Indo-Aryan language family, is the native language of the Sinhalese people who comprise the largest ethnic group in Sri Lanka. Although English is used in Sri Lanka for communicating internationally, Sri Lankans predominantly use Sinhala in both formal and informal activities. Further, it is one of the two official languages in the country alongside Tamil. Most official documents are written in Sinhala. As a low resource language, many government officers face a range of practical problems in the modern digital world with respect to processing Sinhala documents.

Many novel applications that benefit society using a combination of data and computing power have been developed. In particular, supervised and unsupervised computing techniques are used to analyse very large datasets using high end processors and scalable computing frameworks to efficiently produce a range of insights from the input data. Due to a lack of resources and computing tools, Sinhala documents are not currently processed at any kind of scale.

At present, supervised and unsupervised learning algorithms are facing a new set of challenges and are being interrogated more closely. Human interpretability and transparency are missing in such 'black box' algorithms and this has severely affected the level of trust in and accountability of such algorithms. Computing professionals should be able to provide valid justifications for the action and decisions that have been taken. If this is not the case then such methods are less effective because their results are not trusted.

Evolved Search Queries are rich in human interpretability and transparency and they are also explainable. Therefore, this research aims to investigate how to use evolved search queries for effective Sinhala document classification. In order to achieve this, we carried out the following activities: web scraping to produce training and testing datasets, development of a tokenizer for Sinhala, generation of a Sinhala stopword list using tf-idf, development of a Sinhala stemming method. Finally, we integrated the above with an Apache Lucene full text search engine to create an evolved search query builder using a Genetic Algorithm (GA).

For our newly constructed Sinhala datasets, we consider: (i.) the success rate of popular classifiers against our evolved search classifier; (ii.) how to use full-text indexing engines; (iii.) how to fine tune full-text indexing frameworks and (iv.) the level of interpretability in Sinhalese document classification.

The paper is organised as follows: Section 2 discusses the importance of interpretable predictive models; Section 3 is a review of the document classification processes; Section 4 outlines the Methods and tools used; in Section 5, we consider the new Lucene Sinhala analyzer that we developed; in Section 6, we present our experimental results and our conclusion are presented in Section 7.

2 Importance of Interpretable Algorithms

In the recent past, there have been a number of incidences of questionable ethical standards within the computing community in that the adoption of novel technological

changes within data driven applications has not happened in line with relevant ethical frameworks. A few notable cases are given below.

The Durham Constabulary's HART — Harm Assessment Risk Tool (HART) is a risk-assessment tool which is used in the UK police service [18]. This tool was invented by experts at the University of Cambridge in collaboration with Durham Constabulary. Its primary aim is to aid the decision-making process of custody officers when assessing the risk of future offending.

There are several critiques against the HART due to the opaque nature of its decision-making algorithm and the lack of comprehensible explanation of the relationship between the data inputs and the conclusion provided. Without such an explanation, any decisions could be challenged by the individual affected or their legal advisers.

Cambridge Analytica Ltd (CA) was a British political consulting firm. CA developed an extremely powerful software solution to predict and influence voters' choice at elections. It has been found [3] that the 2016 United States presidential election and the 2016 United Kingdom European Union membership referendum (the Brexit referendum) are two main cases where CA played a vital role.

CA used Facebook to harvest millions of users' profiles and then build models to predict users' behaviour. These insights were used to direct and manipulate political campaigns. The data were collected through an app called "thisisyourdigitallife" that was owned by Global Science Research (GSR). With the support of GSR, CA collected millions of users' data. While Facebook's "platform policy" allowed only collection of friends' data to improve user experience in the app and barred it from being sold on or used for advertising, CA has violated this policy and was found to be unlawful. It is important to know what, when, where and how the data were used to determine the scale of the criminality and to take any necessary actions against interested parties. However, this is not easy due to the limited transparency of the systems in question.

COMPAS software — COMPAS is used in bail decision making in the Wisconsin supreme court in the US [16]. There was a complaint against the system where the results were biased against black defendants, despite race not being used as a predictor. This draws attention to the 'technology effect' of 'automation bias' in computerized forecasting that has not been investigated by a system operator or decision maker at the level of human individuals. Inner algorithmic workings and data weightings were not revealed to the defendant due to the commercial confidentiality. This has a negative impact on the level of trust in the system. Once again, computing professionals are challenged to make a trade-off between potential profits and algorithmic transparency.

