



Low-Resource Natural Language Identification

Tendai Midzi

MAY 2021



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Low-Resource Natural Language Identification

Tendai Midzi

A thesis submitted for the degree of Bachelor of Technology
(Computer Systems and Networking)

May 2021

Synopsis

The aim of the project is to identify and propose a method of study capable of being used to identify low resource languages given Malay and Iban and to use this as a base approach to solving the issue of Low-Resource Natural Language Identification.

There is evidence of a gap in the conservation and preservation of Low resource languages as they inch closer and closer to extinction with regardless of the advent of technology. This will result in a huge cultural and identity loss as people become used to using only High Resource Languages due to migration and the unequal advancements in technology.

With Iban and Malay in particular, work has been carried in so as as to build their corpora and identify the structure and overall usage of both languages. However, there has been a lacking in the specific study of identifying whether or not a given text document is in Malay or Iban.

The proposed Methods to be used include;

Bag of Words

For Corpus Creation and Similarity Checks. Whereby all the words in a given document are taken and processed to remove punctuation and any other non-alphanumeric characters that do not add to the word itself

Cosine Similarity

To mathematically calculate the similarities between documents and languages by use of a proven method that is often applied to documents when the similarity of the document has to be measured without size being an influence

Term Frequency-Inverse Document Frequency

To average out term frequency and assist in the identification of the language in the Cosine Similarity function.

Overall, the project will look to cater to text documents and webpages.

Tendai Midzi,
Lot 9238,
Curtin Water 2,
98009 Miri,
Sarawak, Malaysia.

May 31, 2021

A/Prof. Dr. Lenin Gopal,
Head of Department,
Department of Electrical & Computer Engineering,
Faculty of Engineering and Science,
Curtin University, Malaysia Campus
CDT 250, 98009 Miri
Sarawak, Malaysia.

Dear A/Prof. Dr. Lenin,

I, Tendai Midzi, hereby submit my thesis entitled “Low-Resource Natural Language Identification” as part of my requirements for completion of the Bachelor of Technology in Computer Systems and Networking.

I declare that this thesis is entirely my own work with the exception of the acknowledgements and references mentioned.

Yours sincerely,

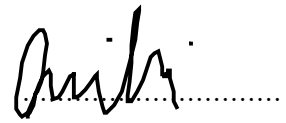
A handwritten signature in black ink, appearing to read 'Tendai Midzi', followed by a dotted line.

Tendai Midzi
700026324

TITLE Low-Resource Natural Language Identification			
AUTHOR Family Name: Midzi Given Name: Tendai			
DATE May, 2021		SUPERVISOR Dr. Thomas Anung Basuki	
DEGREE Bachelor of Technology		OPTION Computer Systems and Networking	
ABSTRACT <p>With the ever-changing world of technology, most people have resorted to using the some core languages such as French, English and Chinese to communicate and build their software on. This leaves low resource languages such as Iban at the fringes of technological development and risk the possibility of them become endangered and largely extinct if not incorporated into the new digital age. In this paper, we aim to develop a language identification system based on text input using two closely related languages namely; Malay and Iban. With Malay being a relatively high resource language and Iban being the low resource language. Such a system would allow the easier implementation and understanding of various text based documents and systems using Iban in relation to Malay. The study can be then replicated using other linked or related languages.</p>			
INDEXING TERMS Natural Language Processing (NLP); Low Resource Languages (LRL); High Resource Languages (HRL); Cosine Similarity; Term Frequency - Inverse Document Frequency(TF-IDF).			
	GOOD	AVERAGE	POOR
Technical Work			
Report Presentation			
Examiner		Co-Examiner	

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made. This thesis contains no material which has been accepted for award of any other degree or diploma in any university. This document also complies with the rules of Curtin University on copyright and plagiarism.



Tendai Midzi

31 May 2021

.....
Date

Acknowledgement

Throughout the research and writing of this project thesis, I have been strongly supported by those around me.

I would first like to thank my supervisor, Dr. Thomas Anung Basuki, for entrusting me with this topic and being an invaluable guiding light in the research and overall implementation of the work. Your advice and guidance pushed my understanding and interest in the work itself.

I would also like to thank and acknowledge my Mother for her constant prayers, support and dedication to education, without which this would not have been possible. And my sister and brother; Tamia and Ivan for lending their names for examples in this document

Finally, I could not have done it without the support of my friends, T. Mukaro, T. Howera, T. Yeboah, E. Antonio and A. Chapani, who through their own will and kindness, provided well-wished distractions and helpful hands.

Contents

Abstract	i
Declaration	v
Acknowledgement	vi
List of Figures	xii
List of Tables	xiii
Abbreviations	xiv
1 INTRODUCTION	1
1.1 Project Overview	1
1.2 Motivation	2
1.3 Objectives	3
1.4 Thesis Structure	3
2 Literature Review	4
2.1 Overview of Natural Language Processing Techniques	4
2.1.1 Named Entity Recognition	4
2.1.2 Tokenisation	5
2.1.3 Stemming and Lemmatization	5

2.1.4	Bag of Words	7
2.1.5	Natural Language Generation	8
2.2	Natural Language Processing Steps	10
2.2.1	Morphological Analysis	10
2.2.2	Syntactic Analysis	11
2.2.3	Semantic Analysis	11
3	Existing Solutions	13
3.1	Solution One: Named Entity Recognition Approach	13
3.2	Solution Two: Automatic Speech Recognition for Iban	14
3.3	Solution Three	14
3.4	Solution Four: Dataset Creation and classification	15
3.5	Solution Review	15
3.5.1	Solution One	15
3.5.2	Solution Two	16
3.5.3	Solution Three	16
3.5.4	Solution Four	17
4	Methodology	18
4.1	Proposed Solution	19
4.2	Corpus Creation	20
4.2.1	Text Collection	20
4.2.2	Text Pre-processing	20
4.3	Text Input	21

4.4	Tokenisation	21
4.5	Term Frequency Inverse Document Frequency	22
4.6	Cosine Similarity	23
4.7	Similarity Report	24
5	Implementation	25
5.1	Corpus Creation	25
5.1.1	Text Pre-processing	26
5.2	Text Input and Tokenisation	30
5.2.1	HTML Website retrieval	31
5.2.2	Text document Retrieval	31
5.3	Term Frequency-Inverse Document Frequency	32
5.3.1	Normalised Term Frequency	32
5.3.2	Subject Document's Term Frequency	33
5.3.3	Inverse Document Frequency	33
5.3.4	Calculating Term Frequency-Inverse Document Frequency	34
5.4	Cosine Similarity	35
5.5	Results and Interpretation	36
5.5.1	Cosine Similarity with Source Articles as Subject Documents	36
5.5.2	Interpretation Approach 1	37
5.5.3	Interpretation Approach 2	40
5.5.4	Other Results	40
6	Conclusion and Future Work	44
6.1	Conclusion	44

6.2	Future Work	44
-----	-----------------------	----

List of Figures

Figure 2.1	Typical Bag of Words Creation Process	7
Figure 2.2	Bag of Words Example	8
Figure 2.3	Natural Language Processing Steps	10
Figure 4.1	The basic flow of a NLP system	18
Figure 4.2	Proposed Solution Using Bag of Words	19
Figure 5.1	Requests method to get website data	26
Figure 5.2	Beautiful Soup ".text" method	27
Figure 5.3	Syntaxt remover function	28
Figure 5.4	Removal of common English words	28
Figure 5.5	Similar Word remover Function	30
Figure 5.6	Subject Document Example	31
Figure 5.7	Choices Function	31
Figure 5.8	Normalised Term Frequency Function	32
Figure 5.9	Output from Normalised Term Frequency Function	33
Figure 5.10	Subject Document Term Frequency	33
Figure 5.11	Inverse Document Frequency Helper Function	34
Figure 5.12	Inverse Document Frequency Function	34
Figure 5.13	Article TFIDF Function	34
Figure 5.14	Corpora and Document TFIDF Function	35
Figure 5.15	Cosine Similarity Helper Function	35
Figure 5.16	Cosine Similarity Function	36
Figure 5.17	Similarity Output from Cosine Similarity Function	36
Figure 5.18	English document Results	40

Figure 5.19 Indonesian Results using Approach 1	43
Figure 5.20 Approach 2 Results	43

List of Tables

Table 4.1	Similarity Report	24
Table 5.1	Top 10 Iban Articles According to Word Frequency	26
Table 5.2	Top 10 Malay Documents According to Word Frequency	27
Table 5.3	Total Corpus Size	28
Table 5.4	Iban Word Frequency	29
Table 5.5	Malay Top 10 Word Frequency	29
Table 5.6	C.S with Iban Articles as Subject Documents Approach 2	38
Table 5.7	C.S with Malay Articles as Subject Documents Approach 2	39
Table 5.8	C.S with Malay Articles as Subject Documents	41
Table 5.9	C.S with Iban Articles as Subject Documents	42

Abbreviations

ASR	Automatic Speech Recognition
BOW	Bag Of Words
C.S	Cosine Similarity
CRL	Closely Related Languages
HRL	High Resource Languages
LRL	Low Resource Languages
MA	Morphological Analysis
NER	Named Entity Recognition
NLG	Natural Language Generation
NLTK	Natural Language Toolkit
NRL	Natural Language Processing
SA	Semantic Analysis
SA	Syntactic Analysis
TF-IDF	Term Frequency- Inverse Document Frequency

Chapter 1

INTRODUCTION

1.1 Project Overview

With technology becoming more and more deeply rooted in our day to day culture, there continues to be a growing chasm between the ever changing field of technological innovation and cultural preservation and relevance. As such, there is a concerning rise in the need to preserve certain aspects of society that can find their demise with a rise in technological advancement. One such area is spoken languages. For processing, the industry has moved leaps and bounds beyond what was thought to be possible, however the progress has mainly been in specific international languages such as English, French and Spanish which have a greater reach.

