2024

MASTER'S THESIS

Inducing Multiple Bilingual Dictionaries by Reusing Hub Language Encoders

ACADEMIC SUPERVISOR: MURAKAMI Yohei

Graduate School of Information Science and Engineering
Ritsumeikan University

MASTER'S PROGRAM
MAJOR in Advanced Information Science and Engineering

STUDENT ID: 6613220010-7

NAME: ABIA Putrama Herlianto

Multiple Bilingual Dictionary Induction by Reusing the Encoders of Hub Languages

Abia Putrama HERLIANTO

Abstract

Indonesia, with over 700 languages spoken by 280 million people, is highly diverse linguistically. Many of these languages are endangered, increasingly replaced by Indonesian. The lack of digital resources for these low-resource languages hinders their revitalisation and integration into Natural Language Processing (NLP) applications, making digital corpora crucial for their survival.

Developing bilingual dictionaries is the first step in creating these corpora. Traditional methods are labour-intensive and costly. Bilingual dictionary induction using neural network models is more efficient, learning word transformations even with minimal corpora. However, creating dictionaries for all language pairs remains costly and the performance can be improved.

This research proposes reusing the encoder of a hub language to induce multiple bilingual dictionaries among closely related languages. In a sequence-to-sequence model, the encoder encodes the input word in the source language which is then fed to the decoder which produces the output word in the target language. By reusing the encoder previously trained on the same source language with a different target language, knowledge already learnt from the previous language pairs would be transferred. By clustering similar languages and producing dictionaries only for pairs involving the hub language, resources and effort are significantly reduced. This method leverages shared linguistic features and transfer learning to enhance performance and streamline the process. To this end, we address the following issues.

Hub Language Identification

This research proposes using a hub language, which is the pivot language between a group of closely related languages. Out of these languages, the question that we address is which language should be the hub language and based on what criteria.

Optimal Encoder-Reusing Training Order

In the proposed multiple bilingual dictionary induction process, the models

for each language pair are not trained together at once but rather individually in order. The encoder is reused after the first language pair and trained again on the second language pair and so on. Whether the order of the language pairs has any effect or not and, if there is, which order is optimal or not is one question in this research.

To solve the first issue, hub language identification methods based on summed distance, distance to the centre, and dataset size were compared. Experiments involving all three identified hub languages were conducted to see which one performed the best.

To solve the second issue, several training orders based on random order, similarity, or dataset size were conducted. The results were compared with baseline models that were trained separately without reusing any encoders. The results were validated using *k*-fold cross validation.

The languages used were Indonesian, the Minangkabau language, Malay, Palembang Malay, and Banjarese Malay, which were all in one cluster. The contribution of this research to the issue is as follows.

Hub Language Identification

Three methods to determine the hub language of a group of closely related languages were described: summed distance-based, medoid-based, and dataset size-based. They are Malay, Minangkabau, and Indonesian, respectively.

Optimal Encoder-Reusing Training Order

The optimal encoder-reusing training order is a combination of considerations of both dataset size and similarity. The first language pair was trained for the language pair with the largest dataset size (Indonesian-Minangkabau) and the rest of the language pairs were trained in descending order of similarity. The results outperformed the baseline models trained individually and models trained with random order. Indonesian-Malay, Indonesian-Banjarese Malay, and Indonesian-Palembang Malay showed accuracies of 66.24%, 65.96%, and 64.43%. These outperformed the baselines and random orders across the board, ranging from improvements of 0.29% to 1.75%.

Inducing Multiple Bilingual Dictionaries by Reusing Hub Language Encoders

Contents

Chapter 1 Introduction

Indonesia is a country in Southeast Asia, notable for being the largest country in the region and the 15th largest country in the world [1]. With almost 280 million people spread across 17,504 islands, significant diversity exists among its population [2, 3, 4]. This is true for linguistic diversity as well with more than 700 different languages spoken by 1,300 ethnic groups [5, 6]. This makes Indonesia the second most linguistically diverse country in the world with about 10% of all the world's languages in Indonesia [6, 7]. According to Ethnologue, most of these languages belong to the Austronesian language family while about 200 belong to various Papuan language families [6].

However, this wealth of linguistic diversity is at risk. Although exactly how many distinct languages are spoken in Indonesia is a matter of academic debate, by any metric the picture is a stark one. According to Ethnologue, 506 out of 704 languages in Indonesia are considered endangered, which is defined as languages where "it is no longer the norm that children learn and use this language" [6]. 14 are already extinct. According to the Atlas of the World's Languages in Danger of Disappearing, at least 83 of the 640 languages in Indonesia are endangered while 14 are extinct [8]. Although there is no direct oppression of local languages in favour of the national language Indonesian, because education is primarily conducted in Indonesian children are increasingly conditioned to see Indonesian as superior to their mother tongue, precipitating the endangerment of these languages [8].

Language revitalisation efforts are in Indonesia are significantly hampered by the lack of resources. Most of the languages of Indonesia are classified as low-resource languages, meaning they have little to no digital corpora [9]. Not only digital, many of these languages also lack physical corpora, such as dictionaries, which are crucial for language revitalisation efforts. The scarcity of these resources makes it challenging to develop language classes and produce educational materials like textbooks, further impeding efforts to preserve and revitalise these languages.

Efforts led by the Social Intelligence Laboratory's Professor Yohei Murakami

at the Indonesia Language Sphere aim to develop comprehensive bilingual dictionaries for Indonesian ethnic languages [10]. Creating these dictionaries is a crucial first step in enriching low-resource languages, providing a foundation for language studies and additional educational materials. This initiative benefits both native speakers and learners. Notably, one advancement of the project includes employing a constraint-based technique for bilingual lexicon induction using a pivot language, particularly effective for closely related languages [11, 12].

However, creating bilingual dictionaries manually is labour-intensive, time-consuming, and resource-expensive. Several challenges need addressing. First, due to high costs, prioritising certain language pairs can maximise effectiveness while minimising expenses. Second, there is room for improving the performance of existing methods. For instance, Resiandi et al. proposed a neural network-based approach that builds on the constraint-based technique to induce bilingual dictionaries more efficiently [13].

This research focuses on 30 languages spoken in Indonesia and 1 in Malaysia, selected based on the number of speakers and their availability in the Automated Judgment Similarity Program (ASJP) database [14]. A language similarity matrix for these 31 languages was constructed to create a coordinate representation on a Cartesian plane, where distances indicate language similarity. K-means clustering was then applied to categorise these languages into groups based on their similarities.

For reasons of corpus availability, a group of languages consisting of Malayic languages was selected. This group includes Indonesian, Malay, Palembang Malay, Banjarese Malay, and Minangkabau, all of which share over 60% similarity according to the ASJP database [14].

Firstly, methods to determine the hub language were investigated. Two methods were considered. The first method involved summing the distances from each language to all other languages and choosing the language with the smallest summed distance. The second method used medoids to find the language with the most central position on the plane. However, due to corpus availability, Indonesian was ultimately chosen as the hub language despite not being selected by either method.

Secondly, multiple bilingual induction models were developed, all using Indonesian as the source language. Baseline models were trained independently for each language pair. For the other models, training followed a specific order, reusing the encoder from the first language pair until the final one. Various orders were investigated, including those based on language similarity, data size (both descending and ascending), and random sequences. These models utilised a Bi-LSTM as the encoder, an LSTM as the decoder, and employed one-hot character-level embedding for tokenisation.

This thesis is organised into 6 chapters. The first chapter covers the linguistic diversity of Indonesia, challenges faced by its endangered languages, and the importance of creating digital corpora for them. It also introduces the main objectives and contributions of the research. The second chapter reviews existing related literature on bilingual dictionary induction, language families, and encoder reuse. The third chapter covers the first part of the research including language selection, similarity-based clustering, and hub language selection. The fourth chapter covers the methodology of the multiple bilingual dictionary induction process. Chapter 5 discusses the training and results of the experiments, while Chapter 6 draws the conclusions.

Chapter 2 Related Work

This chapter provides an overview of existing research and related work on the methods and knowledge utilised in this study. It examines the current state of bilingual dictionary induction and language families, as well as the current state of encoder reuse.

2.1 Bilingual Dictionary Induction

Creating bilingual dictionaries manually is a labour-intensive, time-consuming, and expensive process. It requires not only native speakers of both languages but also bilingual speakers and linguistic experts to ensure accuracy and comprehensiveness. The process is labour-intensive due to the need for detailed manual work in creating each dictionary entry, including definitions and contextual usage. It is time-consuming because each entry must be meticulously crafted and reviewed, which can take significant amounts of time. Additionally, it is expensive because of the substantial resources required, including financial costs and human effort. The automatic induction of bilingual dictionaries could significantly streamline this process, speeding up dictionary creation while reducing associated costs.

Early work focused on high-resource languages. Fung showed the effectiveness of noisy parallel corpora and comparable corpora for English and Chinese [15]. Li and Gaussier proposed a method to improve comparable corpora which in turn can improve bilingual lexicon extraction [16]. But bilingual lexicon extraction becomes a more difficult task for low-resource languages which are by definition lacking, either partially or completely, in parallel and comparable corpora [11].

One avenue of research in bilingual dictionary induction is pivot-based bilingual dictionary induction. This method creates a new bilingual dictionary from two separate dictionaries using a pivot language. For instance, with existing Indonesian-Malay and Indonesian-Minangkabau dictionaries, Indonesian serves as the pivot to induce a Malay-Minangkabau dictionary. Tanaka and Umemura implemented this to create a Japanese-French dictionary using English as the pivot language [17]. Ambiguities and polysemy in the third language cause issues,

however. For example, the Japanese word “競争” (*kyousou*) could be translated into the English words “competition”, “contest”, and “race”. The word “race” complicates matters when looking for the French translations due to it being polysemous. “Race” has two meanings, the first one being “to compete” which is related in meaning to “競争”. However, the second one is in the context of “human race”, which is therefore not a correct translation. This second meaning leads to the French word “race” of the same meaning. The authors proposed a solution using the structure of dictionaries to measure the nearness of word meanings and inverse consultation to select appropriate translations.

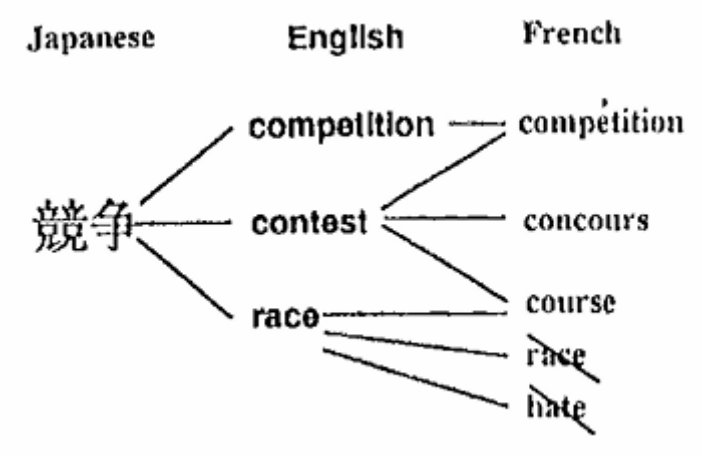

Figure 1 Equivalence candidates for “競争” [17].

Wushouer et al. and Nasution et al. proposed approaches treating pivot-based bilingual lexicon induction for low-resource languages as an optimisation problem [11, 12, 18]. Their approaches used constraints to determine whether two words through a pivot language are translation pairs. However, implementing this approach on a large scale to create multiple bilingual dictionaries remains a challenge. In particular, determining which language pairs should be prioritised is still an open question.

Resiandi et al. proposed utilising neural networks in bilingual dictionary induction [13]. Their purpose to create an Indonesian-Minangkabau dictionary using a model that would learn the patterns to transform words from Indonesian to Minangkabau using a comparatively small dataset, making use of the fact that

Indonesian and Minangkabau are closely related languages. They tackled two questions, namely which tokenisation method would be better and which combination of Long Short-Term Memory (LSTM) models and Bidirectional LSTMs (Bi-LSTM) work best. They experimented with character-level one-hot embedding using a Bi-LSTM as the encoder and an LSTM as the decoder, with character-level one-hot embedding using LSTMs for both the encoder and decoder, and with Byte Pair Encoding (BPE) with a Bi-LSTM as the encoder and an LSTM as the decoder. They found that character-level one-hot embedding using a Bi-LSTM as the encoder and an LSTM as the decoder performed best, with an average accuracy of 83.55%. The details of the results can be seen in Table 1.

Table 1 Resiandi et al.'s results [13].