These cases are evidence of the importance of interpretable algorithms. In Europe, the new General Data Protection Regulation (GDPR) legislation requires that predictive models can be explained [10]. Indeed, ideally, experts need justifiability, defined as being able to show that their models are in line with existing domain knowledge and any relevant legal and ethical frameworks.

3 Background of Document Classification Methods

The process of document classification involves three key phases. These are: document representation, classifier construction and model evaluation [14, 23]. This section briefly discusses the methods that have commonly been used within the three phases. Figure 1 shows a holistic view of all three phases, with Phase I being split into two sub-phases.

Phase I		Phase II	Phase III
Pre-processing and Indexing	Dimension reduction	Classifier construction	Model evaluation

Fig. 1. Process of document classification

3.1 Phase I: Document Representation

Phase I aims to represent a document collection in a form that induction algorithms can use to produce effective classifiers. The unclear semantics of natural language affects the interpretability of a classifier badly. Further, high dimensionality results in low accuracy and efficiency of the classifier. Here, one of the key challenges is to retain the semantics of the natural language while minimizing the dimensionality of the text data. Therefore, preprocessing and dimensiona reduction are key stages in the document classification process.

Preprocessing involves capturing, cleaning, and smoothing the features of textual data and organizing them to support the process of computing. Technically, three main methods have been used in much existing research for initial preprocessing. These are: Bag-of-Words, Bag-of-Phrases and instance selection [25]. Among these, Bag-of-Words is the most popular and widely accepted approach [12]. This method begins by first removing irrelevant and noisy data, followed by breaking the text into tokens (terms), removing stopwords, and stemming. The Bag-of-Phrases method is semantically richer but computationally more expensive in comparison to the Bag-of-Words method as it does this on a phrase-by-phrase rather than word-by-word basis. Instance selection or the use of a sample set from a document collection is the other possible techniques. However, due to the availability of powerful computational facilities and large storage systems, this is rarely used [25].

In most research, preprocessed document collections are represented in a Vector Space Model (VSM). In such models, the vector that represent each document in the collection contains term features of the document. Further, term weighting methods and normalization techniques are used to smoothen the VSM.

In this paper, we have used a full-text search engine and indexed all the documents after preprocessing. This method improves the speed of data access. Additionally, the distributed processing capability of full-text search engines increases the scalability of the system.

Employing Dimensional Reduction (DR) technique is an important step that can improve the efficiency and accuracy of classifiers [6, 7, 26]. Two ways of conducting

DR are feature selection (FS) and feature extraction (FE) [1]-[5]. FS aims to select the subset of features that has the greatest predictive power for classifying the documents in question into categories. These FS methods may be grouped into one of three main approaches, namely filters, wrappers and embedded methods. FE algorithms are sometimes known as feature transformation algorithms. These aim at extracting features by projecting the original high dimensional data into a lower dimensional space through algebraic transformations [28]. Some of the popularly used FE methods are: Principal Component Analysis (PCA), Latent Semantic Analysis (LSA) and Linear Discriminant Analysis (LDA) [2]. Due to its higher classification accuracy and interpretability of the features produced, we have used the Chi-squared FS method [1] for DR in our experiments.

3.2 Phase II: Classifier Construction

Phase II aims to construct a classification model which can assign previously unseen documents into a prelabeled category. Table 1 shows some of the popular classifiers and their categories based on their origin or key characteristics [1]. We have used this same set of algorithms for the experiments in this research.

Table 1. Category of Classifiers

Category	Algorithms
Tree-based	C4.5, Random Forest (RF)
Rule-based	PART, JRip
Distance-based	k-Nearest Neighbours (kNN)
Function-based	Support Vector Machine (SVM), Deep Learning (DeepL)
Statistical	Naïve Bayes (NB)
Genetic Algorithm	Evolved Search Query (eSQ)

In reality, there are no perfect classifiers since each performs well in certain conditions. Also, there may be situations where two humans would not agree on the same category for a particular document. Therefore, selecting most appropriate classifier is a common challenge. Also, some highly accurate classifiers such as SVM, DeepL, RF have no transparency nor human interpretability. This is a major concern for certain applications, due to the limitations in monitoring, fine-tuning and especially in justifying the reasons for any decisions made by the algorithm. The eSQ [9] method is a GA based classifier which has the benefits of being easily interpreted and modified by a human. This paper presents the results obtained when classifying Sinhalese documents using eSQ alongside a comparative analysis of results using more traditional classifiers.