For Low Resource Languages (LRLs), we find that of the roughly 7 000 languages spoken on earth [1], only 20 are considered High Resource Languages (HRL). This means of all the wonderful things that technology has done, there hasn't been a deep enough impact to preserve other languages. This can be easily attributed to the geography of most technological breakthroughs where if an advancement is made in country A, then there is a high chance that the advancement will be published in languages prevalent and spoken in said country. However, with the

power of Natural Language Processing there are now measures to mitigate loss of languages. This not only results in preservation of LRLs but improves the chances of creating relevant and effective educational material, knowledge expansion and even emergency response to name a few [2].

The biggest drawback in working with LRLs is the fact that there are little resources available to build a corpora of words to fully study the languages. As such, issues may arise when collecting resources to fill up said corpora. And once these languages stop being taught in schools due to their lack of official use they risk going extinct. Hence the need to study and create methods and measures that save LRLs.

This study will make use of Malay and Iban languages as a means to add to the work already being done towards creating databases for LRLs. In this case Iban is the low resource with over 700,000 native speakers in Borneo with Malay having over 290 million speakers. The end goal of the project is to create a program/system that can process a given piece of text or document and determine, through various means and processes, whether its in Malay or Iban[3].

1.2 Motivation

Natural Language Processing (NLP) has given a lifeline to LRLs which may be at the brink of extinction. As such, in order to capitalise on the available technology, NLP methods will be applied to Iban and Malay so as to create a program that can tell whether or not a given word is in Malay or Iban or both.

The similarities between the two languages are evident yet some words when spelled the same have totally different meanings. The purpose of this project will be to try and build a program that can clearly differentiate between the two languages.

1.3 Objectives

This research project has objectives to:

1. form a corpus of documents in the Iban Language
2. study the characteristics of Iban Language for text processing
3. propose an approach to identify the language used in text for LRL, with Iban as a case study.

1.4 Thesis Structure

The remainder of the document will be structured as follows; Chapter 2 will go in-depth of Literature Review touching on the differences and similarities between Malay and Iban Languages. It will also give an overview of the various steps that a basic NLP System should go through before a final version is made.

Chapter 3 will look into explaining some of the current approaches that have been used to save LRLs in the context of not only Iban but other languages too. This chapter will be heavily focused on the advantages and disadvantages of each approach and solution. Though solutions are limited in the case of Malay and Iban a handful of solutions are available for other languages. From these solutions a lot can be on the feasibility and limitations of using an NLP on Low Resource languages.

Chapter 4 shows the chosen methodologies used to study to create the final system and the types of methods used on the two languages. It will also touch on the importance of the chosen approach and why some steps have to be repeated.

Chapter 5 will look into how the methods discussed in the previous chapter are put into play. This will also look at the issues faced whilst implementing the chosen methods.

Chapter 2

Literature Review

2.1 Overview of Natural Language Processing Techniques

There are multiple processes used in NLP.

2.1.1 Named Entity Recognition

Named Entity Recognition (NER) is identifying Named Entities such as names and places from a given piece of text [4]. This applies in most cases where a response is required by a user. Examples of the uses of NER are in semantic annotation, news categorization and customer support.

With respect to LRLs, NER is useful when it comes to customer service support where, for one reason or the other, a customer might be able to only communicate in one language (Iban) yet the customer service agent can only converse in Malay. With the use of NER, there is reduced ambiguity in what a certain word means as the customer services agent can easily get the context in which the customer is using certain words.

Such a situation can also help in increasing translation accuracy of websites and translation engines when languages are closely related as in the case of Malay and Iban. Some websites such as A and B would treat text in Iban as if its Malay. This leads to greater confusion in terms of how the reader will understand any content being displayed.

2.1.2 Tokenisation

Tokenisation involves the splitting of raw text to sentences, words or characters depending in the use case needed [5]. This allows any NLP to learn the language by breaking it down to tokens that can be analysed further in order or improve accuracy of translation or use. The tokenisation process is used to remove non-meaningful structures such as punctuation marks, brackets and hyphens.

However, this approach introduces errors by not taking into consideration multi word elements such as people's names and city names. This error is one that might result in an inadequate corpus model being made and introduces in the NLP pipeline. Such elements/tokens require special treatment in order to safeguard the authenticity of the project.

Depending on the source of the text, it may contain abbreviations and acronyms that need to be substituted in full. This can present an issue when the tokenising process is started. A filter can be used to catch such instances but it may not filter out all acronyms and abbreviations.

2.1.3 Stemming and Lemmatization

Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem [6]. Stemming would mean if given a words like "missing", "misses" and "missed" the stem word would be "miss". These computational procedures are usually based on a specific language, for

example; English has Porter and Lancaster stems both of which have differing approaches to the stemming process.

For non-English languages, Python has a Natural Language Toolkit (NLTK) library which contains a Snowball Stemmer. Snowball Stemmer can be used to generate rules for up to thirteen languages depending on the use case as defined by the use [7].

Lemmatization is the process of grouping inflected forms of a word together in order for the to be analysed as a single entity using the words dictionary form (lemma) [8]. For example, such an approach would link "gone", "going" and "went" to "go". This process works well for HRLs as there is a wider base of discovery to build the languages corpus on thus increasing accuracy of the resulting system.

Applications

Stemming and Lemmatization can be used in:

1. Sentimental Analysis
2. Document Clustering
3. Information Retrieval

2.1.4 Bag of Words

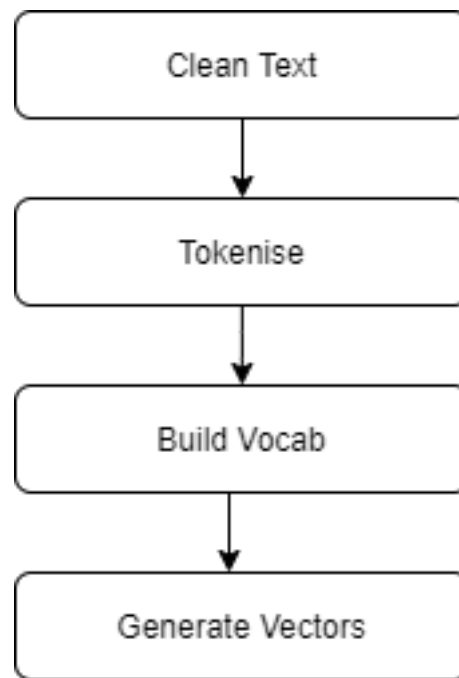


Figure 2.1: Typical Bag of Words Creation Process

The Bag Of Words(BOW) technique is used to simplify natural text or language by removing grammar and word order yet leaving the number of appearances that each word makes. Given text to work with, this method will only count the number of times words appear in the text and then apply various operations depending on the user's.

Given a sentence:

"Tamia has a basketball. Ivan wants a basketball too."

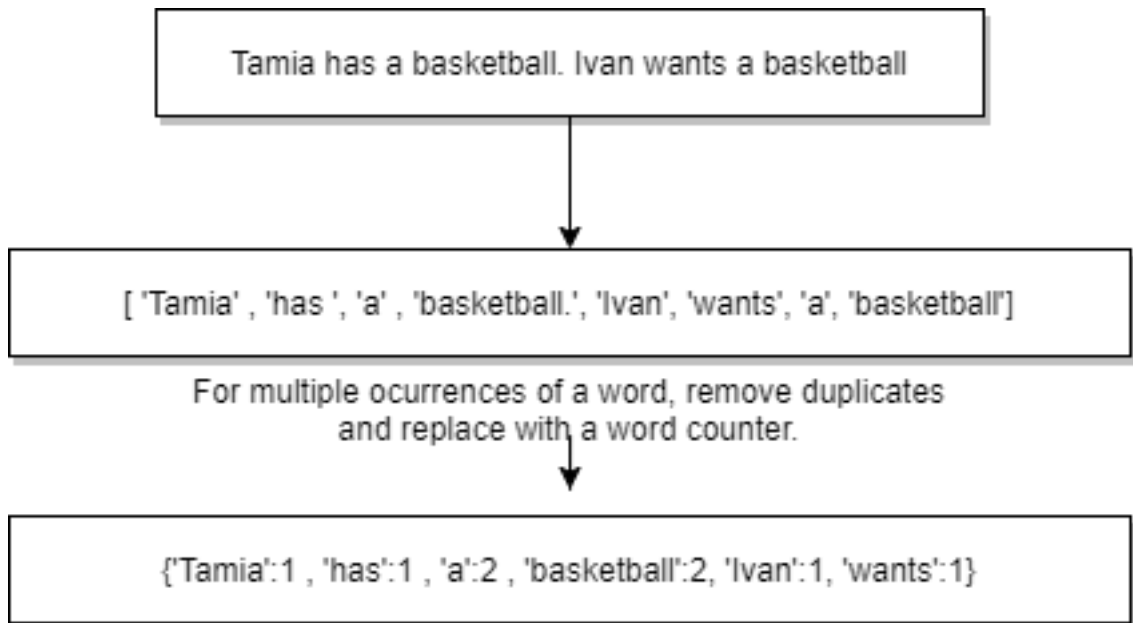


Figure 2.2: Bag of Words Example

The vocabulary of the whole document will be used to create vectors for each sentence, with each vector length being equal to the vocabulary size. Word co-occurrence or word order is not taken into account in the BOW method [9]. However the BOW model can be tailored to take into account pre-determined word order. Such an approach is currently being implemented by WordNet. Another step in the BOW approach is to filter out stock stock words. these are words that appear the most in a language but do not have actually bearing or meaning to the actual words. This step in itself might be skipped if in the case of Iban and Malay the use of filter words can be the determining words of whether or not its Malay or Iban.