Method	**Average Accuracy**
BPE	79.93%
Bi-LSTM -> LSTM with Character-Level One-Hot Embedding	83.55%
LSTM -> LSTM with Character-Level One-Hot Embedding	73%

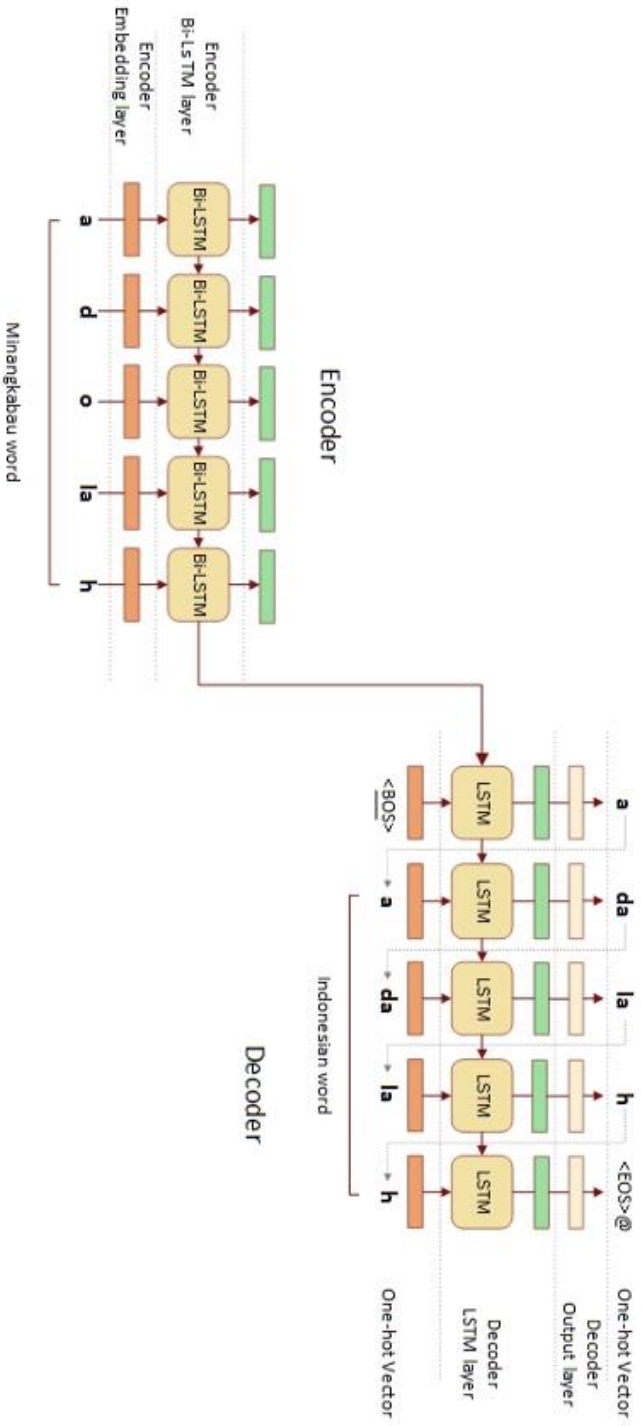

Figure 2 Resiandi et al.'s character-level sequence-to-sequence model architecture [13]. In this example, the Minangkabau word *adolah* is tokenised and the Indonesian translation *adalah* is output.

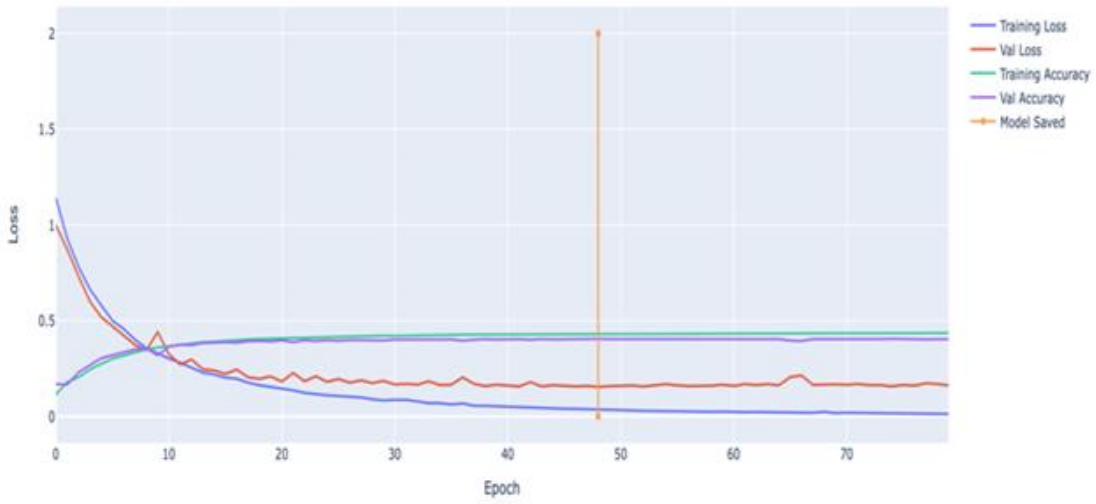

Figure 3 Epoch loss from validation and training on the character-level sequence-to-sequence model from Resiandi et al. [13].

2.2 Language Families and Similarities

Most languages are not isolated; they belong to groupings of languages called language families. A language family is a group of languages related through descent from a hypothetical, often unattested, ancestral language called the proto-language [19]. Over time, this proto-language then diverges into different daughter languages. These changes are due to a variety of causes, including influence from languages previously spoken in the area, changes that happen in some daughter languages are not shared with other daughter languages due to geographical isolation, or cultural elements. These daughter languages would then go on to experience the same thing, creating new daughter languaghes and leading to subfamilies.

One well-known example of a language family is the Romance languages. This family consists of numerous languages spoken initially in Europe but now throughout the world, including Spanish, French, Italian, Portuguese, Romanian, and many others [19]. All these languages were descended from the Vulgar Latin spoken in the Roman Empire. Initially, local dialects of Vulgar Latin developed through the regions of the Roman Empire, but geographical separation and the passage of time led to gradual changes in these dialects eventually leading to

speakers of one region not being able to understand speakers of another region, eventually creating the modern Romance languages and their dialects. A comparison of several words in several Romance languages can be seen in Table 2.

Table 2 Comparison of words from Latin and their descendants across various Romance languages [20, 21, 22, 23, 24].

English	**Latin**	**French**	**Spanish**	**Portuguese**	**Italian**
man	homo	homme	hombre	homem	uomo
son	filius	fils	hijo	filho	figlio
water	aqua	eau	agua	água	acqua
three	tres	trois	tres	três	tre
four	quattuor	quatre	cuatro	quatro	quattro

The Romance languages are themselves a subfamily of the greater Indo-European language family, one of the world's primary language family and the world's most-spoken language family [6, 19]. This family includes 144 languages, comprised of a diverse range of languages such as English and Hindi. All these languages are believed to have descended from a common ancestor language spoken thousands of years ago called Proto-Indo-European. A non-exhaustive family tree of the Indo-European languages can be seen in Figure 4 and comparisons between words from various Indo-European languages can be seen in Table 3.

Membership in a language family is determined through research in historical and comparative linguistics. Languages within the same family are identified by shared features that cannot be explained by chance or the effects of language contact [25]. Therefore, it is common for related languages to share features, especially closely related languages, which often exhibit numerous similarities and sometimes have limited mutual intelligibility [26].

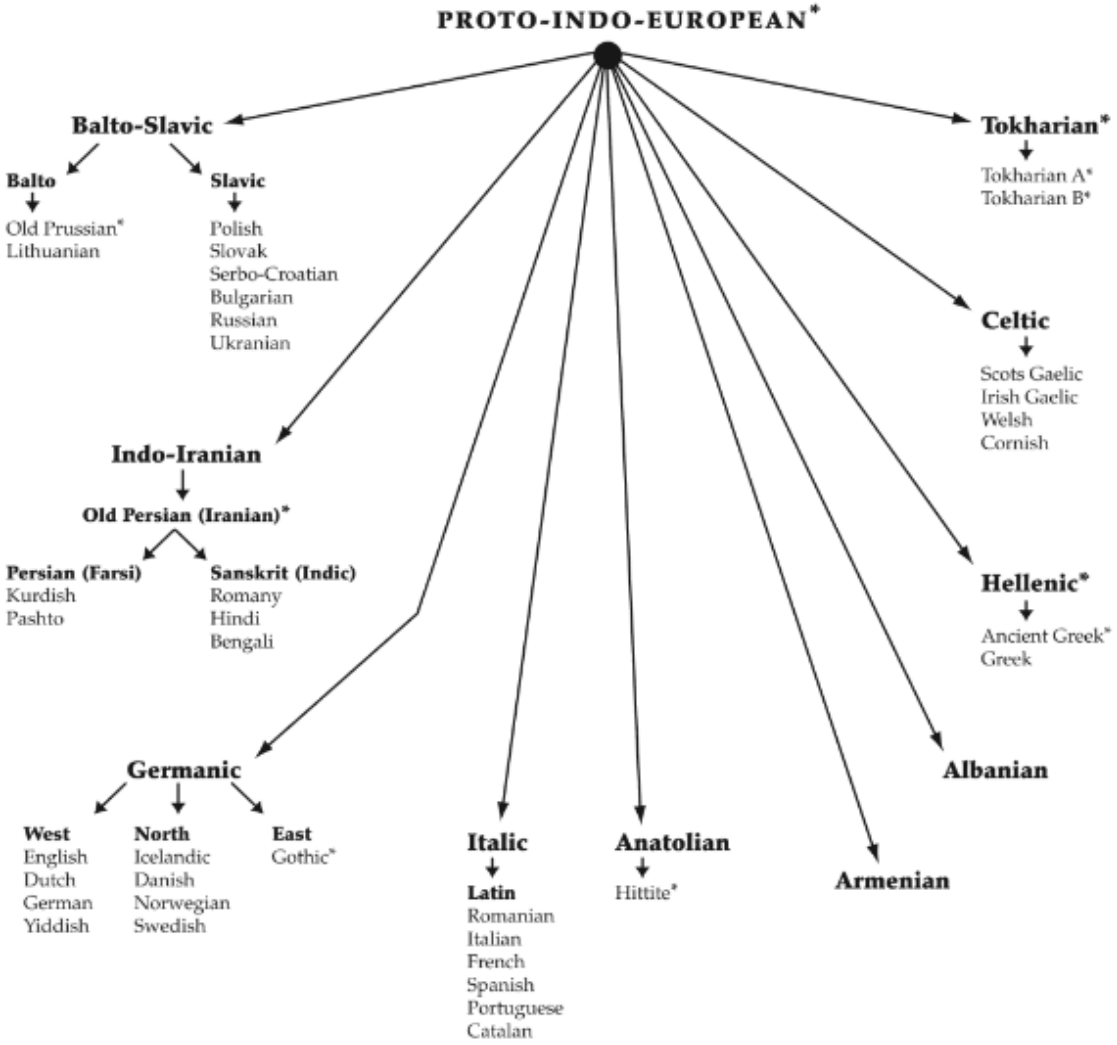

Figure 4 A non-exhaustive family tree of the Indo-European languages [19]. * denotes reconstructed forms.

Table 3 Comparison of words from five Indo-European languages [19].

English	**Sanskrit**	**Greek**	**Latin**	**Gothic**
father	pitar	pater	pater	fadar
foot	padam	poda	pedem	fotu
brother	bhratar	phrater	frater	brother
bear	bharami	phero	fero	baira
senile	sanah	hence	senex	sinista

Most Indonesian languages also belong to a single language family, namely the Austronesian language family, which is one of the world's primary language

families and is spoken in a wide area ranging from Madagascar to Southeast Asia, and to most of Oceania [4]. The Austronesian languages are posited to have originated on Formosa Island (Taiwan) and to have gradually spread from there, including to most of Indonesia [27]. As a result, most Indonesian languages share features and have similarities to each other

Table 4 Comparison of various words across Austronesian languages spoken in Indonesia [28, 29, 30, 31]. The Banjarese words are from the Hulu dialect, while the Javanese words are from the Surabaya dialect.