3.3 Phase III: Model Evaluation

Phase III aims to find the most effective classification model for a particular application. It is common practice to use experiment-based methods to evaluate

classifiers. Two main methods are k-fold cross validation and the hold-out method. k-fold cross validation splits a dataset into k groups and runs the classification experiment k times. Each time, one group of data is used as the test set and the classifier is trained on the other (k-1) groups of data. The classification accuracy is then averaged over the results of the k runs. The hold-out method splits the dataset into a training subset and a test subset. A classifier is trained on the training subset and tested on the test subset [29]. Precision, Recall, and the F-measure are popular accuracy measures for document classification. Note, however, that the macro F-measure provides a more realistic measure when a dataset is balanced, with equal category sizes, while the micro F-measures provides a more realistic measure when a dataset is imbalanced, with unequal numbers of documents in each category [27]. More details are provided in subsection 6.2.

4 Research Method and Materials

4.1 Method in Brief

Sinhala documents were collected using a web scraping tool. All documents are news articles published on the web. The collected documents have been grouped into a small number of categories and labelled. Next, Apache Lucene was used for preprocessing and indexing. Due to the unavailability of Sinhalese preprocessing tools in Lucene, we have developed a Sinhala Analyzer and integrated it into the Lucene framework. Our GA based eSQ classification engine has been designed to access document collections in the Lucene framework. Therefore we have been able to smoothly integrate the eSQ with Lucene and also to build eSQ classifiers for Sinhalese document collections. We then conducted a series of experiments using this classifier and compared the performance of our new Sinhala analyzer against other popular classifiers.

4.2 Datasets

For classifier induction, pre-labeled datasets are essential for training the model and testing its accuracy. Due to the unavailability of benchmark datasets for the Sinhalese language, we have developed two main document collections. Both of these contain news articles that are publicly available to access on the web. Our first dataset, SLNG_rands, contains 81606 randomly collected documents. It does not have the categorized documents needed for classification, but may be useful for unsupervised learning or other NLP-related research. In this paper, we have used it for stopword generation. Our second dataset – the SLNG collection - contains pre-labeled news articles categorized into 7 groups. By combining datasets published in [3] and [15] with the SLNG collection, we have formed 5 datasets for our purpose. Details of these datasets are shown in table 2.

SLNG3 contains news articles from 3 sport categories (cricket, football and rugby). These three categories contain more overlapping terms. SLNG4, SLNG5, SLNG6 and SLNG7 were created by adding the entertainment, politics, crime and religion categories of documents respectively into the previous SLNG dataset. This is so as to

increase the diversity of the document collection gradually and also to increase the number of categories in the datasets.

Table 2. The Structure of Datasets

Dataset name	No. of categories	No. of documents	Category names [Category size]
SLNG3	3	2550	cricket [850] / football [850] / rugby [850]
SLNG4	4	4050	cricket [850] / football [850] / rugby [850] / entertainment [1500]
SLNG5	5	4250	cricket [850] / football [850] / rugby [850] / entertainment [1500] / politics [200]
SLNG6	6	4450	cricket [850] / football [850] / rugby [850] / entertainment [1500] / politics [200] / crime [200]
SLNG7	7	4650	cricket [850] / football [850] / rugby [850] / entertainment [1500] / politics [200] / crime [200] / religion [200]

4.3 Document Indexing with Apache Lucene

Apache Lucene is a widely accepted full-text search engine. It is an open source project providing Java-based indexing and search technology. It provides a simple but powerful API that hides the complexity of indexing and searching [17]. The fundamental concepts in the Lucene data model are documents, fields and indexes. A Lucene document consists of fields, where each field has a name, and unstructured textual content. A Lucene document may contain multiple fields. A Lucene index is a set of documents stored in a persistent storage medium, supported by data structures providing efficient data retrieval. This software framework is enriched with a highly scalable indexing architecture, (relatively) small RAM requirements and supports different query types (for example: boolean, phrase, wildcard, proximity, range) and has multi-language support [30](though not for all languages). Lucene integration leverages the power of document classification. The Apache Lucene framework provide analyzers for over 40 languages, but not the Sinhalese language. One of our main contributions is to fill this gap and to investigate possible further improvements. Sinhala integration with Lucene is discussed in Section 5.