For example, word co-occurrence can be seen when using the term "Kuala Lumpur"; originally this phrase can be classified as 2 different words. However with tailoring, the program can be taught to expect "Lumpur" soon after "Kuala".

2.1.5 Natural Language Generation

Natural Language Generation (NLG) is the process of generating structured text

or language from structured Data, This method is often employed in generating reports for companies for clearer human understanding and easier summarisation of key points from a given data. It is used when mainly working with Big Data or readily employed customer service systems including virtual assistants such as Siri or Google Assistant.

The use of either a dictionary based approach or a corpus based approach is wholly left to what suits the user better. A dictionary based approach would mean making use of online resources such as WordNEt and Merriam Webster. This approach provides a more accurate representation of word semantics especially is semantics are of vital importance in the end program. The corpus based would depend on the original corpus created for the specific program with little to no interaction with an online dictionary.

For the purposes of this project a hybrid solution will be used where a general corpus will be created and studied then if there's need to consult the online dictionary then only will it be included. Reason being, for Iban there's little to no proper dictionary source such as WordNet and for Malay a WordNet database does exist. However, due to the languages being closely related, some Iban words may be included in WordNet Bahasa as part of the Malay language due to the lack of proper distinction.

2.2 Natural Language Processing Steps

2.2.1 Morphological Analysis

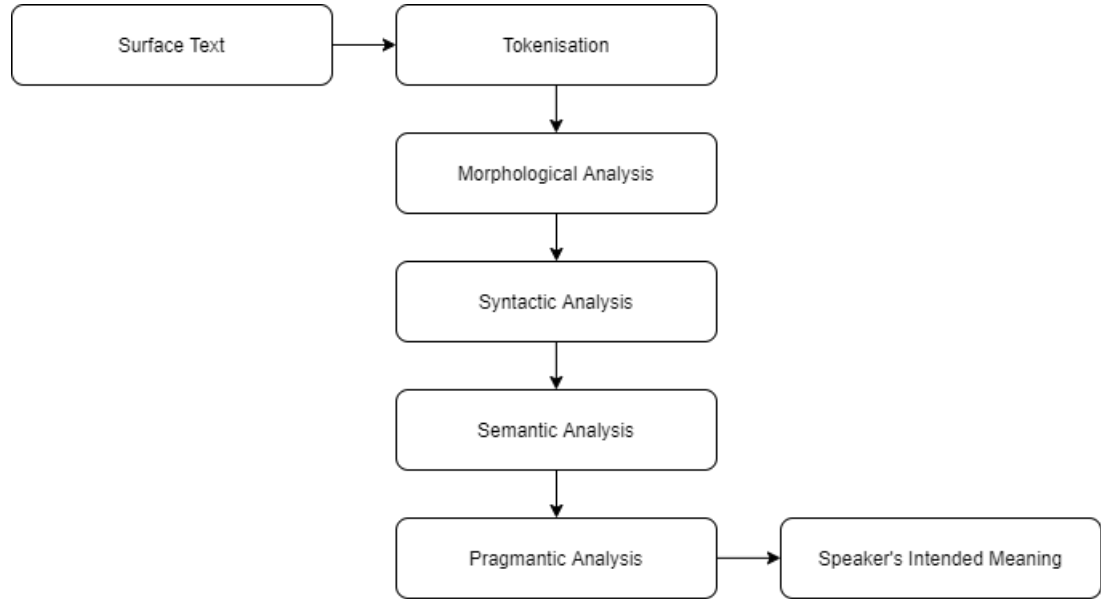


Figure 2.3: Natural Language Processing Steps
[10]

Morphological Analysis (MA) is the process of giving information pertaining to the grammar of a word with regards to the words morpheme [11]. A morpheme is the smallest meaningful part of a word in a given language [12].

For Malay and Iban also, morphemes are either free or bound. The former meaning it can appear on its own as a word with the latter meaning it cannot stand on its own as a word [12], [13]. An for example of a bound morpheme in Malay is the "ber" in "bernyala" which means to burn. An identical morpheme also exists in Iban but instead of "ber" it becomes "be" [13].

The similarity of the two languages goes beyond just the aforementioned morpheme. There is also a visible similarity in the usage of nominal prefixes such as "pe", "se" and "ke". However they differ in what nouns are derived from the prefixes. Using "pe", in Iban it derives human, concrete and abstract nouns whilst in Malay, the prefix derives the first two sub-classes [13].

Malay and Iban are also separated by the number of suffixes present in each language [13]. For Iban there is only one suffix which is "-ka" and Malay has three suffixes namely; "-an", "-kan" and "-i"

These differences have to be fed into the resulting system in order for the system to find which language a given piece of text is in.

2.2.2 Syntactic Analysis

Syntactic Analysis (SA) is the analysis of a sentence or given piece of text with regards to its logical meaning. It can also be called parsing [10]. For the purposes of this project, it will mainly focus on the word construction itself since the main goal is to determine whether a given word or piece of text is in Malay or Iban.

SA often employs the n-gram-based model to determine various differences and similarities between languages. The n-gram model is defined as contiguous grouping of n things from a given example of text or discourse. These can be letters, words, syllables or even phonemes depending on the application. This n-gram model can be difficult or tricky to use once the languages are said to be similar as in the case of Malay and Iban. A study carried out on two closely related languages, Malay and Indonesian [14], reiterated this.

2.2.3 Semantic Analysis

Semantic Analysis (SA) is the process of extracting the meaning of text structures with relation to the text's context. The process entails taking into account the dictionary meaning of given words, phrases or pieces of text. Each word is analysed as a singular item and then as a part of the whole given data. It can also be described as a measure or analysis of a term or piece of text is from being negative or positive [15]. This is a broad generalisation that carries over the basis on which SA is established.

There are various elements to the SA, including; hyponymy, homonymy and polysemy. Hyponymy is the relationship between a term and its instances. For example, the term 'colour' (often called a hypernym) and the colour red, brown or green become its hyponyms (the relation).

Homonymy is when words have the same spelling or same form but have different and unrelated meaning. An example is address which can mean to speak to or a location.

Polysemy is almost the same as a homonymy except that a polysemy has a related meaning to the given word.

Semantic Analysis takes into account a number of building blocks such as, Entities, Concepts, Relations and Predicates. Entities represent a particular person, object, location etc. Concepts are the category in which the entities fall such as city, family or group. Relations are simply the connection between the entities and concepts. Predicates are then the verb structures that play a part in the given sentence or piece of text.

Due to the nature and expected results of this project, Semantic Analysis would not play a critical role in determining the outcome of the results. This is mainly because of the amount of time it would take to develop a proper Semantic library for both Iban and Malay. Also, the requirements of this project mainly focus on determining the language of a given piece of text instead of the meaning of it.

Chapter 3

Existing Solutions

3.1 Solution One: Named Entity Recognition Approach

The approach used by Rayner Alfred [16] was that of using NER based on a Rule-Based Approach. This was done mainly to cater for the Malay language alone without taking into account Iban. This can be considered a solution as it can be used to further review the characteristics of the Malay language and how it appears in text form especially in the NER approach.

However it can be problematic when it comes to telling the difference between Malay and Iban as the NER approach did not take into account the differences between the two languages specifically. As such, one might find that words that are natively Iban may be included in the Approach as Malay words due to the closely relatedness between the two languages. For this project, such a difference is critical to the success of the system.

Some key takeaways from this study would be the use of the NER method to create a corpus of the Malay language which can be helpful in identifying given Malay terms.

3.2 Solution Two: Automatic Speech Recognition for Iban

[17]

This solution is based on work done by [17] the Faculty of Computer Science and Information Technology at the University of Grenoble Alpes. They worked with both Malay and Iban with their focus mainly on Automatic Speech recognition for Iban. Malay was used in a 'reference' capacity due to Iban being a LRL.