English	Indonesian	Minangkabau	Banjarese	Javanese	Sundanese
one	satu	ciek	asa	siji	hiji
two	dua	duo	dua	loro	dua
person	orang	urang	urang	uwong	jelema
house	rumah	rumah	rumah	omah	imah
we	kita, kami	kito, kami	kami	awaké dhéwé	arurang, urang sadayana, urang sararea

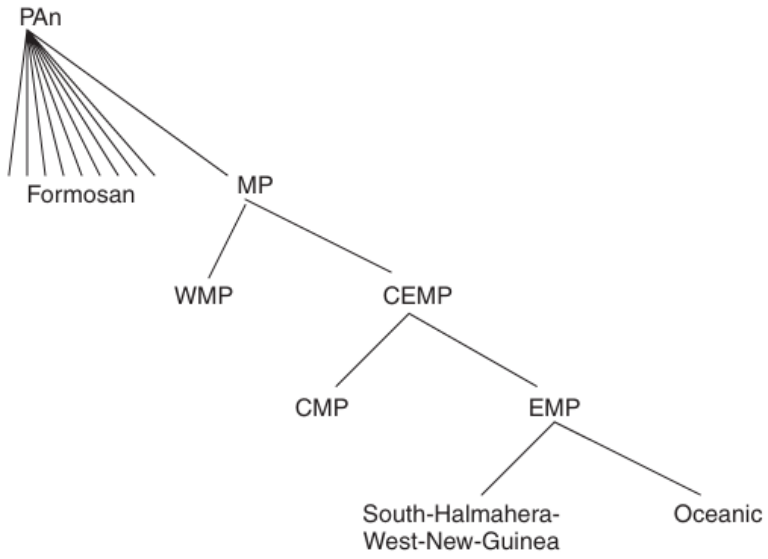

Figure 5 One interpretation of the Austronesian language family tree [32]. PAn refers to Proto-Austronesian, MP to Malayo-Polynesian, WMP to West-Malayo-Polynesian, CEMP to Central-East-Malayo-Polynesian, CMP to Central-Malayo-Polynesian, and EMP to East-Central-Malayo-Polynesian.

2.3 Encoder Reuse

Encoder reuse in neural network models, particularly in the context of natural language processing (NLP), involves sharing the same encoder across multiple tasks or language pairs. This approach leverages the shared linguistic knowledge captured by the encoder, which can improve model efficiency and performance. Several studies have demonstrated the benefits of encoder reuse.

Johnson et al. experimented with creating a multilingual neural machine translation system capable of zero-shot translation [33]. They did this without changing the architecture, using a regular neural machine translation model composed of an encoder, a decoder, attention, and a shared wordpiece vocabulary. The only change they made was to the input data. They added a token to the input sentence to indicate the required target language. For example, for English to Japanese data they would add the <2ja> token to signify that this sentence is for translating to Japanese, resulting in "<2ja> How are you? -> お元気ですか？".

The model was trained on various language pairs depending on the experiment they conducted. These languages were English, German, French, Portuguese, Spanish, Japanese, and Korean. For zero-shot translation specifically, the model was trained for English-Portuguese and English-Spanish. They trained two models. Model 1 was trained with Portuguese -> English and English -> Spanish data while Model 2 was trained with English <-> Portuguese and English <-> Spanish data. BLEU score was used for evaluation.

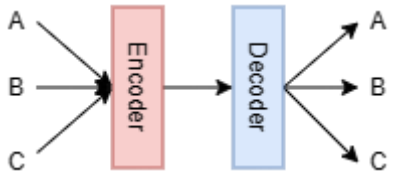

Figure 6 Illustration of the architecture of Johnson et al. in terms of encoders and decoders.

Their results showed that the model succeeded in implicit bridging, being able to translate between Portuguese and Spanish despite never seeing Portuguese <-> Spanish data. The performance achieved was reasonable, with details available

in Table 5. This shows the potential benefit of shared knowledge in a single model.

Table 5 Portuguese -> Spanish BLEU scores using various models from Johnson et al. [33].

Model	BLEU
PBMT bridged	28.99
NMT bridged	30.91
NMT Pt -> Es	31.50
Model 1 (Pt -> En, En -> Es)	21.62
Model 2 (En <-> [Es, Pt])	24.75
Model 2 + incremental training	31.77

Meanwhile, Lee at al. also experimented with sharing encoders. At the time of their paper's publication, word-based neural machine translation was the norm. Their paper proposed character-level neural machine translation and showed that it was possible to share a single character-level encoder across multiple languages by training a model on a many-to-one translation task [34]. Their results showed significant improvements, potentially sharing knowledge between several languages.

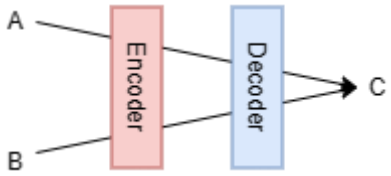

Figure 7 Illustration of the architecture of Lee et al. in terms of encoders and decoders.

Dong et al. created a machine translation model that could simultaneously translate sentences from one source language to multiple target languages [35]. A single encoder was used for all language pairs but utilising a different decoder for each language. The result was that their model outperformed models with different encoders and decoders for each translation direction. They showed that sharing the encoder can lead to positive results. However, they did not specify the order of training, nor did they take into account the genetic relationship between

the languages. The results for their experiments using large-scale corpora is shown in Table 6.

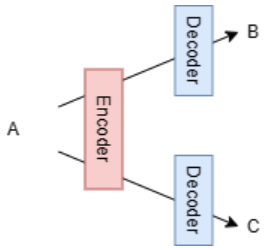

Figure 8 Illustration of the architecture of Zhang et al. in terms of encoders and decoders.

Table 6 Dong et al.'s results vs. single model given large-scale corpora in all language pairs [35].

Lang-Pair	**En-Es**	**En-Fr**	**En-Nl**	**En-Pt**
Single NMT	26.65	21.22	28.75	20.27
Multi Task	28.03	22.47	29.88	20.75
Delta	+1.38	+1.25	+1.13	+0.48

Chapter 3 Similarity-Based Language Clustering

3.1 Overview

The first part of this research involves several steps in order to process the languages. This consisted of selecting the languages, generating a language similarity matrix using ASJP, and clustering these languages based on similarity.

3.2 Language Selection

To leverage the advantage of language similarity, 30 Austronesian languages from Indonesia and 1 Austronesian language from Malaysia were chosen. These languages were chosen based on three criteria:

1. Number of speakers according to Ethnologue [6]
2. Language availability on the ASJP database
3. Dataset availability

The list of languages can be seen in Table 7.

30 of the most-spoken Austronesian languages in Indonesia were selected, with some exceptions. Southern Min (nan), the 17th most-spoken language with around 1.3 million speakers, was not selected due to being a Sinitic language. Hakka Chinese (hak), spoken by around 600,000 people, was also not selected for the same reason. Three Austronesian languages were not selected due to not being available on the ASJP database. These were namely Ngaju (nij), North Moluccan Malay (max), and Toraja-Sa'dan (sda). For Javanese, the Yogyakarta dialect was chosen because of its status as the prestige dialect [36]. Northern Nias was similarly chosen for the Nias language [37].

Malay was added later due to training data being available and the fact that it is an Austronesian language. Linguistically speaking, Indonesian and Malaysian Malay are two standardised registers of the same language [38] . They can be treated as two dialects of the same language. Significant differences still exist, however, especially in terms of vocabulary.

Table 7 List of selected languages.

No.	Country	Language	ISO 639-3 Code	Speakers (Millions)
1	Indonesia	Indonesian	ind	210
2	Indonesia	Javanese	jav	84.3
3	Indonesia	Sundanese	sun	42
4	Malaysia	Malay	zlm	32
5	Indonesia	Madurese	mad	13.6
6	Indonesia	Minangkabau	min	5.5
7	Indonesia	Buginese	bug	5
8	Indonesia	Palembang Malay	mui	3.9
9	Indonesia	Banjarese	bjn	3.5
10	Indonesia	Acehnese	ace	3.5
11	Indonesia	Balinese	ban	3.3
12	Indonesia	Betawi	bew	2.7
13	Indonesia	Sasak	sas	2.1
14	Indonesia	Batak Toba	bbc	2
15	Indonesia	Makassarese	mak	2.1
16	Indonesia	Ambonese Malay	abs	1.9
17	Indonesia	Batak Dairi	btd	1.2
18	Indonesia	Batak Simalungun	bts	1.2
19	Indonesia	Batak Mandailing	btm	1.1
20	Indonesia	Jambi Malay	jax	1
21	Indonesia	Gorontalo	gor	1
22	Indonesia	Nias	nia	0.8
23	Indonesia	Manado Malay	xmm	0.8
24	Indonesia	Batak Angkola	akb	0.7
25	Indonesia	Batak Karo	btx	0.6
26	Indonesia	Uab Meto	aoz	0.6
27	Indonesia	Bima	bhp	0.5
28	Indonesia	Manggarai	mqy	0.5
29	Indonesia	Komering	kge	0.5
30	Indonesia	Tetum	tet	0.4
31	Indonesia	Rejang	rej	0.4

3.3 ASJP-Based Similarity Calculation

The Automated Similarity Judgment Program (ASJP) is a project initiated by the Max Planck Institute for the Science of Human History in Germany [14]. Its primary goal is to apply computational approaches to historical comparative linguistics by analyzing basic vocabulary lists from 5,590 languages. Each language is represented by a 40-item basic vocabulary wordlist, akin to the Swadesh list.

Table 8 The 40 words in each language in the ASJP database [39].

blood	fish (noun)	mountain	star
bone	full	name (noun)	stone
breast (woman's)	hand	new	sun
come	hear	night (dark time)	tongue
die	horn (animal part)	nose	tooth
dog	I	one	tree
drink (verb)	knee	path	two
ear	leaf	person	water
eye	liver	see	we
fire	louse	skin	you

3.3.1 ASJPcode

The ASJP employs a simplified system to represent sounds known as ASJPcode [40]. Unlike the International Phonetic Alphabet (IPA), which uses a unique character for each distinctive sound [41], in ASJPcode a single character can represent one or more IPA sounds (or phones), supplemented by additional symbols to denote specific pronunciation features. Table 9 provides a comprehensive list of ASJPcode characters alongside their corresponding IPA phonemes.

Table 9 List of ASJPcode characters and their equivalent phones or features in IPA [40].

Character	IPA Phoneme / Feature
i	i, ɪ, y, ʏ
e	e, ø
E	a, æ, ɛ, œ, œ, e
3	ɨ, ɘ, ə, ɜ, ʉ, ɵ, ɞ
a	ɐ, ä
u	ɯ, u, ʊ
o	ɤ, ʌ, ɑ, o, ɔ, ɒ
p	p, ɸ
b	b, β
m	m
f	f
v	v
8	θ, ð
4	ɳ
t	t
d	d
s	s
z	z
c	t͡s, d͡z
n	n
S	ʃ
Z	ʒ
C	t͡ʃ
j	d͡ʒ
T	c, ɟ
5	ɲ
k	k
g	g
x	x, ɣ
N	ŋ
q	q
G	ɢ
X	χ, ʁ, ħ, ʕ
7	ʔ
h	h, ɦ
l	l
L	ʟ, ɭ, ʎ
w	w
y	j
r	r, ʀ, etc. (all varieties of “r-sounds”)
!	!, ǀ, ǁ, ǂ
~	Follows two consonants so that they are considered to be in the same position
$	Follows three consonants so that they are considered to be in the same position
“	Marks the preceding consonant as glottalised

3.3.2 Calculating Language Similarity

The ASJP calculates language similarity using Levenshtein Distance (LD), which measures the minimum number of edits (insertions, deletions, or substitutions) required to transform one word into another. To normalize for word length differences, the Normalized Levenshtein Distance (LDN) is used, which divides the LD by the length of the longer word. Further refinement is achieved through LDND (LDN Divided), which adjusts for chance similarities by dividing the average LDN for word pairs with the same meaning by the average LDN for word pairs with different meanings [42].

The LDND distances between all languages pairs were calculated and a language similarity matrix was created.

Figure 9 Language similarities (LDND) calculated using the ASJP for the 31 selected languages.

3.4 Clustering

Based on Nasution et al.'s work, the languages were clustered using *k*-means clustering [43]. Initially, only 30 languages were taken into account as Malay (zlm) was added at a later stage. Several comparisons were made.

K-means clustering is an unsupervised clustering algorithm used to partition a dataset into *k* distinct, non-overlapping groups called "clusters". *K* must first be chosen by the user. Then *k* initial centroids are selected from the dataset. Each data point is then assigned to the nearest centroid based on their distance to it, resulting in *k* clusters where each data point belongs to the cluster with the nearest centroid. New centroids are then calculated by taking the mean of all the data points assigned to each cluster. The new centroid is the average position of all points in the cluster. This is repeated until convergence, which is when centroids no longer change significantly or the assignment of data points no longer change between iterations [44].

Two types of input and two values of *k* were investigated. Nasution et al. in their research determined that a value of 5 for *k* was optimal for 32 Indonesian ethnic languages [43]. This research compares that with the value of 6 for *k*. As input to the *k*-means clustering algorithm, this research compares two types of input. The first is that each language has as its vector its similarity to all 29 other languages. This will henceforth be referred to as the vector method. The second is that the language similarity matrix is transformed into coordinates using Classic Multidimensional Scaling. This will henceforth be referred to as the coordinate method. The clusters are evaluated using purity against genetic relationships as established by linguists.