4.4 Evolved Search Query Engine: An Overview

Our Evolved Search Query (eSQ) engine is a GA-based document classification query builder [9]. It produces a single search query (classifier) for each category. Such a search query consists simply of a small set of human readable words that represent all documents in a category. Each category has a unique search query. An eSQ is a binary classifier for that particular category. Thus, to classify a document into a category, it is required to determine whether the document is returned by the search query.

As discussed in subsection 3.1, we have used a filter-based FS method (namely chi-squared) to rank features. Using the chi-squared method, the top 200 terms of each category were taken as the initial population for the eSQ engine. The eSQ engine uses the F-measure as a fitness function. It is an objective function used for achieving the optimal solution when it is evolving. The ECJ (http://cs.gmu.edu/~eclab/projects/ecj/) Java library is used for evolutionary computation and Apache Lucene is used for full-text indexing when producing results for the search query. Table 3 shows other important parameters used in our GA system. eSQ has performed well with these parameters when used with English documents [9].

Table 3. GA Parameters

Parameter	Value	Parameter	Value
Population	1024	Reproduction probability	0.1
Generations	500	Crossover probability	0.7
Selection type	Tournament	Elitism	No
Tournament size	5	Subpopulations	2
Termination	Max generations	Chromosome length	variable
Mutation probability	0.1	F1WordList length	200

5 Preprocessing tools for Sinhala

5.1 Overview of New Lucene Sinhala Analyzer

The language analyzer is a core component of any document processing system. Processing documents in the Sinhalese language was a challenge due to the unavailability of the basic preprocessing tools required. Consequently, we have designed and developed a language analyzer for the Sinhalese language. This consists of a Sinhala tokenizer, stopword list and stemmer and these are fully compatible with the Lucene framework.

5.2 Sinhala Tokenizer

Tokenization is the separating and (possibly) breaking into small units of a string of input characters. The resulting tokens (terms) are then passed on to other language processing tools. A tokenizer forms the initial step and creates a starting point for other preprocessing operations.

Tokenization is highly language-dependent. For an example, tokenizers developed for English cannot be used as is for Chinese or Arabic since the languages are inherently different in many ways. Therefore, it is useful to have language-specific tokenizers. Rule-, statistical-, fuzzy-, lexical- and feature-based techniques are often employed when designing a tokenizer. Our Sinhala tokenizer was developed using a rule-based technique and in developing it we considered languages that are similar in terms of tokenzation. It has two main components: (i.) punctuation-based tokenization and (ii.) dependent word tokenization.

The Sinhalese language has 15 punctuation symbols and some of these are unique to the language. For example, the ෴ symbol is used to show the end of a sentence or a paragraph in old documents. Furthermore, the meaning of some Sinhala punctuation marks differs depending on the context in which it is used. Therefore, in the Sinhala tokenizer, language-specific patterns have been identified and appropriate rules have been applied to produce the token set. The dependent word tokenization component is aimed at identifying words that differ in meaning when they are together rather than separate. However, finding dependent words is a computationally high-cost operation. The details of the production of the tokenizer and our experiments are published in [24].

5.3 Sinhala Stopword List

Stopword removal is a basic preprocessing step in NLP. It filters out redundant words that hold little information and have low or no semantic meaning for the given text [22]. “Is”, “are”, “in”, “for”, “that” are some examples of stopwords in English. Removal of stopwords helps us to decrease the size of the corpus and increases the efficiency and accuracy [22] of NLP tasks.

Fox [5] is one of the main contributors to find stopwords and his method has created a standard list for English consisting of 421 words. In the literature, we found that stopword lists have been generated for Arabic [4] and regional languages such as Sanskrit [21], Punjabi [20], Gujarati [19] and many more languages. We also found some attempts for Sinhala which used a rather small group of documents [8, 15].

In this research, we generated a stopword list using our SLNG_rands dataset which contains 81606 randomly collected news articles. Our stopword list was produced using tf-idf ranking and consists of 210 Sinhala words. We further tested it using a number of classifiers – see [11] for details. This list is called inside the Lucene Sinhala analyzer that we have developed.