Some of the issues faced with implementing the solution was that since their aim wasn't to know the language of the origin via text, they had a smaller corpus to work with when it came to Iban. The pronunciations in both languages can be very similar as such Malay was used to obtain some of the needed information.

The difference between the aims of this solution and the current project are that the case study was done without much regard to the deciphering of whether given information is in Malay or Iban. This project has its main aim being deciphering whether or not the text is in one or the other language.

However this can help to identify key areas that Malay and Iban are similar thereby clearing up the borderline between the two closely related languages.

3.3 Solution Three

[18]

The study carried out by B.Rainaivo-Malancon in 2006 at Universiti Sains Malaysia makes use of Malay and Indonesian to study the automatic identification of Closely Related Languages . Their aim was to build a language identifier to determine whether a text is written in Malay or Indonesian.

In this case they made use of trigrams of characters, the lists of exclusive words and number formatting to determine the overall language that the text is in. This approach made use of lexical analysis also taking note of the gap that exists with most language identifiers where they classify closely related languages as one language.

3.4 Solution Four: Dataset Creation and classification

Marivate, Sefara, Chabalala [19] made use of two similar languages found in South Africa namely; Setswana and Sepedi using a two step text augmentation method as to achieve an improved models of classification for Setswana and Sepedi. They also noted that the use of CRLs is pivotal in how corpora for LRL's can be furnished.

Here, CRLs are defined as languages that descended from each other or if both have the same ancestor. In the case of this project, the latter definition would apply where Malay and Iban have the same ancestor; the Proto-Malayo-Polynesian language.

Their approach was aimed at classifying headlines in Setswana and Sepedi using data augmentation. Data augmentation can be defined as the augmentation of a training set with artificially generated and trained sets. Such an approach can be seen mostly in the neural networks.

3.5 Solution Review

3.5.1 Solution One

The first approach by Rayner Alfred [16] is an approach that works if we take the whole umbrella term of Malay Languages without breaking it down into its

constituent languages, hence it lacks the specificity of identify Iban and Malay separately.

However the approach can be a starting point in itself as they make note of the importance of named entities, that is, the difference of naming common entities such as a crocodile; which is called "baya" in Iban and "buaya" in Malay can be used in the resulting approach.

3.5.2 Solution Two

In terms of the Second Solution; ASR for Iban, it goes beyond the scope of this project as its focus is on the varying speech patterns in Iban whilst using Malay as a reference language. This meant they used the similarities between the two languages to their advantage as compared to the identifying specifically the differences between the two languages.

However, the referencing of Malay in their work bears testament to the closely relatedness of Iban and Malay themselves.

3.5.3 Solution Three

This solution solves the gap that is often missed by most language identifiers as they group similar or closely related languages instead of finding the differences in their lexical analysis.

The resulting use of trigrams and number formatting allowed for a greater degree of accuracy in language identification, especially in the case of Malay and Indonesian which are closely related languages. A key takeaway would be the use of exclusive words to identify the language which can be used in the proposed bag of words approach.

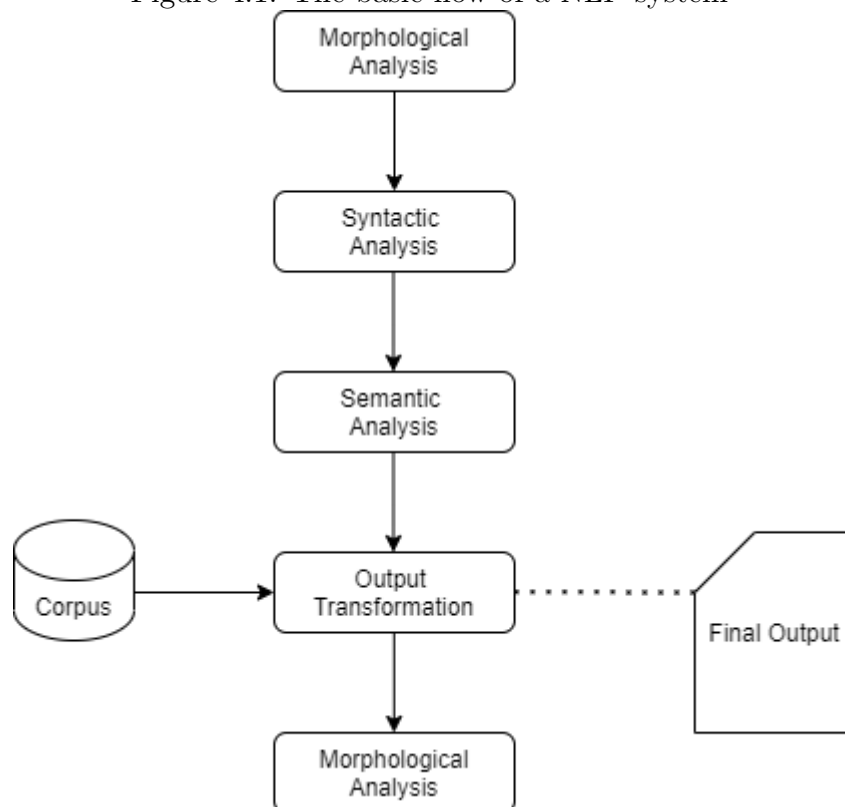
3.5.4 Solution Four

Marivate, Sefara, Chabalala [19]' approach is closely related to the proposed solution in this project, however the methodology of using data augmentation to populate the corpora is not viable in this case given the time frame of the project. Since their aim was to use CRLs to come to a conclusion, this can then be used in the project at hand.

Chapter 4

Methodology

Figure 4.1: The basic flow of a NLP system



The figure above shows a rudimentary approach to Natural Language Processing. This covers the grounds of not only language identification but that of language interpretation too.

However, the processes of making Morphological and Syntactic analysis are often tedious and require an indepth understanding of both languages' structures and the differences between them. As is the case for Iban and Malay, applying a full on linguistical study of the languages would take more time than that made available.

In turn, the proposed solution is using the Bag of words approach coupled with Cosine Similarity and Term Frequency- Inverse Document Frequency. This reduces the overall understanding need of the 2 languages and can make use of publicly available data such as articles, books and any written text that's publicly available.

4.1 Proposed Solution

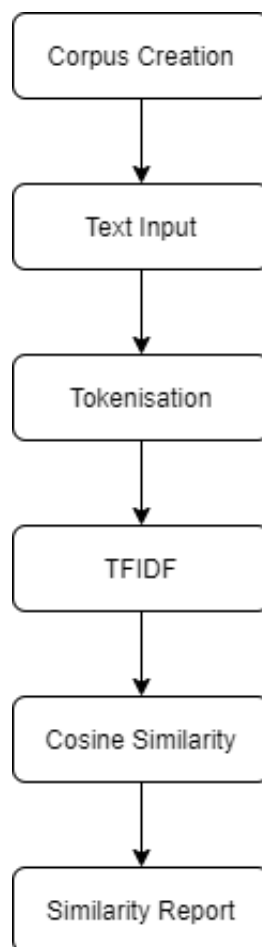


Figure 4.2: Proposed Solution Using Bag of Words

The six steps above combine all the processes that will need to take place for the program to be able to identify whether the text is in Iban, Malay or none of the two.

4.2 Corpus Creation

4.2.1 Text Collection

For the text collection, articles were obtained from the Utusan Borneo website [20]. Articles topics were mainly news articles with a direct translation from Malay to Iban. A total of 30 articles were obtained. Each article had two versions; one in Malay and another in Iban. Article topics ranged from Current Affairs to general entertainment.

The content and context of the data is import if the Named Entity Recognition technique is employed. This insures that whenever a person is named in one article, the nature of their title can be easily recognised and analysed.

Utusan Borneo was a good source in that the language used and the overall structure of the HTML would suit the purposes of the project. It also provided a good source of the articles that were direct or close to direct translations of each other, hence the choosing article pairs with similar topics.

4.2.2 Text Pre-processing

The articles were originally downloaded in PDF format but this presented problems when it came to text extraction as HTML is not fine tuned to be saved in the PDF format. As such, the HTML webpages had to either be downloaded as is or accessed by a Python webscrapper library.

The library used is BeautifulSoup4 [21]. This permits accessing live data on websites to study and extract the text elements(HTML level) of a website. Once

text is extracted, HTML or CSS tags have to be removed including any excess data such as images or excess links to external websites. This leaves the text that is required.

For the Utusan Borneo website, the website had a basic structure to how they formatted their articles. This meant a single program could be used to extract all the text from a given set of webpages. The resulting words are saved in two separate files; one that saves each unique word and adds it to the relevant corpus, that is a Malay word to the Malay corpus and an Iban word to the Iban corpus.

Corpus creation only takes place once during the initial run of the system.

4.3 Text Input

This part involves the inputting of the subject document or text that's under review. If it is a link to a website it would go through the website and run it through the text pre-processing stage above to remove HTML code, CSS and any other non words parts out.

If the review document is in PDF format it will be reproduced and taken through a different text preprocessing that results in the same list of words.

Once the words (tokens) have been made, they will be run through a text processing method that removes punctuation marks, quotation marks and makes all the words into lower case.