Multidimensional Scaling is a set of data analysis methods which "allow one to infer the dimensions of the perceptual space of subjects" [45]. The input to this method is a measure of similarity or dissimilarity of the objects under investigation. The output is a spatial configuration in which the objects are represented as points on a Cartesian plane. Similar objects are represented by points close to each other; dissimilar objects are represented by points that are far apart. If we have a matrix of distances between data points, multidimensional scaling outputs the coordinates of these points. Therefore, if we have a matrix of

distances between languages, multidimensional scaling outputs the coordinates of these languages.

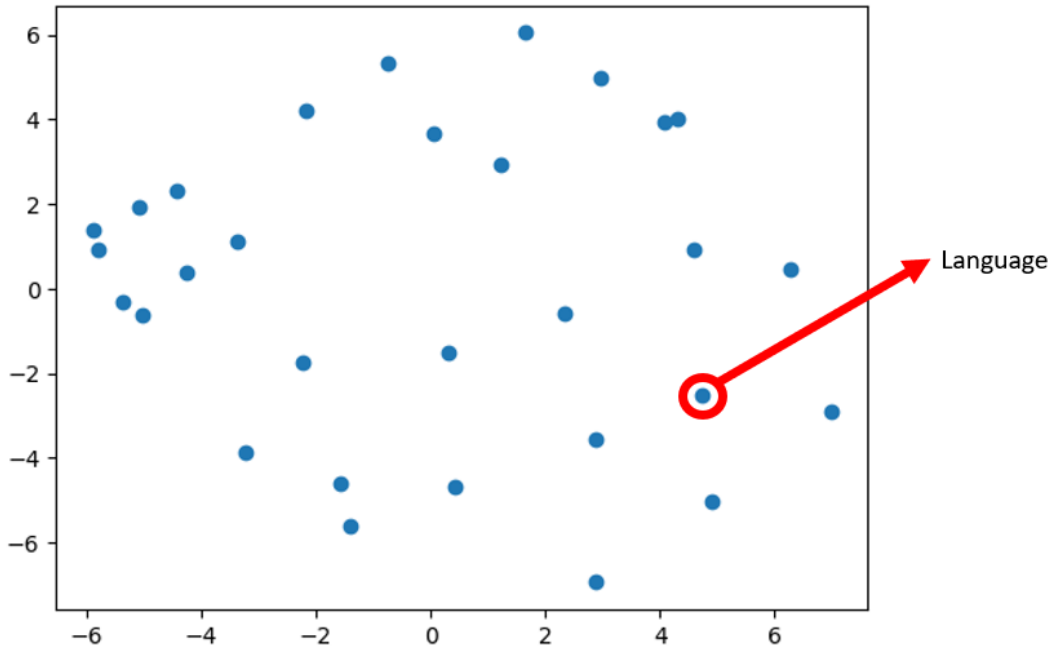

Figure 10 Illustration of the results of multidimensional scaling on a language similarity matrix.

Purity is a quantitative evaluation of a cluster compared to the gold set or ground truth [46]. Each result cluster is assumed to be the cluster that holds the most points from the clusters of the gold set (ground truth). The purity is computed using the sum of how many maximum points of each result cluster match with a considered gold set cluster divided by the total number of data points.

$$\frac{\sum_{i=1}^{k} \max_{j=1}^{t} \left(c_i \cap g_j \right)}{N}$$

Figure 11 Purity formula for *k*result clusters c1, c2, …, ck and *t* gold set clusters g1, g2, …, gt.

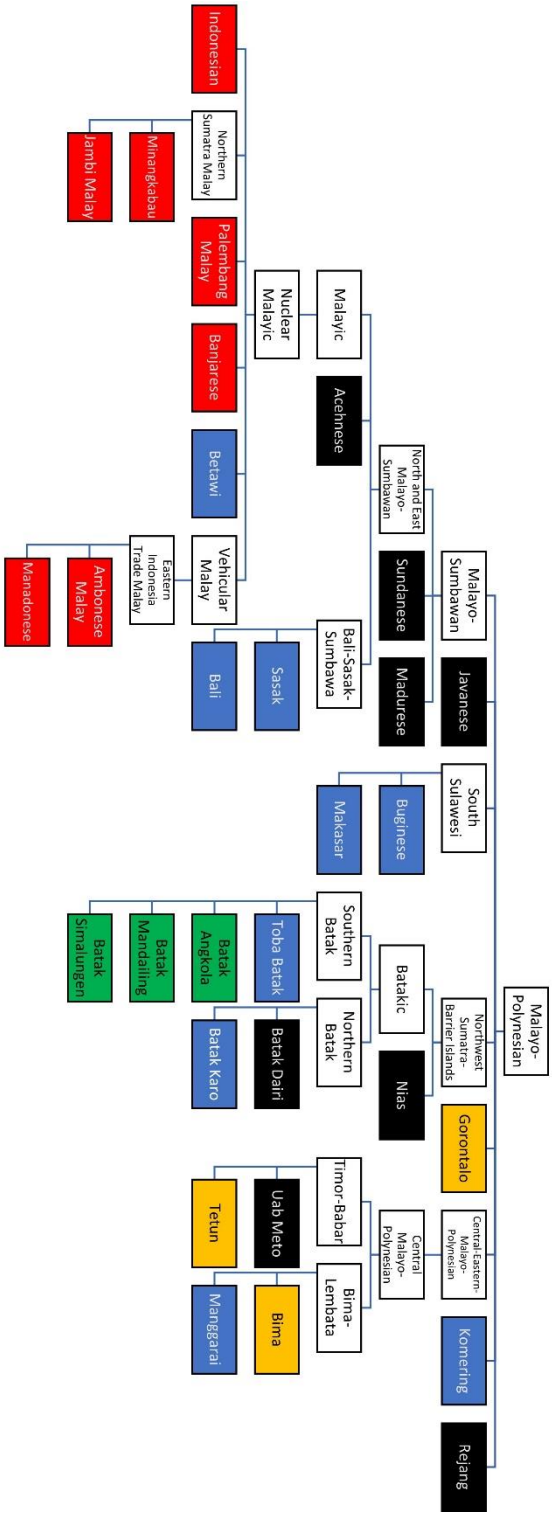

Figure 12 Results of the vector method with k = 5 and their positions in the genetic linguistic tree [47].

Firstly, the results of the using vector method as input and with the value of 5 for *k* was evaluated. The results are compared with their positions in the genetic linguistic tree as determined by linguists. The data is from Glottolog [47]. This can be seen in Figure 12.

The blue cluster is geographically spread out into clusters from Northwest Sumatra, South Sulawesi, the Lesser Sunda Islands, and 2 outliers. It has all the South Sulawesi and Bali-Sasak-Sumbawa languages and 2 of the 6 Batak languages. Betawi is an outlier from the Nuclear Malayic languages, all the others of which are in the red cluster. Manggarai is an outlier from the Central Malayo-Polynesian languages.

The yellow cluster has 2 of the 4 Central Malayo-Polynesian languages.

The green cluster has 3 of the 6 Batak languages.

The red cluster has most of the Nuclear Malayic languages except for Betawi, which is in the blue cluster.

The black cluster has 1 of the 4 Central Malayo-Polynesian languages and 1 of the 6 Batak languages.

Next, the results of using vector method as input and with the value of 6 for *k* was evaluated as illustrated in Figure 13.

The blue cluster is geographically spread out. Acehnese, Sundanese, and Madurese are from the Malayo-Sumbawan languages excluding the Malayic languages. Uab Meto and Bima, which are part of this cluster, are related. Tetun, which is closer to Uab Meto than Bima, and Manggarai, which is closer to Bima than Uab Meto, are not included.

The yellow cluster is clustered around Northwestern Sumatra and the Lesser Sunda Islands. It has 2 Batak languages and the related Nias language. It also has the closely related Bali and Sasak languages. Bima, which is closely related to Manggarai, is not included. The other Batak languages are also not included.

The green cluster is composed of closely related varieties of Malay. Batak Karo is the only outlier since no other Batak languages are included. Jambi Malay is not included, despite being closely related to Minangkabau.

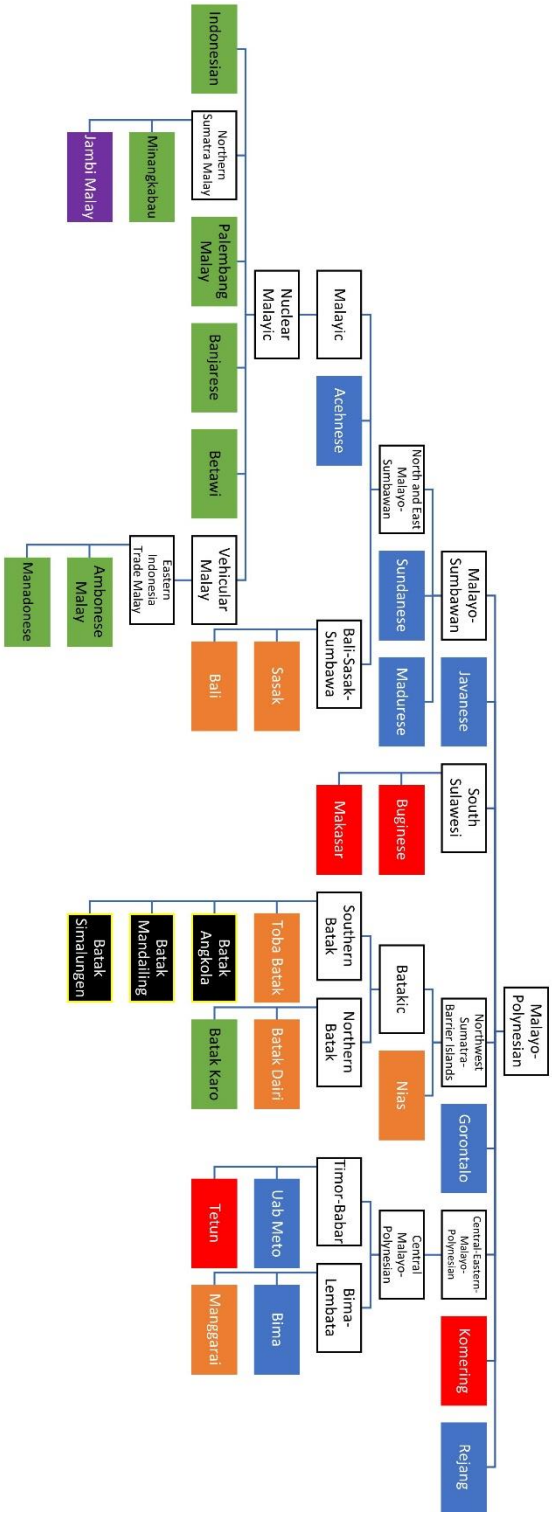

Figure 13 Results of the vector method with $k = 6$ and their positions in the genetic linguistic tree [47].

The red cluster has the closely related Buginese and Makasar languages and has Tetun without Uab Meto, its closest relative among the chosen languages.

The purple cluster only has one language, which is Jambi Malay. Based on linguistic genetic relationships, Jambi Malay could be part of the green cluster instead.

The black cluster has 3 of the 4 chosen Southern Batak languages, while Toba Batak is in the yellow cluster.

Overall, the vector method with $k = 6$ has some coherent clusters but several outliers such as Toba Batak and Jambi Malay. Notably, the green cluster of $k = 5$ is exactly the same as the black cluster of $k = 6$. Furthermore, the red cluster of $k = 5$ is almost exactly the same as the green cluster of $k = 6$. The black cluster of $k = 5$ is similar to the blue cluster of $k = 6$.

Afterwards, the coordinate method with both $k = 5$ and $k = 6$ were investigated and the results compared with the results of the vector method. The results can be seen in Figures 14, 15, 16, and 17.

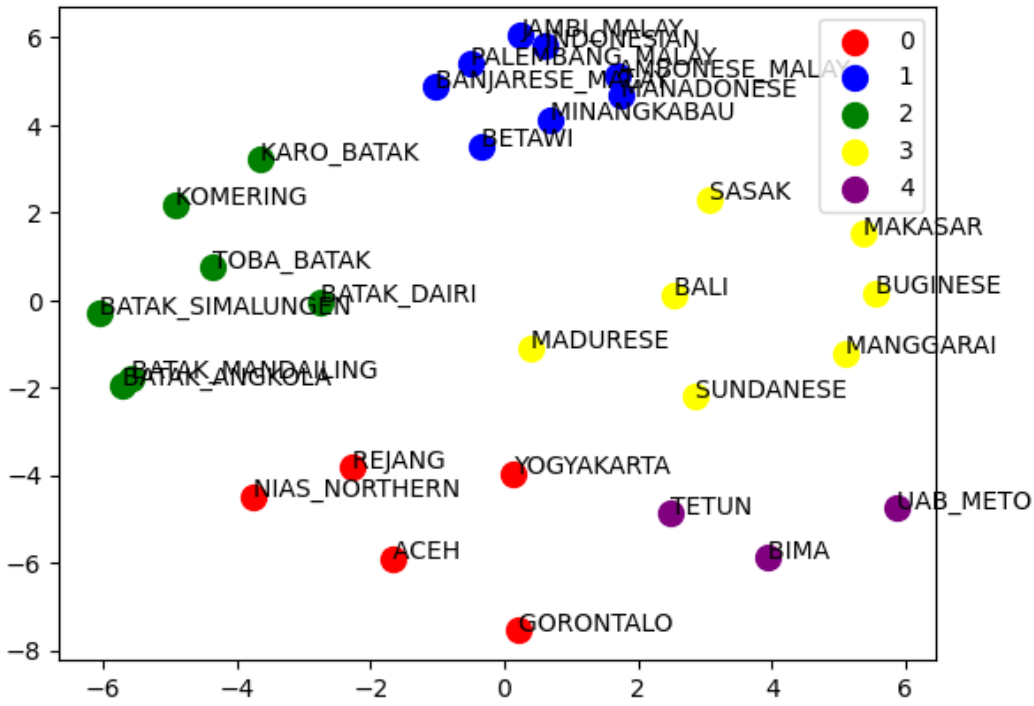

Figure 14 The results of the coordinate method using $k = 5$ illustrated on a Cartesian plane.