5.4 Sinhala Light Stemmer

Stemming converts the original word into its root format, which is called its stem. The stemming process plays a prominent role in NLP because it makes applications more efficient and effective. There are five types of words in Sinhala. Each of these words is formed by combining one or more morphemes with the base form. Sinhala morphemes are divided into four main types known as bases, suffixes, prefixes and infixes. We have developed a set of rules that reduce words back to their base form. However, initial experiments found that the stemming algorithm over truncates and this is damaging to effectiveness [13]. Also, in the Sinhala language, if a prefix is removed from a word then it gives the opposite meaning of the word. Again, this badly affected the effectiveness of classification. Therefore, the Sinhala ‘Light’ Stemmer that we have integrated with Lucene does not consider prefix removal of words during stemming.

6 Experiments and Results

6.1 Experimental objectives

We conducted a series of experiments with the following three objectives:

a. To investigate whether the accuracy of our eSQ classifier when classifying Sinhalese documents deviates significantly from other popular classifiers.

b. To compare classification accuracy between our Lucene Sinhala Analyzer and the Lucene Standard Analyzer when classifying Sinhalese document using eSQ.

c. To Investigate the human readability of the eSQ classifiers produced by our Sinhala analyzer and the Lucence Standard analyzer.

6.2 Evaluation metrics

Classification model evaluation was carried out using the hold-out method. The datasets shown in table I were used for all the experiments. 50% of the data was used for training and the remaining 50% for testing. The number of categories in the datasets varies from 3 to 7 and the datasets other than SLNG3 are imbalanced. Therefore, we have computed Micro F using micro precision and micro recall measures to make more realistic assessments. Equation 1, 2 and 3 represent micro precision, micro recall and micro F respectively.

$$Micro\ Precision, p = \frac{\sum TP}{\sum TP \ + \ \sum FP} \tag{1}$$

$$Micro\ Recall, r = \frac{\sum TP}{\sum TP \ + \ \sum FN} \tag{2}$$

$$Micro\ F = \frac{2pr}{(p+r)} \tag{3}$$

A classifier may erroneously classify a non-member of the target category as positive and a member of the category as negative. These documents are called False Positives (FP) and False Negatives (FN) respectively. The correctly classified positive documents are the True Positives (TP), while the correctly classified negative documents are the True Negatives (TN).

6.3 Does eSQ perform well in Sinhala document classification?

The results in Table 4 show the Average Micro F score for the 9 popular classifiers discussed above in Table 1 for the five datasets detailed in Table 2, with the best results displayed in bold. The average Micro F score has been computed after executing each classifier 10 times on each dataset.

Table 4. Summary of Average Micro F

	eSQ	C4.5	RF	PART	JRip	kNN	SVM	NB	DeepL
SLNG3	0.9441	0.9405	0.9443	0.9303	**0.9452**	0.8451	0.9420	0.8897	0.9361
SLNG4	0.9369	0.9525	0.9518	0.9457	0.9554	0.7229	**0.9589**	0.9176	0.9404
SLNG5	0.9315	0.9547	0.9491	0.9460	0.9550	0.7041	**0.9607**	0.9056	0.9424
SLNG6	0.9044	0.9472	0.9422	0.9375	0.9430	0.6820	**0.9596**	0.9020	0.9223
SLNG7	0.9038	0.9388	0.9426	0.9304	0.9304	0.6544	**0.9590**	0.9079	0.9245

Figure 2 uses boxplots to show the variability of the Micro F values for each classifier. It shows that most of the classifiers have comparable results, with the exception of the kNN classifier, which is considerably worse. For this reason, we omit the kNN classifier from further analyses. This also confirms that our eSQ classifiers are among the top classification techniques when classifying Sinhalese documents.