4.4 Tokenisation

Tokenisation would be the issuing of an "Id" for each word in the now inputted text so as to allow for easier manipulation of the data. Each text document is then broken down into individual words disregarding the punctuation and grammar as

this is not a sentiment analysis system. After the breakdown, it becomes easier for the Term Frequency function to take effect.

4.5 Term Frequency Inverse Document Frequency

Term Frequency is the process of quantifying a word in the given documents and then computing the weight of said word in relation to the two corpora and the review document itself and then assigning a weight to represent that word. As such, the relative importance of that word is then signified by the weight it has [22]. This means for the review document it is necessary to not remove duplicate words unless if they appear in both languages.

$$TermFrequency, tf = \frac{wordcountindocument}{totalofwordsindocument} \quad (4.1)$$

Inverse Document Frequency (IDF) indicates the informativeness of a given word or term in a document [22]. Ideally, the most frequent words that appear in the document and corpus end up having a lower Inverse Document Frequency. This works in eliminating the effect of stop words in the English language, and in a language such as Iban and Malay where due to their relative similarity; some words may be similar, it reduces the amplification of these words and only amplifies the ones that are distinctively different. Another use of this is considering that Iban has no Natural Language Toolkit to remove stopwords, this gives us a rough estimation of what words constitute stopwords in the language.

$$IDF(word) = \log\left(\frac{countofcorpus}{numberofwordsindocument + 1}\right) \quad (4.2)$$

The final version of the Term Frequency-Inverse Document Frequency equation combines both the Term Frequency equation and the Inverse Document Fre-

quency multiplicatively;

$$TF - IDF(w, d) = TF(w, d) * \log\left(\frac{C}{DF + 1}\right) \quad (4.3)$$

where,

- w - word
- d - document
- TF - Term Frequency
- DF - Document Frequency
- IDF - Inverse Document Frequency
- C - Count of Corpus

4.6 Cosine Similarity

Cosine Similarity is a form of document similarity measurement using the cosine equation, the equation used to measure an angle in multi-dimensional space [23].

$$\cos \theta = \frac{a.b}{||a||||b||} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2} \sqrt{\sum_i^n c_i^2}} \quad (4.4)$$

In said multi-dimensional space, each dimension represents a word in the document. The C.S then measures the angle of documents to check the similarity. As compared to a more traditional Euclidean distance, where word frequency is counted and then used as a measure of similarity. The flaw with the Euclidean distance is it would be affected by the frequency of individual words meaning if

one word appears numerical more times in a document than in another document, the similarity result would be skewed.

Cosine Similarity does not take into account word meaning or sentimental analysis hence its application across different languages is not affected.

4.7 Similarity Report

The overall similarity report will be based on the result of the Cosine Similarity check and will be presented in a list/table format. Each numerical value will range from 0 to 1.0 with a +/- 0.000001 discretion.

Table 4.1: Similarity Report

Document Title		
Original Document	Malay Similarity	Iban Similarity
1.0	0.459567	0.022314

The overall criteria on deciding whether or not a document is in Malay or Iban is by measuring the difference between the two languages similarities. For a higher similarity, the program would require more time to determine the language.

Chapter 5

Implementation

5.1 Corpus Creation

Corpus Creation is the most vital part of the system as it would determine the validity and accuracy of the results of the system. Two corpora would be created, one in Malay and another in Iban. The text was taken from the Utusan Borneo website with each article subject having a close or direct translation in Malay and Iban. This reduced the overall spread of topics to be used and offered a more focused approach to the corpus creation process.

The tables 5.1 and 5.2 shows the top 10 documents arranged in terms of descending number of unique words relative to each document. The term unique words does not take into account the frequency of each word in relation to the other documents.

Table 5.1: Top 10 Iban Articles According to Word Frequency

Iban		
Document Title	Publish Date	Unique words
PKPP enti dilanjar meri empas ngagai pengerai, ekonomi	2020/08/26	619
Tentuka penyakal enda ngeruga projek pemansang ke pengelantang rayat	2020/09/14	612
TTUG keterubah di Sarawak deka digaga di Sibui: Fatimah	2020/09/03	610
KM seruran ngarapka SUPP, PSB bebaik	2020/09/06	604
Projek pemansang ti dipejalaika GPS ngayanka perintah ngemeratka peranak menua pesisir	2020/09/14	592
Bantu belanja ke diberi perintah patut dikena ngemansangka rumah panjai	2020/09/01	578
14 rumah jalai pengarap, Gempung jalai pengarap di Layan nermana bantu ari UNIFOR	2020/09/12	573
Uras ari darat ngamahka sungai, pantai Miri: Lee	2020/09/13	539
Pengajar tau dianggap injinir rama ti nempa modal mensia	2020/08/28	521
Rebak biak diperansang ngaul diri ba pengawa betanam betupi moden	2020/08/26	488

5.1.1 Text Pre-processing

Running an automated python script that accesses a list of pre-chosen articles using the BeautifulSoup library [24] the resulting document is a HTML file that contains all the data relating to the webpage that contains the article.

```
for line in corpus:
    #print(line)
    # driver.get(line[3:])
    driver = requests.get(line[start_point:-1])
    #contents = driver.page_source
```

Figure 5.1: Requests method to get website data

By using a a method available in the library; ".text", to find the article text inclusive of any other website text that may be available on the site without

Table 5.2: Top 10 Malay Documents According to Word Frequency

Malay		
Document Title	Publish Date	Unique words
Lanjutan tempoh PKPP beri kesan positif kepada kesihatan rakyat, ekonomi dijangka mengucup - Penganalisis	2020/08/25	574
Saya teruskan usaha perdamaian SUPP, PSB untuk kebaikan komuniti Cina - Abang Johari	2020/09/08	562
Projek pembangunan luar bandar bukti kesungguhan GPS	2020/09/14	553
Usah terpengaruh agenda untuk menjatuhkan kerajaan	2020/09/14	547
Guna peruntukan naik taraf rumah panjang sebaiknya	2020/09/01	541
Sampah dari darat cemarkan sungai, pantai Miri	2020/09/13	523
Projek pembangunan luar bandar bukti kesungguhan GPS	2020/09/14	512
Jumlah kes denggi di daerah Selangau naik mendadak 213 peratus: Sempurai	2020/08/30	510
Joshua Ting baiki jeti usang Rumah Dawi Ringgil	2020/08/26	492
Rumah MAKSAK Sibul dijadikan tempat transit gelandangan	2020/09/02	470

including HTML tags or scripts.

```
# print(contents)
soup = BeautifulSoup(driver.content, 'lxml').text
# print(soup)
```

Figure 5.2: BeautifulSoup ".text" method

Once the text has been retrieved, the next phase of text processing begins which is the cleaning phase. Cleaning removes punctuation, numbers or digits, special characters that do not contribute to the structure of the word and removes English words in the case of words such as Facebook, Google or Twitter than can be use as hyperlinks to the the website's social media.

Removal of the English words is only subjective to the source of corpus hence the need for it when using a website as a source.

```
def syntax_remover(original_text):

    final_list = []
    # print(original_text)
    for word in original_text:
        if word is string.punctuation or word.isdigit():
            print(word)
        else:
            final_list.append(word)

    return final_list
```

Figure 5.3: Syntact remover function

```
unwanted_words = ['Online', 'Advertisement', 'Toggle',
                  'navigation', '◆', 'Facebook', 'Twitter', 'Email', 'Print', 'Google+', 'Google']

# adds all words to one list to prep for processing
for filename in os.listdir(language_folder):
    new_path = language_folder+"\\ "+filename
    if filename.endswith(".txt"):
        open_file = open(new_path, "r")
        for line in open_file:
            if line not in language_dict and line not in unwanted_words:
                language_dict.append(line[:-1])
        print(len(language_dict))
        list_of_words[filename] = len(language_dict)

    language_dict = []

    open_file.close()
```

Figure 5.4: Removal of common English words

In total; 175,772 words were collected to make up the Malay corpus and 158,163 for Iban. Of all the words, only 1,388 and 1,859 words for Iban and Malay respectively were unique. This left a usable corpus of the above numbers representing words found in the two languages. This is evident in the table 5.3.

Table 5.3: Total Corpus Size

Language	Total Words	Unique words
Malay	158163	1859
Iban	175772	2107

Table 5.4 and Table 5.5 show the overall distribution of the top ten most frequent words in each language. This distribution can go to show how much weight will be assigned to each word later on during Cosine Similarity.

Table 5.4: Iban Word Frequency

Iban	
Word	Frequency
enggau	5119
ke	3894
nya	3364
iya	2861
Sarawak	2152
deka	1702
ari	1588
ku	1483
Borneo	1473
sereta	1465

Table 5.5: Malay Top 10 Word Frequency

Malay	
Word	Frequency
Sarawak	2924
untuk	2068
ini	1678
Borneo	1466
dalam	1458
Utusan*	1404
itu	1213
tidak	1108
kepada	1094
ke	1063

Once the frequencies are calculated, the next phase is to compare the number of similar words between Malay and Iban with the aim being to reduce the amount of similarity between the 2 languages. Considering that the subject languages are similar, there's a need to reduce the similarity by amplifying the differences that are evident in the languages. One way of doing this especially in the bag of words approach is to remove any words that appear in both Malay and Iban without regarding the meaning of the words as there is no lexical or sentimental analysis to be carried out.