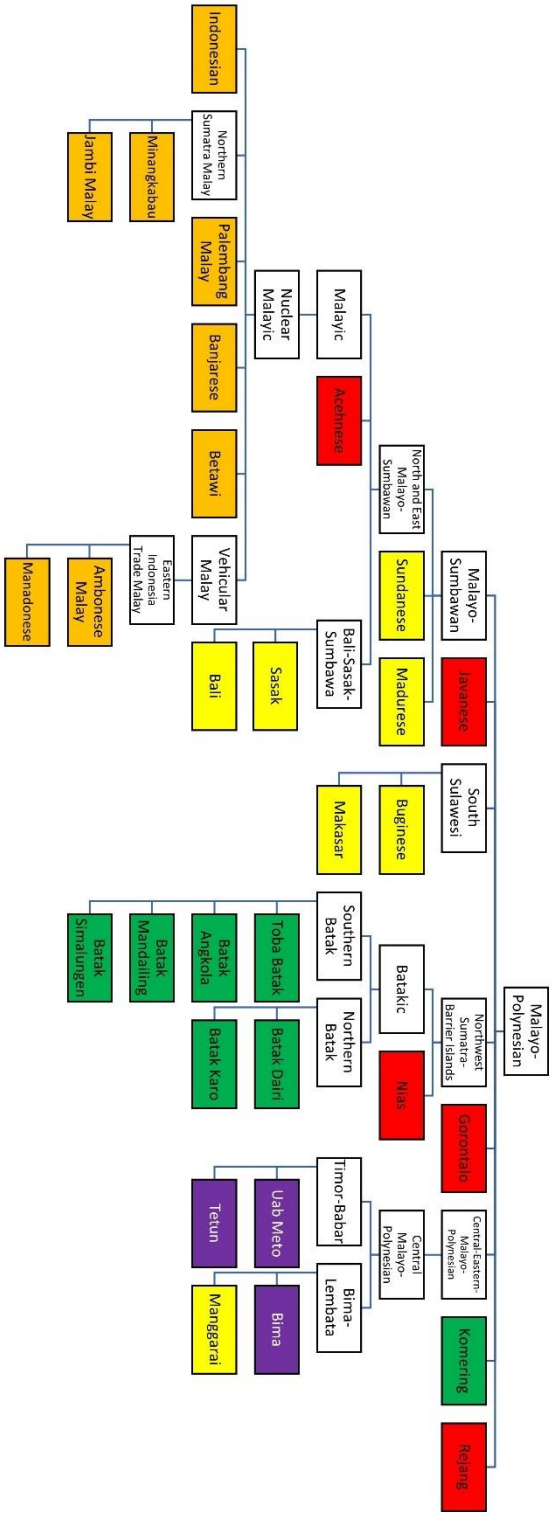

Figure 15 Results of the coordinate method with k = 5 and their positions in the genetic linguistic tree [47].

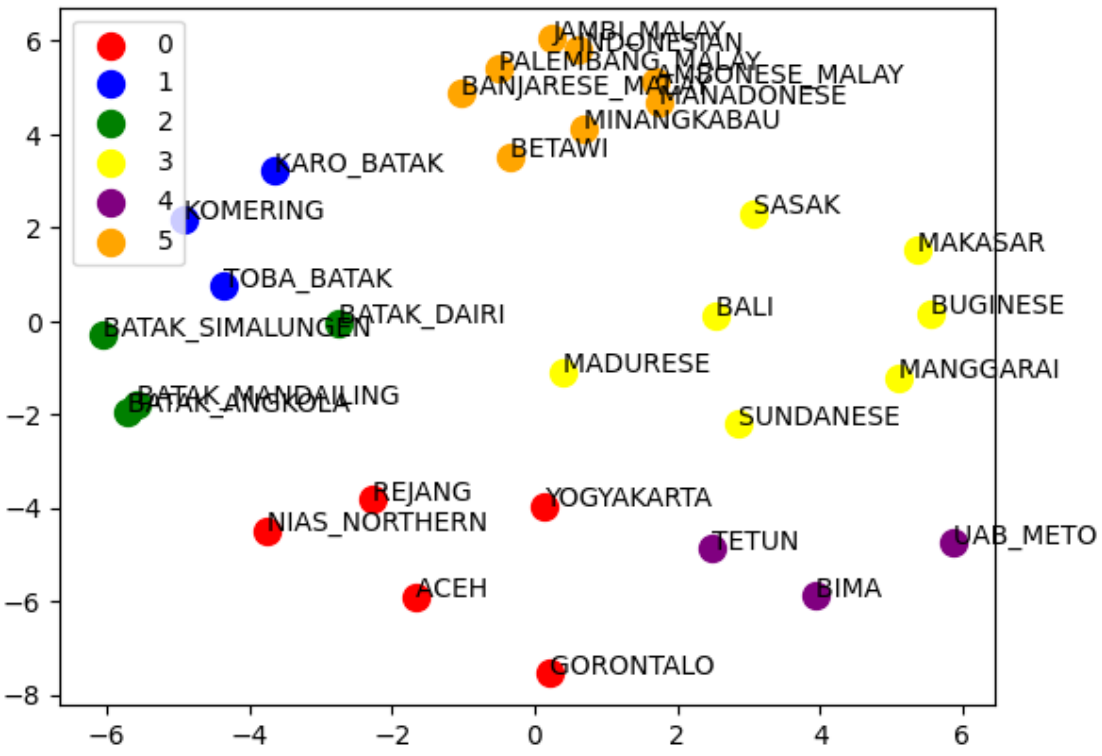

Figure 16 The results of the coordinate method using $k = 6$ illustrated on a Cartesian plane.

Comparing the results of all combinations, the Malayic languages are consistently grouped as one cluster with the coordinate method. For the vector method, for both values of k, there was one Malayic language which was in another group as can be seen in Figures 12 and 13.

The other Malayo-Sumbawan languages excluding Acehnese are consistently in one cluster with the coordinate method. For the vector method, half are always in a different cluster as can be seen in Figures 12 and 13.

The Batakic languages are grouped into one or two clusters in the coordinate method. In the coordinate method with $k = 5$, all the Batakic languages are notably grouped into one. For the vector method, the Batakic languages are consistently split in three different clusters as can be seen in Figures 12 and 13.

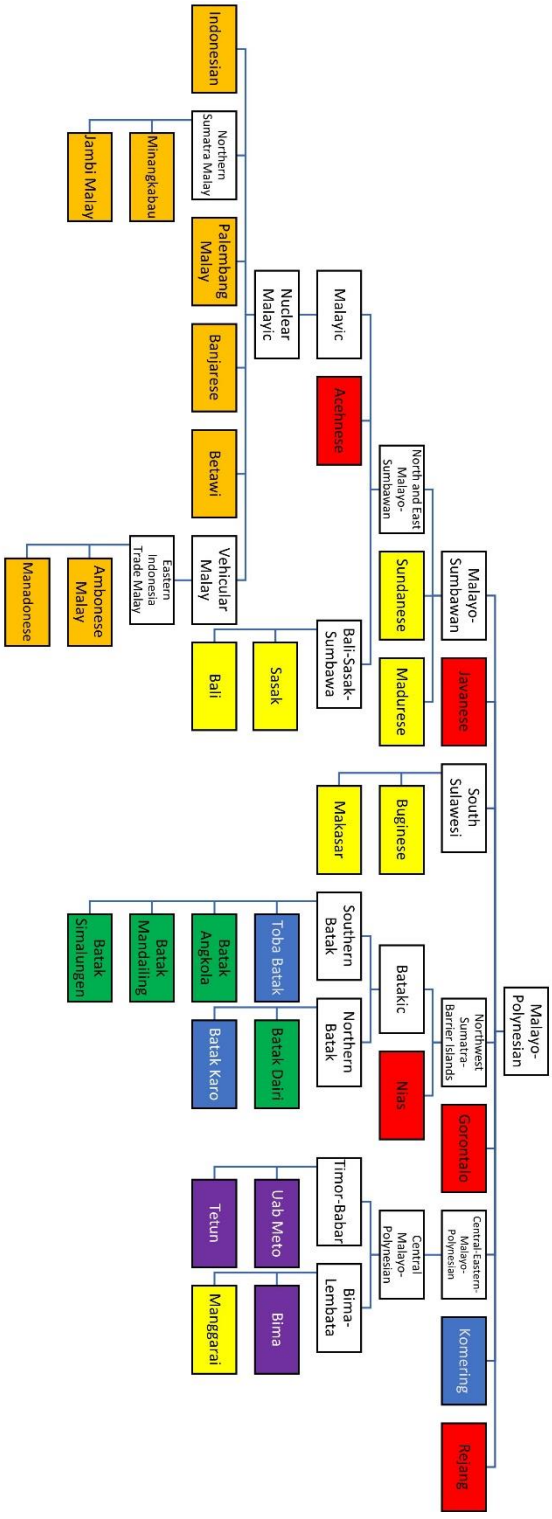

Figure 17 Results of the coordinate method with k = 6 and their positions in the genetic linguistic tree [47].

For the vector method, the Central-Eastern-Malayo-Polynesian languages are consistently split into three clusters. In contrast, in the coordinate method three languages – except Manggarai – are consistently grouped as one cluster as can be seen in Figures 15 and 17.

Overall, the vector method's groupings are less in line with the linguistics-derived genetic relationships between the languages. The vector method with k = 6 also has a group with only one language in it. The coordinate method is more in line with the genetic relationships between the languages and has clusters with relatively balanced numbers, especially for k = 5.

The purity was then calculated for a quantitative assessment of the clustering results. The ground truth for k = 5 and k = 6 can be seen in figures 22 and 23 and are based on the genetic relationship between the languages. The result was that for the vector method, purity was 0.77 for k = 5 and 0.73 for k = 6. For the coordinate method, purity was 0.8 for both k = 5 and k = 6.

Table 10 Purity results.

	Vector Method	**Coordinate Method**
k = 5	0.77	0.8
k = 6	0.73	0.8

The vector method purities are consistently lower than the coordinate method purities. Since the results of the coordinate method with k = 5 has a more cohesive cluster for the Batakic languages with all the Batakic languages in one cluster compared to k = 6 where the Batakic languages are separated into two clusters, k = 5 will be used.

Malay was later added. Using the coordinate method with k = 5, it was grouped with the other Malayic languages as in Figure 24.