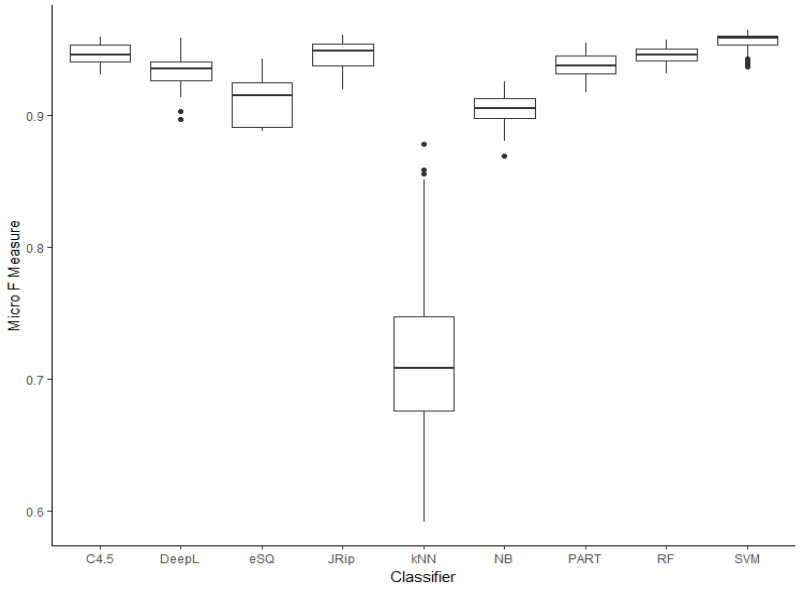

Fig. 2. Variability in Micro F

Before conducting a detailed analysis using ANOVA, we transformed Micro F scores into logit Micro F values. This ensures that our target variable has the whole real line as its range of possible values, rather than just values in the interval [0,1].

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Algorithm	7	5956	850.8	304.15	<0.00001	***
Dataset	4	932	233.0	83.31	<0.00001	***
Algorithm:Dataset	28	1255	44.8	16.02	<0.00001	***
Residuals	360	1007	2.8			

Fig. 3. Summary of ANOVA Test

Figure 3 shows the ANOVA test output and it confirms that there is a significant interaction between the algorithm used and the dataset being considered, so that different algorithms are better with different datasets.

Figure 4 shows the relationship between classifiers and logit F for each dataset. This shows that our eSQ classifier performs well when a dataset has more overlapping categories than when it has a number of diverse categories.

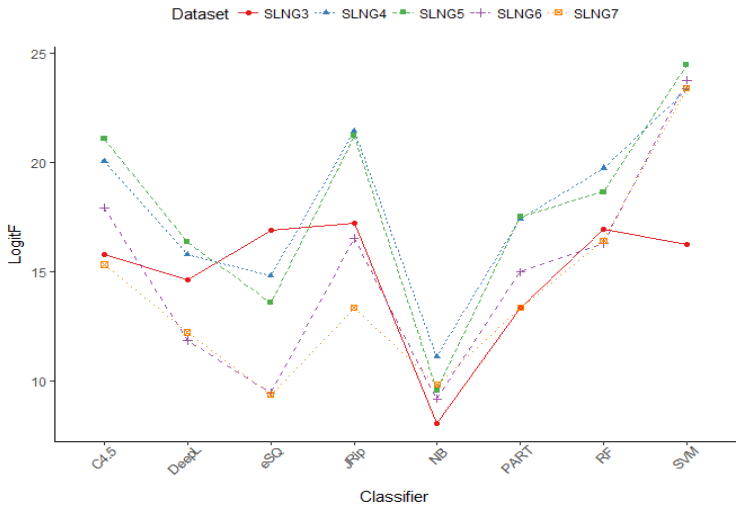

Fig. 4. Interaction between Classifiers, Datasets and LogitF

6.4 Does our proposed Sinhala Analyzer perform well?

Experiments were conducted using both the Standard Lucene Analyzer and our recently developed Sinhala Analyzer. Table 5 shows the relative performance observed using

Table 5. Micro F for the Lucene and Sinhala Analyzers

	SLNG3	SLNG4	SLNG5	SLNG6	SLNG7
Lucene Analyzer	0.9437	0.9498	0.9297	0.8995	0.8948
Sinhala Analyzer	0.9186	0.9088	0.9080	0.8915	0.8854

our eSQ classifier. As can be seen from our experimental results, the overall accuracy of the Sinhala analyzer is slightly less compared with that of the Standard Lucene Analyzer. However, we found that for some categories, the Sinhala analyzer performed better than the Standard Lucene Analyzer despite the fact that its overall accuracy is slightly lower. These category-level details are presented in Figure 5.