Figure 5.5 shows the similar word remover function. The function produces three

Comma Separated Value files that store the non-similar words in Malay and Iban in two separate files and one file that has all the similar words. The Similar words file will be used down the line by the article under review. This would conclude the Corpus creation.

```
for i in range(dict1_len):
    for y in range(dict_len):
        if i < dict1_len and y < dict_len:
            if final_dict1[i] == final_dict[y]:
                similar_words.append(final_dict[i])
                del final_dict[y]
                dict_len -= 1
                del final_dict1[i]
                dict1_len -= 1

final_corpus = open(filename[:-10] + "unique_corpus.csv", "w")

for item in final_dict1:
    final_corpus.write(item)

final_corpus2 = open(filename2[:-10] + "unique_corpus.csv", "w")

for item in final_dict:
    final_corpus2.write(item)

similar_corpus = open("Similar_words.csv", "w")
for item in similar_words:
    similar_corpus.write(item)
```

Figure 5.5: Similar Word remover Function

The final output from this stage are as follows:

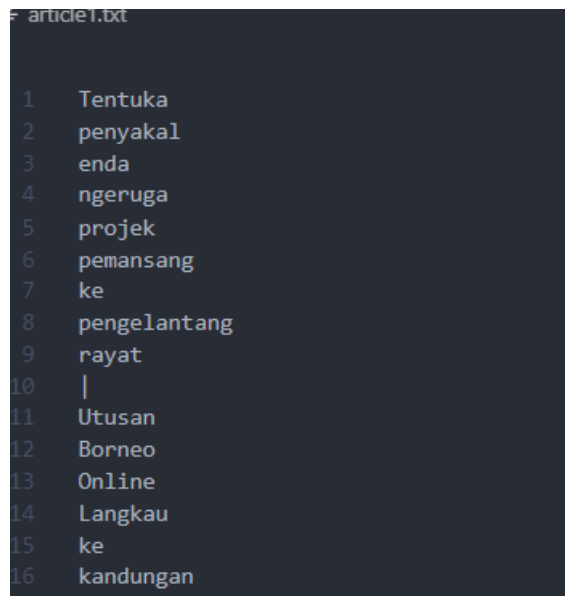
- Malay Corpus without similar words removed
- Iban Corpus without similar words removed
- Malay Corpus with similar words removed
- Iban Corpus with similar words removed
- File with all Similar words obtained in Malay and Iban

5.2 Text Input and Tokenisation

This section concerns the subject document which needs to find the language.

5.2.1 HTML Website retrieval

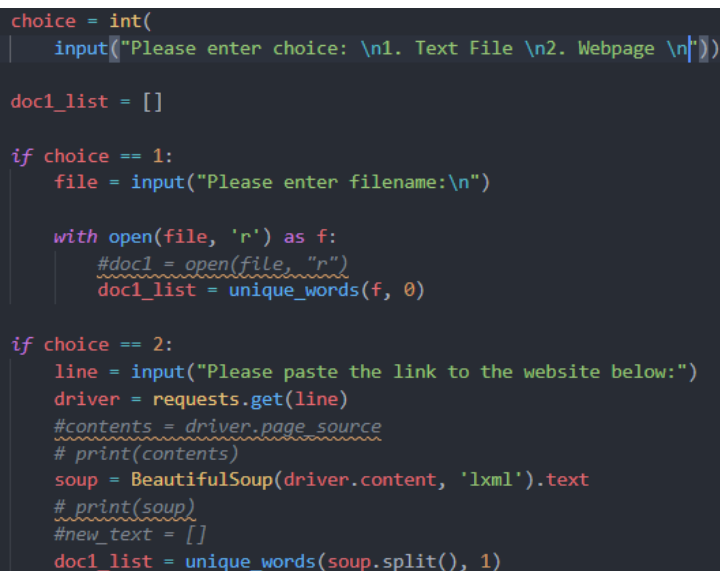
When supplied with a link to an article or document, the downloaded data will be run through the same process as the corpus to end up producing a text file containing a list of words obtained from the article. The final version of the article would look like 5.6



```
article1.txt
1  Tentuka
2  penyakal
3  enda
4  ngeruga
5  projek
6  pemansang
7  ke
8  pengelantang
9  rayat
10 |
11 Utusan
12 Borneo
13 Online
14 Langkau
15 ke
16 kandungan
```

Figure 5.6: Subject Document Example

5.2.2 Text document Retrieval



```
choice = int(
    input("Please enter choice: \n1. Text File \n2. Webpage \n"))

doc1_list = []

if choice == 1:
    file = input("Please enter filename:\n")

    with open(file, 'r') as f:
        #doc1 = open(file, "r")
        doc1_list = unique_words(f, 0)

if choice == 2:
    line = input("Please paste the link to the website below:")
    driver = requests.get(line)
    #contents = driver.page_source
    # print(contents)
    soup = BeautifulSoup(driver.content, 'lxml').text
    # print(soup)
    #new_text = []
    doc1_list = unique_words(soup.split(), 1)
```

Figure 5.7: Choices Function

When provided with a text file name, the program opens and inserts each word as a single entity to a list. The list is then edited to remove punctuation and make all words lowercase to allow for easier similarity checks.

5.3 Term Frequency-Inverse Document Frequency

5.3.1 Normalised Term Frequency

```
def compute_normalisedtf(article):
    # computes normalised tf of a term relative to the corpus and document itself
    # returns a list of term dictionary objects (tf_art)
    tf_art = []
    #tf_cor = []

    for doc in article:
        art_norm_tf = dict.fromkeys(set(doc), 0)
        for word in doc:
            art_norm_tf[word] = term_frequency(word, doc)

        # print(art_norm_tf)
        tf_art.append(art_norm_tf)

        idx = 0
        new_col = ["Normalised TF"]
        art_df = panda.DataFrame([art_norm_tf])
        art_df.insert(loc=idx, column='Document', value=new_col)
        # print(art_df)

    return tf_art
```

Figure 5.8: Normalised Term Frequency Function

Normalised Term Frequency is the frequency of a term or word relative to its origin document and the Malay and Iban Corpus. The normalised term frequency is calculated by using the equation;

$$TermFrequency, tf = \frac{\sum t}{N} \quad (5.1)$$

This creates 3 dictionaries object of size N ; where N is the size or number of words in the subject document. The first dictionary contains all the document terms t as keys matched to their term frequency in the subject document as shown here (5.9). The second and third dictionaries contains all the subject document's words as keys matched to their frequency in the Malay and Iban language respectively. This data is then presented as a dataframe using the pandas library for Python [25].

doc	tentuka	penyakal	enda	ngeruga	projek	pemansang	ke	...	twitter	google	email	print	suka	@	copyright	2021
0	0	0.006890	0.018372	0.009186	0.006890	0.011438	0.016076	0.024510	...	0.003429	0.001634	0.001634	0.003429	0.001634	0.003429	0.003429
1	1	0.000744	0.000744	0.001487	0.000744	0.001058	0.001487	0.001058	...	0.000000	0.000529	0.000529	0.000000	0.000529	0.000000	0.000000
2	2	0.000000	0.000000	0.000000	0.000000	0.000847	0.000000	0.000847	...	0.000000	0.000424	0.000424	0.000000	0.000424	0.000000	0.000000

Figure 5.9: Output from Normalised Term Frequency Function

5.3.2 Subject Document's Term Frequency

```
def get_article_tf(article):
    article_tf = {}

    for word in article:
        article_tf[word] = term_frequency(word, article)

    return article_tf
```

Figure 5.10: Subject Document Term Frequency

For the purposes of calculating the Cosine Similarity, there needs to be the calculation of the original subject document's term frequency relative to itself. This is done by the *get article tf* function (5.10). This returns a dictionary value matching each term in the subject document to the relevant term frequency.

5.3.3 Inverse Document Frequency

To calculate the Inverse Document Frequency, the subject document, Malay and Iban corpus are run through the function with the following equation applied;

$$IDF(word) = \log\left(\frac{countofcorpus}{numberofwordsindocument + 1}\right) \quad (5.2)$$

The resulting effect is that of amplifying the values with a lower term frequency and the values are written to a dictionary matching the term as a key to its resulting inverse frequency relative to the corpora and the document itself.