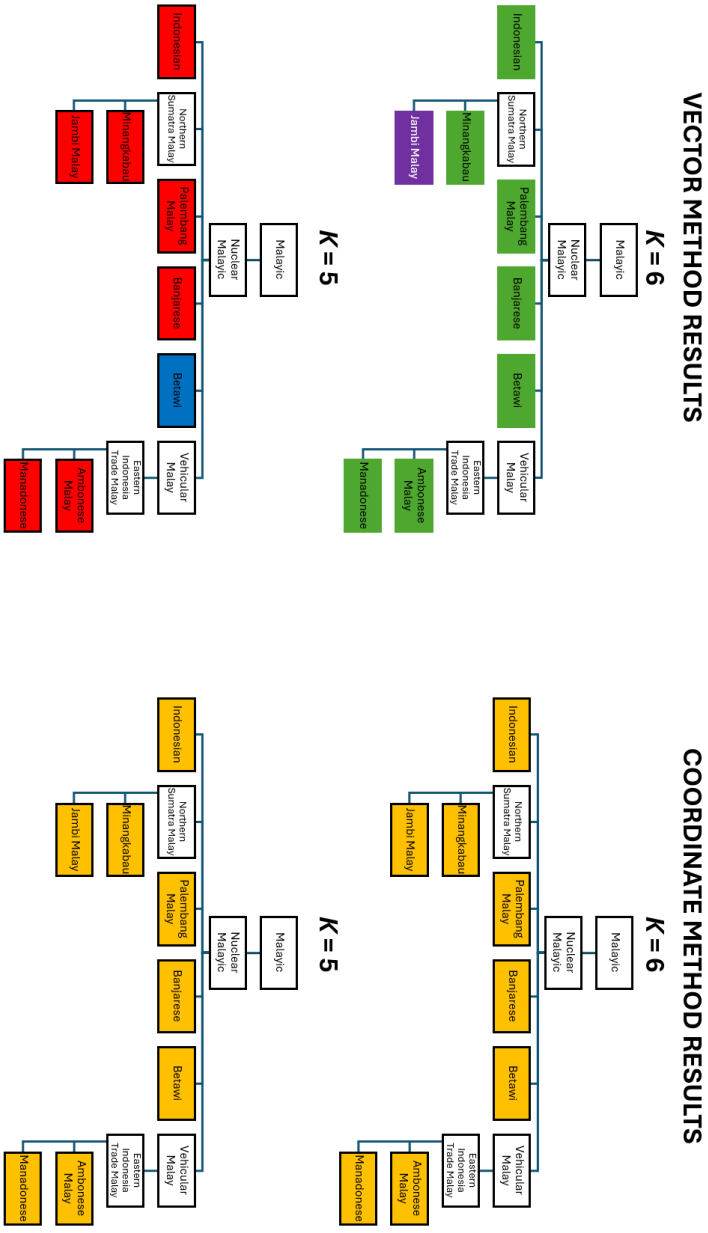

Figure 18 Comparison of the Malayic languages across the clustering results.

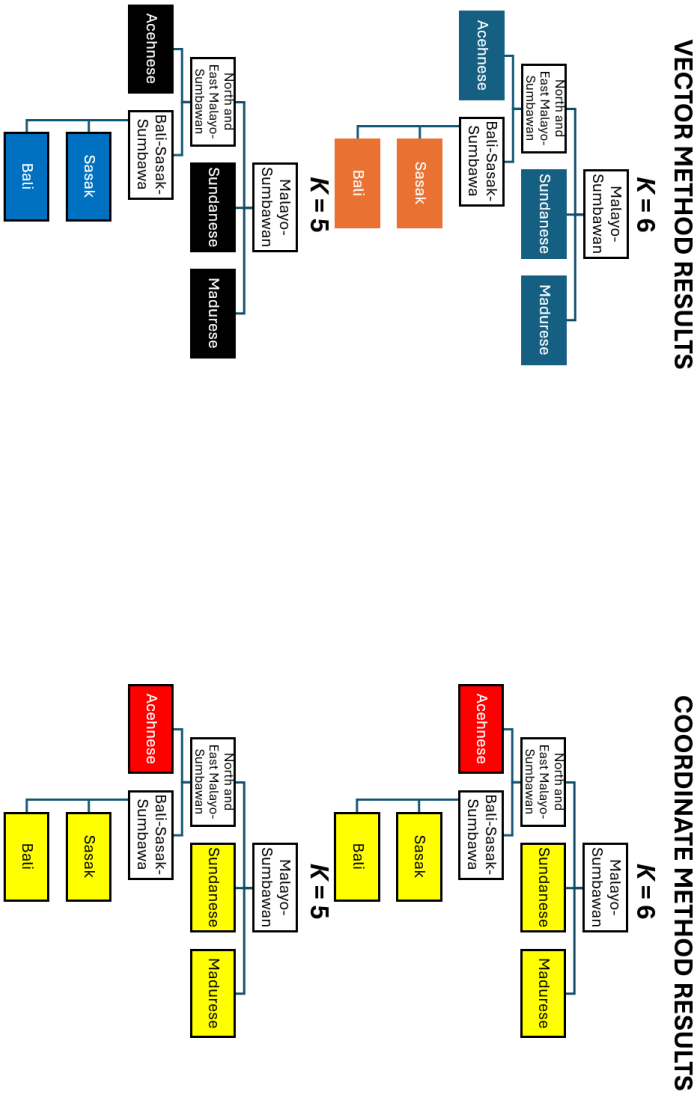

Figure 19 Comparison of the other Malayo-Sumbawan languages across the clustering results.

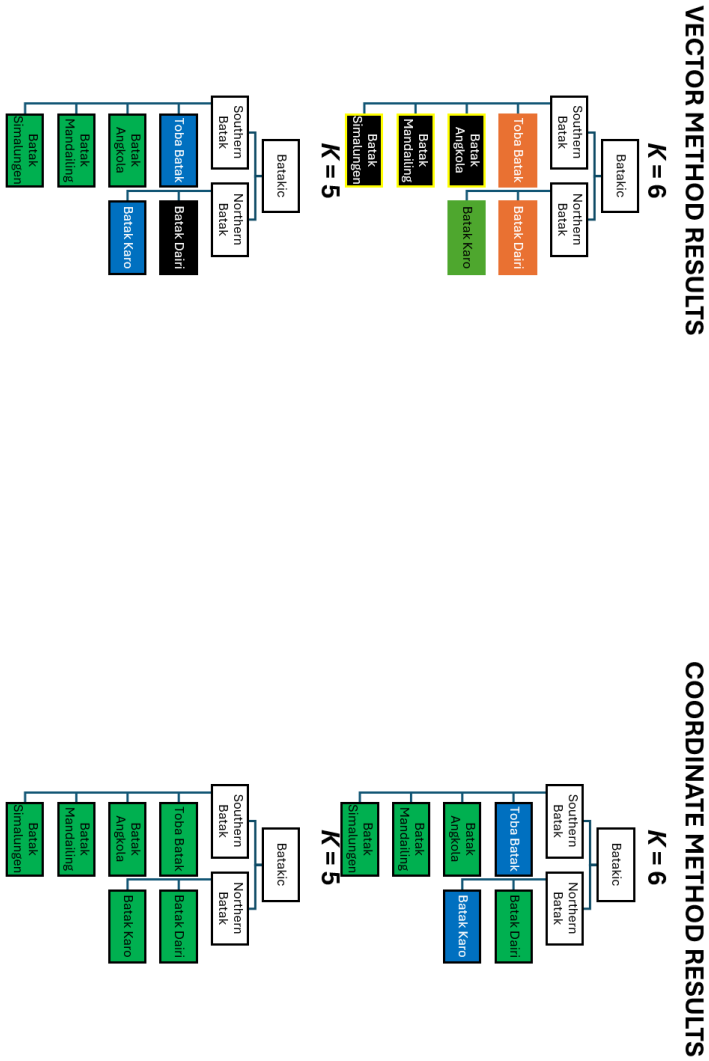

Figure 20 Comparison of the Batakic languages across the clustering results.

Figure 21 Comparison of the Central-Eastern-Malayo-Polynesian languages across the clustering results.

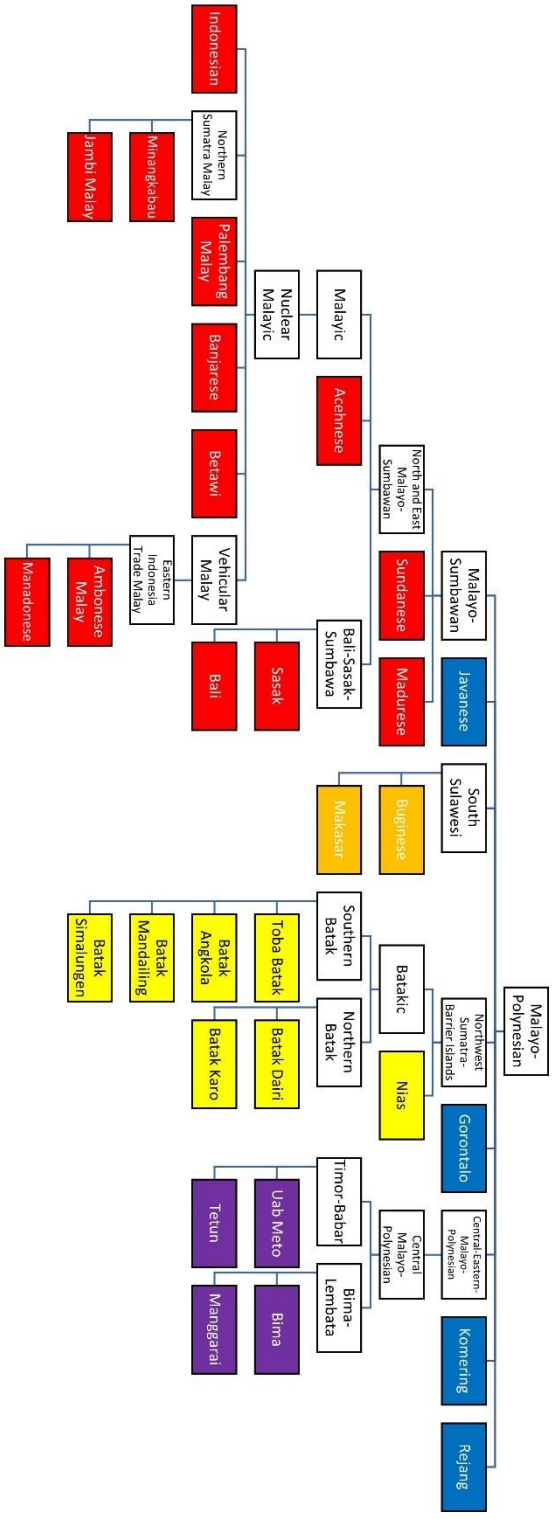

Figure 22 Purity ground truth for k = 5 along with their position in the genetic linguistic tree [47].

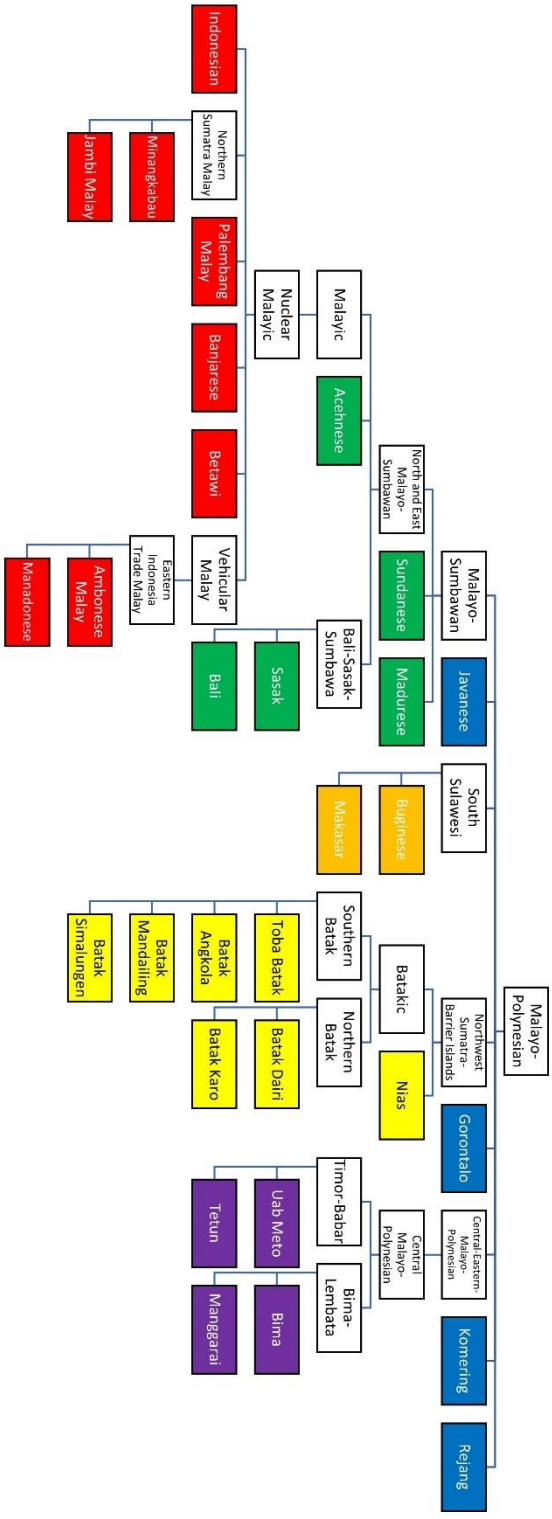

Figure 23 Purity ground truth for k = 6 along with their position in the genetic linguistic tree [47].