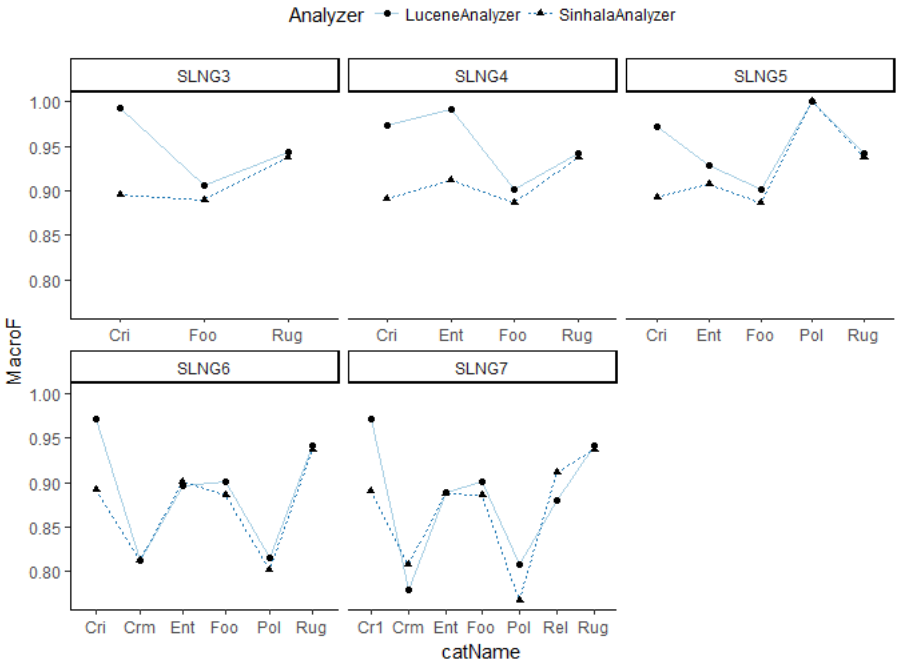

Fig. 5. Performance for Category level

6.5 Interpretability of eSQ Classifiers

The eSQ classifiers are highly human interpretable. Ultimately, they are merely a small number of words put together. Table 6 shows the eSQ classifiers produced for the SLNG3 dataset using both the Standard Lucene Analyzer and our new Sinhala analyzer.

Table 6. Sinhala eSQ classifiers

Cat Name	Analyzer	F score	eSQ classifier
Rugby	Lucene Analyzer	0.952	උත්සාහක(try) හැව්ලොක්ස් (Havelock) දිනුමකින්(win) රග්බි (rugby)
	Sinhala Analyzer	0.952	දිනුම(win) රග්බි (rugby) හැව්ලොක්ස් (Havelock)
Football	Lucene Analyzer	0.906	පාපන්දු (football) ගෝලය (goal) සම්මේලනයේ (association)
	Sinhala Analyzer	0.893	පාපන්දු (football) සම්මේලන (association)
Cricket	Lucene Analyzer	0.973	ඉනිම (innings) ක්‍රිකට් (cricket) දැවී (out) කඩුලු (wicket) ඉනිමේ (innings)
	Sinhala Analyzer	0.910	පිත්(bat) ඉනිම (innings) කඩුල් (wicket) නොදැවෙ (not out) ඉනිම් (innings) දැවෙ (out)

For the Rugby and Football categories, our Sinhala analyzer has produced more compact queries without losing much accuracy. However, the opposite is true in the cricket category, where the query is both longer and less accurate.

7 Conclusion

Sinhala Document classification is not a well-studied subdomain of text analytics, despite the fact that this field is well matured for some languages. Our experiment has shown that eSQ is a good text classifier and produces comparable results to other popular methods, while having the added advantage of human interpretability. As a part of this study, we have created a new Sinhala analyzer for the Lucene full-text search engine. Results confirm that our new analyzer performs better for some categories than the standard analyzer does. It is also capable of producing more compact search queries. However, we note that the Sinhala stemmer integrated in our new analyzer should be further improved to improve the analyzer as a whole.