For the Cosine Similarity function to work, there needs to be two distinct data sets of the Inverse Document Frequency, one relative to the subject document itself without including the corpus and another inclusive of the subject document and both corpora.

```
def compute_idf(docs):
    idf_dict = {}
    for document in docs:
        for term in document:
            idf_dict[term] = idf(term, docs)
    return idf_dict
```

Figure 5.11: Inverse Document Frequency Helper Function

```
def idf(term, docs):
    term_in_docs = 0
    # print(len(docs))

    for doc in range(0, len(docs)):
        if term in docs[doc]:
            term_in_docs = term_in_docs + 1

    if term_in_docs > 0:
        return 1.0 + math.log(float(len(docs)) / term_in_docs)
    else:
        return 1.0
```

Figure 5.12: Inverse Document Frequency Function

5.3.4 Calculating Term Frequency-Inverse Document Frequency

Similar to Inverse Document Frequency, there's need for two values of the TFIDF; one for the subject document shown by Fig.5.13 and one that is inclusive of both corpora and subject document shown here by Fig.5.14. The function returns a panda dataframe that will be used in the next step of calculating the cosine similarity.

```
def get_article_tfidf(article, art_tf, art_idf):
    tfidf_dict_art = {}
    for word in article:
        tfidf_dict_art[word] = art_tf[word] * art_idf[word]

    return tfidf_dict_art
```

Figure 5.13: Article TFIDF Function

```

def get_tfidf(documents, article, tf_doc, idf_dict):
    tf_idf = []
    index = 0
    df = pandas.DataFrame(columns=['doc'] + article)
    for doc in documents:
        df['doc'] = numpy.arange(0, len(documents))
        doc_num = tf_doc[index]
        for word in doc:
            for text in article:
                if(text == word):
                    idx = doc.index(word)
                    tf_idf_score = doc_num[word] * idf_dict[word]
                    tf_idf.append(tf_idf_score)
                    df.iloc[index, df.columns.get_loc(word)] = tf_idf_score
            index += 1
    df = df.loc[:, ~df.columns.duplicated()]
    df.fillna(0, axis=1, inplace=True)
    return tf_idf, df

```

Figure 5.14: Corpora and Document TFIDF Function

5.4 Cosine Similarity

Once all term frequencies and inverse document frequencies have been calculated, the Cosine Similarity of the subject document to the two corpora can be calculated using the equation;

$$\cos \theta = \frac{a \cdot b}{||a|| ||b||} = \frac{\sum_i^n a_i b_i}{\sqrt{\sum_i^n a_i^2} \sqrt{\sum_i^n b_i^2}} \quad (5.3)$$

The helper function, Fig.5.15, takes in the subject document and corpora as a list represented by *data*, the article term frequency as *tfidf*, the overall TFIDF as *df* and the subject document as *article*. These are passed on to the Cosine Similarity Function, Fig.5.16, as each document in the *data* list to the function.

```

def rank_sim(data, tfidf, df, article):
    sim = []
    for doc_num in range(0, len(data)):
        sim.append(cos_sim(tfidf, df, article, doc_num).tolist())
    return sim

```

Figure 5.15: Cosine Similarity Helper Function

```

def cos_sim(tfidf, df, article, doc_num):
    dtp = 0
    art_mod = 0
    doc_mod = 0
    print(df)

    for word in article:
        dtp += tfidf[word] * df[word][df['doc'] == doc_num]

        art_mod += tfidf[word] * tfidf[word]

        doc_mod += df[word][df['doc'] == doc_num] * \
            df[word][df['doc'] == doc_num]

    art_mod = numpy.sqrt(art_mod)
    doc_mod = numpy.sqrt(doc_mod)

    denom = art_mod * doc_mod

    similarity = dtp/denom

    return similarity

```

Figure 5.16: Cosine Similarity Function

Once they are passed to the Cosine Similarity, the similarity of the subject document(*article*) against itself (*data[0]*), the subject document against the Malay Corpus (*data[1]*) and finally subject document against the Iban Corpus (*data[2]*) is calculated and returned and saved to the *sim* list and returned to the main function. An example of the final similarity output is shown at Fig.5.17.

```
[1.0, 0.5670239270614824, 0.15476751817340184]
```

Figure 5.17: Similarity Output from Cosine Similarity Function

5.5 Results and Interpretation

5.5.1 Cosine Similarity with Source Articles as Subject Documents

The results in Table 5.8 and Table 5.9 show that by deduction, a greater decimal similarity indicates the origin language of the document. As for the Original column, the values alternate between 1.0 and 0.999 due to the manipulations done on the subject document such as punctuation removal, lowercase implementation and digit removal too.

For the Iban Similarity Column, the values are higher but closer to the Malay scores as the Iban corpora was smaller as compared to the more expansive corpus for Malay.

The use of the raw Similarity Score as the final indicator is due to the fact that only the corpus or a document containing all the words in the corpus at least once, can be said to be perfectly similar (Similarity Score of 1.0). Hence a document with less words will receive a lower Similarity Score as compared to the one with more words. The only condition to this is that the similarity should be at least greater than 0.150 of whatever language is indicated. This is to ensure that anything lower can go through the process using the Interpretation Approach 1.

Therefore, the highest similarity score, to a degree of 3 decimal places, between Malay and Iban will be the document's language as defined by the Cosine Similarity.

5.5.2 Interpretation Approach 1

This seeks to find the similarity of each document with relation to words that are exclusive only in a Malay text or in an Iban text. Hence, the removal of any similar words that appear in both corpora so as to match exclusively with the Malay or Iban language. The results will be interpreted the same way as before; a higher Similarity Score shows which language the document is in.

The overall effects of this is to lower the similarities scores in Malay and Iban as the number of terms consider in each corpora would reduce but the accuracy of the language detection would be higher(Refer to Table 5.3). And referencing the point of the smaller the document size, the smaller the similarity, in the case of the approach it would be *the fewer the unique words in a document the smaller the similarity and vice versa*.

Table 5.6: C.S with Iban Articles as Subject Documents Approach 2

Similarity Score			
Article	Original	Malay	Iban
Tentuka penyakal enda ngeruga projek pemansang ke pengelantang rayat	1.0	0.068	0.283
Tegapka penyerakup ungkup Malaysia	1.0	0.116	0.318
Projek pemansang ti dipejalaika GPS ngayanka perintah ngemeratka peranak menua pesisir	1.0	0.253	0.396
MA63 entara isu ti ditegika PSB ba manifesto PRN ke deka datai	1.0	0.292	0.356
Uras ari darat ngamahka sungai, pantai Miri: Lee	1.0	0.158	0.287
14 rumah jalai pengarap, Gempung jalai pengarap di Layar nerima bantu ari UNIFOR	1.0	0.106	0.484
60 iku RELA Julau nyereta aktiviti beripai ngaga palan pengentap pendiau sementara ungkup peranak Rh Chat	1.0	0.104	0.229
Pemisi perengkaguna beguna kena ngemansangka pelajar nembiak sekula	1.0	0.210	0.393
Nembiak sekula dikearapka nyaup baru komuniti ba kandang menua sida empu	0.999	0.159	0.488
KM seruran ngarapka SUPP, PSB bebaik	0.999	0.471	0.535

If the Similarity scores are equal, the document can be assumed to contain the similar words and hence can be read as Malay or Iban and result will be inconclusive. This situation can appear when a foreign language is introduced as the subject document, in this case, the fallback is Interpretation Approach 2 which includes all words in the Malay and Iban corpora and not just the unique ones. If this check also returns a similar result then the document is undefined.

Table 5.7: C.S with Malay Articles as Subject Documents Approach 2

Similarity Score			
Article	Original	Malay	Iban
Tentuka penyakal enda ngeruga projek pemansang ke pengelantang rayat	1.0	0.559	0.097
Tegapka penyerakup ungkup Malaysia	1.0	0.556	0.074
Projek pemansang ti dipejalaika GPS ngayanka perintah ngemeratka peranak menua pesisir	1.0	0.514	0.221
MA63 entara isu ti ditegika PSB ba manifesto PRN ke deka datai	0.999	0.493	0.187
Uras ari darat ngamahka sungai, pantai Miri: Lee	1.0	0.441	0.104
14 rumah jalai pengarap, Gempung jalai pengarap di Layar nerima bantu ari UNIFOR	1.0	0.636	0.101
60 iku RELA Julau nyereta aktiviti beripai ngaga palan pengentap pendiau sementara ungkup peranak Rh Chat	1.0	0.387	0.160
Pemisi perengkaguna beguna kena ngemansangka pelajar nembiak sekula	0.999	0.516	0.073
Nembiak sekula dikearapka nyaup baru komuniti ba kandang menua sida empu	0.999	0.560	0.159
KM seruran ngarapka SUPP, PSB bebaik	0.999	0.543	0.232

5.5.3 Interpretation Approach 2

This approach seeks to take away the overall similarity as presented by the Cosine Similarity function without any other manipulation done to the results.

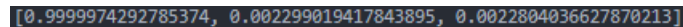
For example the article titled, *Tentuka penyakal enda ngeruga projek pemansang ke pengelantang rayat* (first row in the Iban Table 5.9) is written in Iban as it has a greater similarity with the Iban corpus of 0.307 as compared to the the Malay corpus which has 0.102. For the most part, the values do not reach 1.0 (perfect similarity) with any of the corpora as there are words present in the corpora that may not be present in the subject document itself.

This approach can give a clear estimate of what language the document is in but can leave room for improvement where values of each corpus maybe equal or too close to call.