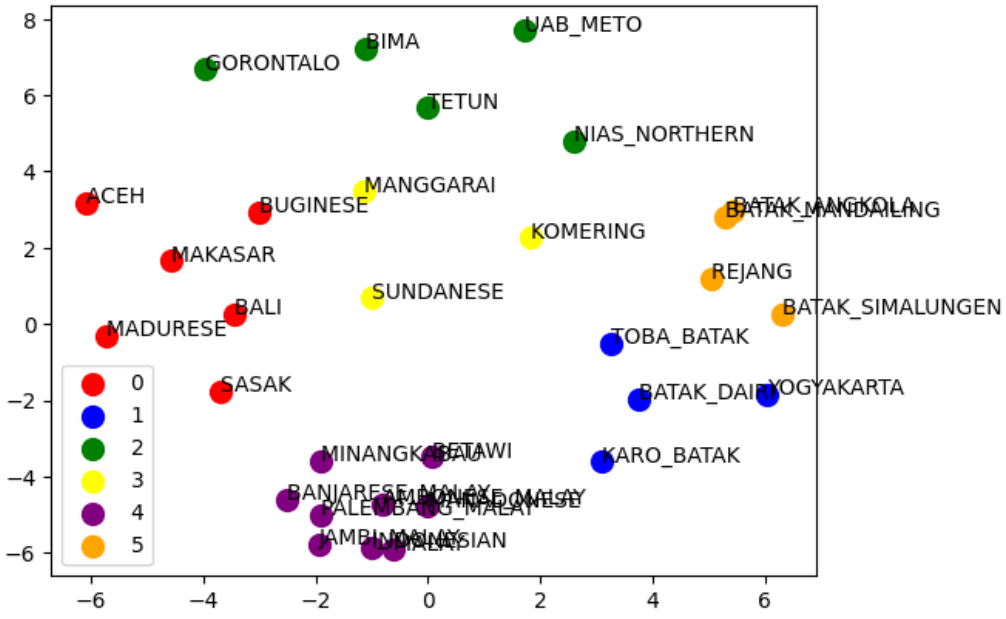

Figure 24 Final clusters for the selected 31 languages.

3.5 Hub Language

In this research, the hub language is defined as the central pivot language which is the source language of all the language pairs. The encoder, which encodes input from this source language, will be reused for all the models. There are three methods to choose the hub language. Two are based on similarity while one is based on dataset availability. Due to issues with the availability of datasets, the purple cluster in Figure 24 of the final set of clusters which contained the Malayic languages was chosen. Only a subset of these languages were considered due to there being data for these languages, namely Indonesian, Malay, Palembang Malay, Banjarese Malay, and Minangkabau.

The first method will henceforth be referred to as the summed distance method. This method to select the hub language involves calculating the sum of all distances to all other languages for each language. The language with the least sum of distances is considered the hub language.

The second method involves the use of medoids, hence the medoid method. The medoid is the data point with the lowest distance to the centre (centroid) of

the cluster [48]. This is done by first calculating the position of the centroid and then calculating the distance to the centroid from each language, choosing the language with the lowest distance.

The third method is based on the availability of the dataset. Specifically, the language with the highest amount of data available as a source language is chosen as the hub language.

The results of the summed distance method point to Malay as the hub language, with the lowest summed distance at 209.08. These results can be seen in Table 11. The results of the medoid method point to Minangkabau as the medoid. While in terms of dataset availability, only Indonesian as the source language is available for each language pair.

Table 11 Summed distances for all the selected languages.

Language	**Summed Distance**
Malay	209.08
Indonesian	213.50
Palembang Malay	231.40
Banjarese Malay	233.58
Minangkabau	252.76

Chapter 4 Multiple Bilingual Dictionary Induction

4.1 Overview

The primary part of the research makes use of a sequence-to-sequence model utilising an LSTM, a Bi-LSTM, character-level one-hot embedding, and hub language encoder reuse to induce bilingual dictionaries.

4.2 Sequence to Sequence Model

A sequence-to-sequence model is a type of machine learning approach which turns one sequence into another sequence. It consists of an encoder which reads input one timestep at a time and transforms it into a vector and a decoder which then transforms the vector one timestep at a time into the desired output sequence [49]. It makes use of Recurrent Neural Networks (RNNs) for the encoder and the decoder and is particularly useful for sequential problems such as machine translation. In that case, it reads words from the input one by one and then predicts the output sentence word by word.

RNNs however have a problem. This is the so-called *vanishing gradient problem*. For longer sequences, "as the error gradients are backpropagated through the RNN, they might shrink exponentially to zero," which makes it difficult for RNNs to learn long-term dependencies [50]. To solve this issue, the Long Short-Term Memory (LSTM) was developed that was capable of learning with long-term dependencies [49]. An illustration can be seen in Figure 25.

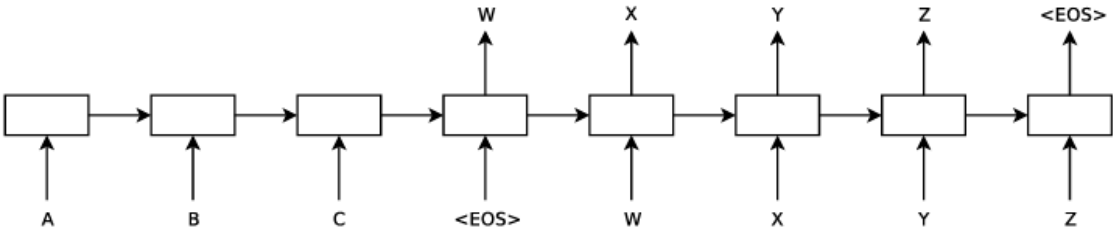

Figure 25 Illustration of a sequence-to-sequence LSTM [49]. The model reads an input sentence “ABC” and produces “WXYZ”. After outputting the end-of-sentence token (<EOS>), it stops making predictions.

4.3 Long-Short Term Memory (LSTM)

Long-Short Term Memory (LSTM) networks are an upgraded variant of Recurrent Neural Network (RNN) architecture designed to solve the issue of vanishing gradients [51]. It can model temporal sequences and long-range dependencies more effectively than traditional RNNs. They are useful in situations where context from previous time steps is important, such as language modeling, speech recognition, and time series prediction.

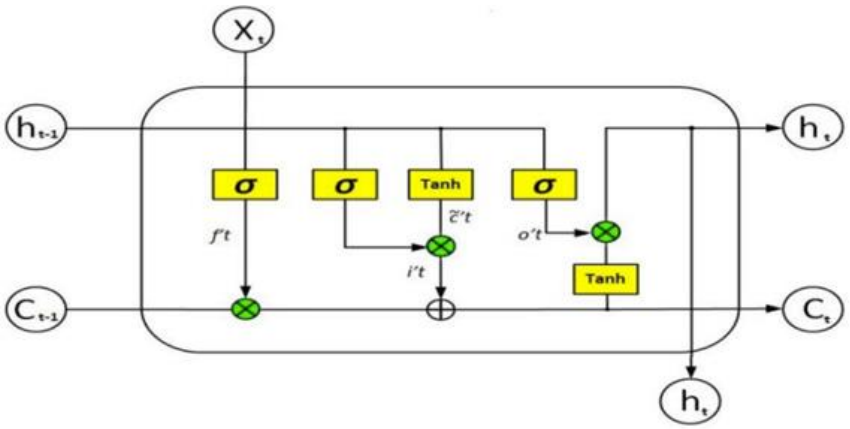

Figure 26 LSTM unit structure [13].

LSTMs solve these limitations of RNNs by using a more complex structure, including mechanisms called gates to regulate the flow of information. An LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. These gates control the flow of information into, out of, and within the cell. The input gate

controls how much of the new information from the current input and the previous output should be added to the cell state, while the forget gate determines how much of the information from the previous output should be retained or forgotten. The output gate controls how much of the cell state should be output to the next time step. These additions allow them to handle long-term dependencies and reduce the vanishing gradient problem.

4.4 Bidirectional Long-Short Term Memory (Bi-LSTM)

Besides the vanishing gradient problem, Recurrent Neural Networks (RNNs) also face another issue. Since they process input in temporal order, the output of RNNs tends to be based mostly on the previous context [52]. The future context is not taken into account. The solution to this problem was the Bidirectional Recurrent Neural Network (BRNN). BRNNs read the input training sequence both forwards and backwards to two separate RNNs which are both connected to the same output layer. Thus, for every point in a sequence, the BRNN has information about the points before and after it.

Bidirectional Long-Short Term Memory (Bi-LSTMs or BLSTMs) work much the way same way. There are two hidden LSTM layers. One reads the input forwards, while one reads the other backwards. They are both connected to the same output layer. Reading the input data from both forward and backward directions makes it particularly effective for tasks where context from both directions is essential, such as in natural language processing.

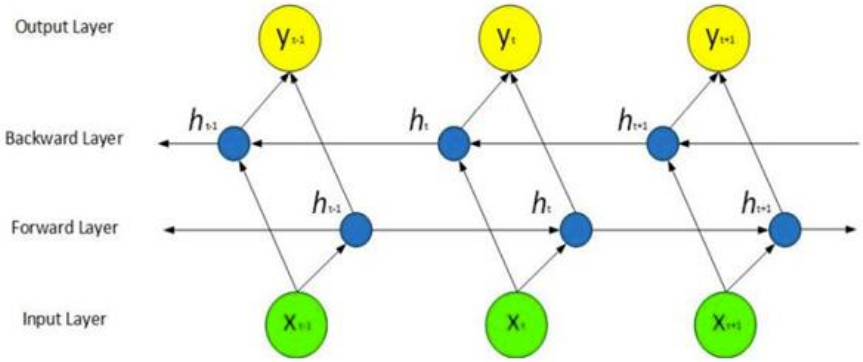

Figure 27 Bi-LSTM architecture [13].

4.5 Character-Level One-Hot Embedding

The tokenisation method used is character-level one-hot embedding. In this process, words are broken down into their individual characters, with each vector having a uniform length based on the total number of characters. This one-hot vector is filled with zeros except for a single entry indicating the character's position in the vocabulary. For example, assuming all 26 characters of the Latin alphabet are present in the corpus, the one-hot vector for the character 'a' would be [1, 0, 0, ..., 0]. For the character 'b', it would be [0, 1, 0, 0, ..., 0], and so forth. This sequence serves as the input for the Bi-LSTM encoder.

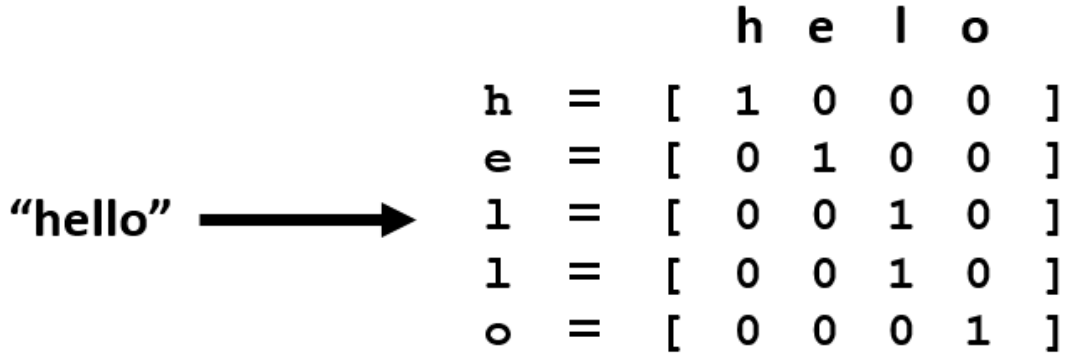

Figure 28 Illustration of character-level one-hot encoding for the word "hello" assuming that there is only "hello" in the entire corpus.

4.6 Hub Language Encoder Reuse

In this research, the encoder of the hub language was reused for all subsequent models. The first model between the hub language and the first target language is trained normally, with a separate encoder and decoder. For training the second model, the first model's encoder is reused as the encoder of the second model. Thus, the weights are updated and not created from scratch. Following that, the third model will reuse the second model's encoder which itself was reused from the first model. The fourth model will do the same with the third model's encoder.

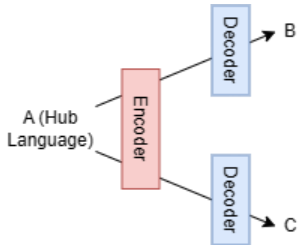

Figure 29 Illustration of the architecture of this research. The encoder, which is trained to encode the input words which are in Indonesian, is reused across several language pairs such that for each language pair they reuse the encoder but have a separate decoder.

Chapter 5 Evaluation/Discussion

5.1 Training Data

Training data was obtained from Nasution et al. and Koto and Koto [28, 53]. The training data is composed of translation pairs between Indonesian and Banjarese Malay, Indonesian and Malay, Indonesian and Minangkabau, and Indonesian and Palembang Malay. Each contains a list of words in Indonesian followed by the same word in the target language, separated by a tab ("\t"). Pre-processing was done to standardise the various datasets into a single template by replacing the " - " variant of "-" which symbolises reduplication with "-" and the removal of the carriage return ("\r") character.