References

1. Aggarwal, C.C., Zhai, C.X.: A survey of text classification algorithms. In: Mining Text Data. pp. 163–222 Springer US, Boston, MA (2012).
2. Cunningham, P.: Dimension reduction. Machine learning techniques for multimedia. 91–112 (2008).
3. Ekanayaka, R.K.S.K. et al.: Sinhala news analysis using text mining and machine learning. In: 5th Ruhuna Int. Science and Technology Conference. , Matara (2018).
4. El-Khair, I.A.: Effects of stop words elimination for Arabic information retrieval: a comparative study. International Journal of Computing & Information Sciences. 4, 3, 119–133 (2006).
5. Fox, C.: A stop list for general text. ACM SIGIR Forum. 24, 1–2, (1990).
6. Fragoso, R.C.P. et al.: Class-dependent feature selection algorithm for text categorization. In: 2016 International Joint Conference on Neural Networks (IJCNN). pp. 3508–3515 IEEE, Vancouver, BC, Canada (2016).
7. Gonçalves, E.C. et al.: Simpler is Better: a Novel Genetic Algorithm to Induce Compact Multi-label Chain Classifiers. In: 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO '15). pp. 559–566 ACM, Madrid, Spain (2015).
8. Gunasekara, S., Haddela, P.: Context aware stopwords for Sinhala Text classification. In: National Information Technology Conference. IEEE, Colombo, Sri Lanka (2018).
9. Hirsch, L., Brunsdon, T.: A Comparison of Lucene Search Queries Evolved as Text Classifiers. In: Applied Artificial Intelligence. pp. 768–784 Taylor and Francis Inc. (2018).
10. Intersoft Consulting: General Data Protection Regulation (GDPR) – Official Legal Text. In: GDPR-info.eu. (2018).
11. Jayaweera, A. et al.: Dynamic Stopword Removal for Sinhala Language. In: National Information Technology Conference. , Colombo, Sri Lanka (2019).
12. Jindal, R. et al.: Techniques for text classification: Literature review and current trends.

Webology. 12, 2, 1 (2015).
13. Kariyawasam, P. et al.: A Rule Based Stemmer for Sinhala Language. In: 14th IEEE International Conferene on Industrial and Information Systems. , Sri Lanka (2019).
14. Khan, A. et al.: A Review of Machine Learning Algorithms for Text- Documents Classification. Journal of Advances in Information Technology. 1, 1, 4 (2010).
15. Lakmali, K., Haddela, P.: Effectiveness of rule-based classifiers in Sinhala text categorization. In: National IT Conference. IEEE, Sri Lanka (2017).
16. McKay, C.: Predicting risk in criminal procedure: actuarial tools, algorithms, AI and judicial decision-making. Current Issues in Criminal Justice. 1–18 (2019).
17. Milosavljević, B. et al.: Retrieval of bibliographic records using Apache Lucene. The Electronic Library. 28, 4, 525–539 (2010).
18. Oswald, M.: Algorithm-assisted decision-making in the public sector: Framing the issues using administrative law rules governing discretionary power. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 376, 2128, (2018).
19. Patel, P.H.: Pre-Processing Phase of Text Summarization Based on Gujarati Language Gender and Number Identification: Rule-Based Approach View project. (2014).
20. Puri, R. et al.: Automated Stopwords Identification in Punjabi Documents. Research Cell: An International Journal of Engineering Sciences. 8, (2013).
21. Raulji, J.K., Saini, J.R.: Generating Stopword List for Sanskrit Language. In: 7th International Advance Computing Conference (IACC). IEEE, Hyderabad, India (2017).
22. Saini, J.R. et al.: Stop-Word Removal Algorithm and its Implementation for Sanskrit Language. Article in Int. Journal of Computer Applications. 150, 2, 975–8887 (2016).
23. Sebastiani, F.: Machine learning in automated text categorization. ACM computing surveys (CSUR). 34, 1, 1–47 (2002).
24. Senanayake, S. et al.: Enhanced Tokenizer for Sinhala Language. In: National Information Technology Conference. , Colombo, Sri Lanka (2019).
25. Tsai, C.-F. et al.: Evolutionary instance selection for text classification. Journal of Systems and Software. 90, 104–113 (2014).
26. Uğuz, H.: A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems. 24, 7, 1024–1032 (2011).
27. Uysal, A.K.: An improved global feature selection scheme for text classification. Expert Systems with Applications. 43, 82–92 (2016).
28. Yan, J. et al.: Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing. IEEE Transactions on Knowledge and Data Engineering. 18, 3, 320–333 (2006).
29. Yu, B.: An evaluation of text classification methods for literary study. Literary and Linguistic Computing. 23, 3, 327–343 (2008).
30. Apache Lucene Documentation, http://lucene.apache.org/, last accessed 2020/01/14.