5.5.4 Other Results

English

When give an English document with a listing of roughly 194433 words, the results were below the required 0.150 mark as expected (Refer to Fig.5.18)



[0.9999974292785374, 0.002299019417843895, 0.0022804036627870213]

Figure 5.18: English document Results

Indonesian

Since Indonesian is a Closely Related Language to Malay, it would make a good test ground. However the results would not be too conclusive as they were outside

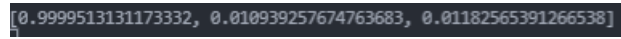
Table 5.8: C.S with Malay Articles as Subject Documents

Similarity Score			
Article	Original	Malay	Iban
Usah Terpengaruh Agenda untuk menjatuhkan Kerajaan	0.999	0.693	0.0284
Kukuhkan perpaduan demi Malaysia, kata Mawan	0.999	0.643	0.235
Projek pembangunan luar bandar bukti kesungguhan GPS	1.0	0.579	0.322
MA63, Akta Petroleum dan tanah NCR tiga isu utama dalam manifesto PSB	1.0	0.522	0.231
Sampah dari darat cemarkan sungai, pantai Miri	1.0	0.660	0.418
Peruntukan di bawah UNIFOR bukti keprihatinan kerajaan	1.0	0.753	0.365
60 anggota RELA Julau gotong-royong bina penempatan sementara Rh Chat	0.999	0.546	0.365
Prasana pendidikan mustahak dalam mengembangkan akademik dan bakat murid - Mawan	0.999	0.710	0.347
Pelajar berjaya suatu hari nanti digalak sumbang balik kepada masyarakat	0.999	0.613	0.271
Saya teruskan usaha perdamaian SUPP, PSB untuk kebaikan komuniti Cina - Abang Johari	0.999	0.675	0.413
Francisca Luhong dinobatkan Miss Universe Malaysia 2020	1.0	0.564	0.211
JAS giat adakan program kesedaran larangan pembakaran terbuka di Miri	1.0	0.710	0.176
GPS terus sekat UMNO ke Sarawak	1.0	0.562	0.299
Pertahan lapan kerusi majoriti Bidayuh di bawah GPS - Manyin	0.999	0.613	0.360
Amalkan sikap toleransi untuk mengekalkan perpaduan kaum	1.0	0.632	0.163

Table 5.9: C.S with Iban Articles as Subject Documents

Similarity Score			
Article	Original	Malay	Iban
Tentuka penyakal enda ngeruga projek pemansang ke pengelantang rayat	1.0	0.102	0.307
Tegapka penyerakup ungkup Malaysia	1.0	0.269	0.439
Projek pemansang ti dipejalaika GPS ngayanka perintah ngemeratka peranak menua pesisir	1.0	0.295	0.453
MA63 entara isu ti ditegika PSB ba manifesto PRN ke deka datai	1.0	0.314	0.406
Uras ari darat ngamahka sungai, pantai Miri: Lee	1.0	0.344	0.429
14 rumah jalai pengarap, Gempung jalai pengarap di Layar nerima bantu ari UNIFOR	1.0	0.168	0.535
60 iku RELA Julau nyereta aktiviti beripai ngaga palan pengentap pendiau sementara ungkup peranak Rh Chat	1.0	0.390	0.449
Pemisi perengkaguna beguna kena ngemansangka pelajar nembiak sekula	0.999	0.266	0.445
Nembiak sekula dikearapka nyaup baru komuniti ba kandang menua sida empu	0.999	0.372	0.673
KM seruran ngarapka SUPP, PSB bebaik	0.999	0.471	0.622
Francisca Luhong James ngemegahka Sarawak pengudah bulih gelar Miss Universe Malaysia 2020	0.999	0.183	0.313
JAS Miri chakah ngatur program nagang pengawa nunu chara terbuka	1.0	0.156	0.325
KM: UMNO ditagang ngerembai ke nengeri tu	1.0	0.560	0.592
Bansa Bidayuh dipinta beserakup ngetanka lapang DUN ke diuan bansa nya	0.999	0.415	0.433
Kemeranka ulah bebatak asur ngambika penyerakup ba pupu bansa majak tegap	1.0	0.276	0.652

the testing grounds. However a test conducted with a document containing 10887 publicly sourced Indonesian words had the a similarity of Malay - 0.0136, Iban - 0.104 initially (refer to Fig.5.19) using the initial approach and Malay -0.0136, Iban - 0.0104 for the secind approach too (Refer to Fig.5.20)



```
[0.9999513131173332, 0.010939257674763683, 0.01182565391266538]
```

Figure 5.19: Indonesian Results using Approach 1



```
[0.9999513131173332, 0.013649116881800904, 0.010391678811026688]
```

Figure 5.20: Approach 2 Results

Chapter 6

Conclusion and Future Work

6.1 Conclusion

By using the Bag of Words approach coupled with Term Frequency-Inverse Document Frequency and Cosine Similarity, it can be show that Malay and Iban can be Identified, to a degree of 3 decimal places, as separate languages. This in turn means Low Resource languages with a closely related language of a relatively larger resource orientation can be identified using said closely related language.

The biggest limitation however would be the fact that building the corpora for LRLs can be highly tasking as the sources are few and for some LRLs there is no standard context of how the language is written or used.

6.2 Future Work

Future Work would include the expansion of both corpora as due to time and resources, the corpora is relatively small as compared to the actual language sizes.

Another point to note is the introduction of morphological analysis of the lan-

guages themselves to provide a more in-depth review of the languages as they stand.

Finally, expanding the Corpus to cater for more unique words between Malay and Iban would increase the overall accuracy of the system.

Bibliography

- [1] Ethnologue, “How many languages are there in the world?,” 2020.
- [2] Sciforce, “Nlp for low-resource settings,” Oct. 2019.
- [3] T. Midzi, “Low resource language identification: Using malay and iban,” 2020, (accessed: 25.05.2021).
- [4] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbis, “Named entity recognition: Fallacies, challenges and opportunities,” *Computer Standards & Interfaces*, vol. 35, no. 5, pp. 482–489, 2013.
- [5] T. Verma, R. Renu, and D. Gaur, “Tokenization and filtering process in rapidminer,” *International Journal of Applied Information Systems*, vol. 7, no. 2, pp. 16–18, 2014.
- [6] J. B. Lovins, “Development of a stemming algorithm,” *Mech. Transl. Comput. Linguistics*, vol. 11, no. 1-2, pp. 22–31, 1968.
- [7] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural language processing: python and NLTK*. Packt Publishing Ltd, 2016.
- [8] O. Dereza, “Lemmatisation for under-resourced languages with sequence-to-sequence learning: A case of early irish,” in *Proceedings of Third Workshop* “Compu”, vol. 4, 2019, pp. 113–124.
- [9] K. Soumya George and S. Joseph, “Text classification by augmenting bag of words (bow) representation with co-occurrence feature,” *IOSR J. Comput. Eng*, vol. 16, no. 1, pp. 34–38, 2014.

- [10] R. Dale, H. Moisl, and H. Somers, *Handbook of natural language processing*. CRC Press, 2000.
- [11] N. Mehdi, “Developing a kashmiri morphological analyzer generator,”
- [12] H. Abdullah, “The morphology of malay,” Ph.D. dissertation, University of Edinburgh, 1972.
- [13] A. H. Omar, “The iban language of sarawak: A grammatical description.,” Ph.D. dissertation, SOAS University of London, 1969.
- [14] M. Zampieri, B. G. Gebre, and S. Diwersy, “N-gram language models and pos distribution for the identification of spanish varieties (ngrammes et traits morphosyntaxiques pour la identification de variétés de l’espagnol)[in french],” in *Proceedings of TALN 2013 (Volume 2: Short Papers)*, 2013, pp. 580–587.
- [15] A. Al-Saffar, S. Awang, H. Tao, N. Omar, W. Al-Saiagh, and M. Al-bared, “Malay sentiment analysis based on combined classification approaches and senti-lexicon algorithm,” *PloS one*, vol. 13, no. 4, e0194852, 2018.
- [16] R. Alfred, L. C. Leong, C. K. On, and P. Anthony, “Malay named entity recognition based on rule-based approach,” 2014.
- [17] S. S. Juan, L. Besacier, B. Lecouteux, and M. Dyab, “Using resources from a closely-related language to develop asr for a very under-resourced language: A case study for iban,” 2015.
- [18] B. Ranaivo-Malançon, “Automatic identification of close languages-case study: Malay and indonesian,” *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 2, no. 2, pp. 126–134, 2006.
- [19] V. Marivate, T. Sefara, V. Chabalala, K. Makhaya, T. Mokgonyane, R. Mokoena, and A. Modupe, “Investigating an approach for low resource language dataset creation, curation and classification: Setswana and sepedi,” *arXiv preprint arXiv:2003.04986*, 2020.

- [20] U. Borneo. (), [Online]. Available: <https://www.utusanborneo.com.my/>. (accessed: 14/09/2020).
- [21] R. Mitchell, *Web scraping with Python: Collecting more data from the modern web.* ” O’Reilly Media, Inc.”, 2018.
- [22] W. Scott, *Tf-idf from scratch in python on real world dataset*, 2020.
- [23] L. Huang, “Measuring similarity between texts in python,” *linea*]. *Disponibile en: <https://sites.temple.edu/tudsc/2017/03/30/measuring-similarity-between-texts-inpython/>*. [Accedido: 30-Marzo-2017],
- [24] L. Richardson, *Beautiful soup*. [Online]. Available: <https://www.crummy.com/software/BeautifulSoup/>.
- [25] *Python data analysis library*. [Online]. Available: <https://pandas.pydata.org/>.