The sizes of the datasets are different. There 10,343 translation pairs between Indonesian and Minangkabau, 5,099 between Indonesian and Palembang Malay, 5,230 between Indonesian and Malay, and 2,029 between Indonesian and Banjarese Malay. Each dataset is divided into two sets, with 80% being used for training and 20% being used for testing. This yields 8,254 Indonesian-Minangkabau translation pairs for training and 2,089 for testing, 4,079 Indonesian-Palembang Malay translation pairs for training and 1,020 for testing, 4,184 Indonesian-Malay translation pairs for training and 1,046 for testing, and 1,621 Indonesian-Banjarese Malay translation pairs for training and 408 for testing.

Table 12 Dataset size for each language pair.

Language Pair	Total Translation Pairs	Training Translation Pairs	Testing Translation Pairs
Indonesian-Minangkabau	10,343	8,254	2,089
Indonesian-Palembang Malay	5,099	4,079	1,020
Indonesian-Malay	5,230	4,184	1,046
Indonesian-Banjarese Malay	2,029	1,621	408

5.2 Parameters

The parameters utilised for the experiments can be seen in the table below. The embedding size is 512 and batch size 64, trained for 120 epochs. The learning rate schedule technique used is learning rate decay. First, an initial learning rate is chosen, then it is reduced progressively according to a scheduler. The learning rate is set at 0.001 and it will decrease by 1% for every epoch after the 15th epoch. A slower learning rate may be desirable to acquire a more optimal set of weights, but training the model will also take more time.

Table 13 Model parameters.

	Value
Embedding Size	512
Epoch	120
Batch Size	64

5.3 K-Fold Cross Validation

Validation was carried out using K-Fold Cross Validation. First, the data is randomly partitioned into k equally size subsets known as "folds", hence k-fold. The value of k is specified by the user. The model is then trained and validated k times. For each iteration, one fold is held out as the validation set while the other $k - 1$ folds are used to train the model. Every data point is used for both training and validation exactly once across all iterations. For example, in the first iteration the model is trained on folds 2 through k and validated on fold 1. In the second iteration, the model is trained on folds 1 and 3 through k and validated on fold 2, and so on. This has the benefit of reducing overfitting by training and validating on different subsets of the data, maximises the use of available data by using it for both training and validation, and provides a better estimate of the model's performance by looking at results from multiple folds to mitigate the effect of random variation in the data splits. In this research, k is set at 5.

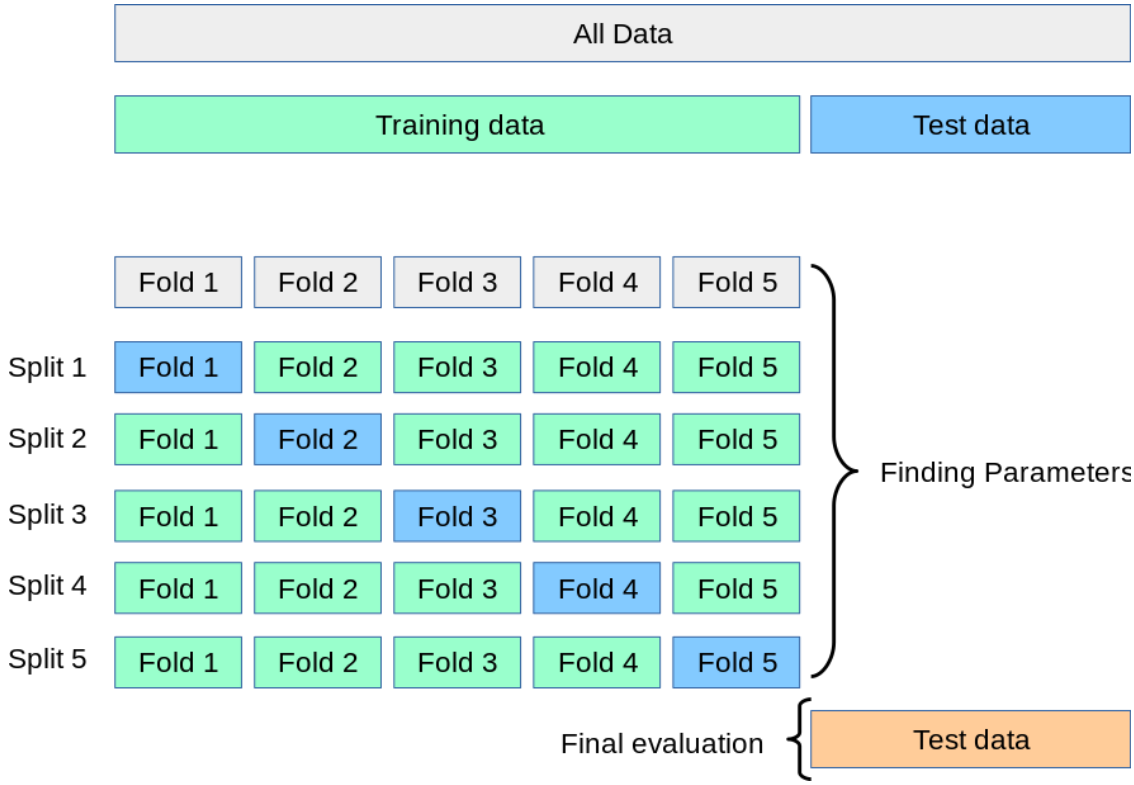

Figure 30 *K*-Fold Cross Validation illustration [54].

5.4 Baseline

As a baseline, four models are trained separately (i.e., without encoder reuse) to serve as a point of comparison for evaluating the performance of the proposed multiple bilingual dictionary induction reusing the hub language encoder.

5.4.1 Description

For all baseline models, the same architecture as the proposed method is used. A standard Bidirectional Long Short-Term Memory (Bi-LSTM) with character-level one-hot encoding was used for the encoder, with each language pair having a separate encoder. For the decoder, a standard Long Short-Term Memory (LSTM) model also with character-level one-hot encoding was used, again with each language pair having a separate decoder. This is exactly the same method used by Resiandi et al. in their experiments [13].

5.4.2 Training and Validation

For all baseline models, training and validation were conducted using the same

datasets as the proposed method. Specifically, datasets for the Indonesian-Banjarese Malay, Indonesian-Malay, Indonesian-Minangkabau, and Indonesian-Palembang Malay language pairs. The Indonesian-Minangkabau dataset is exactly the same dataset used by Resiandi et al. [13]. 80% of the data was used for training, and 20% was used for testing. The validation method used was K-Fold Cross Validation with *k* set to 5.

5.4.3 Results

The results of the baseline models are summarised in Table 11. These results provide a benchmark against which the performance improvements of the proposed method will be measured.

Table 14 Evaluation of the baseline models. IND-BJN refers to Indonesian-Banjarese Malay, IND-ZLM to Indonesian-Malay, IND-MIN to Indonesian Minangkabau, and IND-MUI to Indonesian-Palembang Malay.

Language Pair	**K-Fold Cross Validation Results**					
	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	Average Accuracy
IND-BJN	57.14	61.33	60.84	71.68	71.43	64.48
IND-ZLM	59.37	63.19	67.21	67.40	66.92	64.82
IND-MIN	83.09	84.30	86.00	85.42	85.42	84.85
IND-MUI	58.82	57.75	67.16	68.43	68.14	64.06

5.4.4 Discussion

The results of the baseline Indonesian-Minangkabau model showed an average accuracy of 84.85%, slightly outperforming Resiandi et al.'s results at 83.55% [13]. The results of the other language pairs are about 20% lower than Indonesian-Minangkabau. This is most likely due to the difference in the dataset size, with Indonesian-Minangkabau having about double the dataset size of Indonesian-Malay and Indonesian-Palembang Malay and five times that of Indonesian-Banjarese Malay. Noteworthy is the fact that despite having more than double the dataset size of Indonesian-Banjarese Malay, Indonesian-Malay and Indonesian-Palembang Malay have roughly the same average accuracy. This suggests that increases in the performance of the model lies somewhere between the roughly 5,000 word pairs of Indonesian-Malay and Indonesian-Palembang Malay and the

roughly 10,000 word pairs of Indonesian-Minangkabau.

5.5 Evaluation Results

A total of 18 models were trained according to 6 training orders. The baseline models were used as the first model and its encoder was reused in the subsequent models in the same training order. The effect of language similarity with the source language as well as the size of the dataset on the performance were investigated. The training orders are as follows:

- Random (2 orders)
- Descending based on similarity
- Ascending based on similarity
- Descending based on dataset size
- Ascending based on dataset size
- Descending similarity with largest dataset as the start

The average accuracies of all the results are shown alongside the baseline model performance in Table 12.

Table 15 Average accuracy of all trained models.

Training Order	IND-BJN	IND-ZLM	IND-MIN	IND-MUI
Baseline	64.48	64.82	84.85	64.06
Random 1	65.27	64.82	84.22	64.14
Random 2	64.53	64.49	84.68	64.06
Descending Similarity	64.58	64.82	84.70	64.10
Ascending Similarity	64.38	66.56	84.85	64.02
Descending Dataset Size	64.09	65.26	84.85	64.76
Ascending Dataset Size	64.48	65.60	84.60	64.02
Descending Similarity with Largest Dataset as the Start	65.96	66.24	84.85	64.43

The results for each order will be compared with the baseline. Additionally, the

similarity-based and dataset size-based orders will also be compared with the random orders.

5.5.1 Random 1

The training order of Random 1 is Malay-Minangkabau-Palembang Malay-Banjarese Malay. First, the baseline Indonesian-Malay model is used as the starting point. The Indonesian-Minangkabau model reuses the encoder of Indonesian-Malay, and then its encoder is in turn reused by Indonesian-Palembang Malay, and so on.

Table 16 Average accuracies of the baseline models and Random 1, presented in order of Random 1 training. Language similarities are also shown under the language pair.

	IND-ZLM / 85%	**IND-MIN / 62%**	**IND-MUI / 68%**	**IND-BJN / 72%**
Baseline	64.82	84.85	64.06	64.48
Random 1	64.82	84.22	64.14	65.27
Delta	0.00	-0.63	+0.08	+0.79

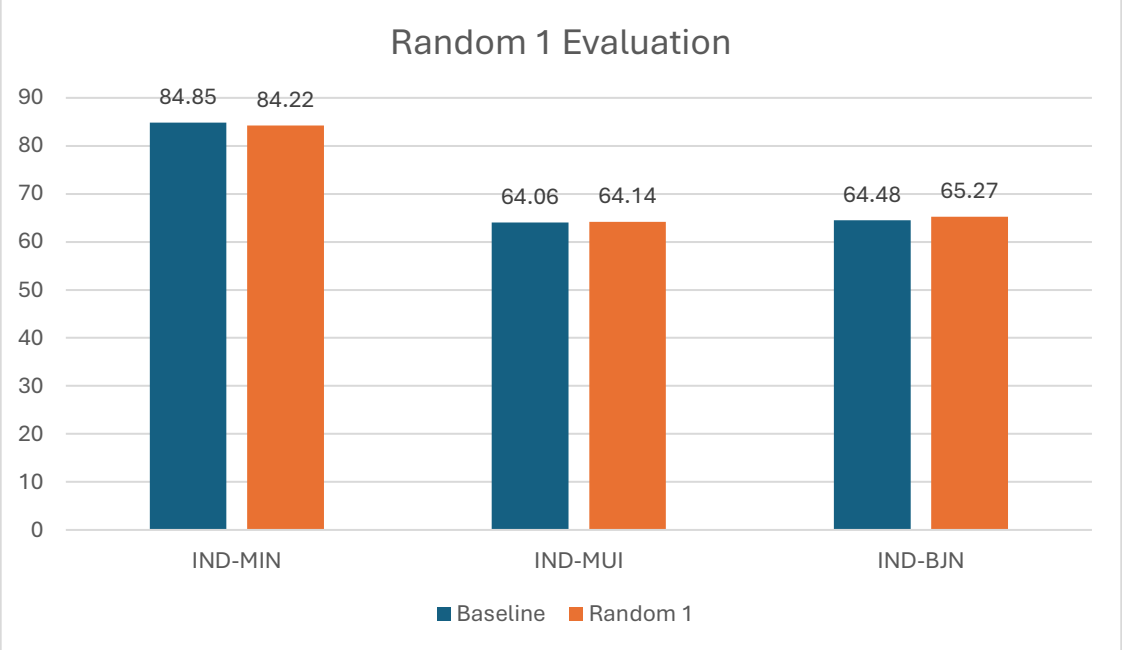

Figure 31 Comparison of the baseline and Random 1 results.

There was a slight decrease in the performance of Indonesian-Minangkabau, while slight increases were evident for Indonesian-Palembang Malay and Indonesian-Banjarese Malay. Things to note include the fact that Indonesian-Minangkabau has the lowest similarity to each other. In terms of dataset size, Indonesian-Minangkabau with around 10,000 word pairs suffers a decrease while those of Indonesian-Palembang Malay with around 5,000 word pairs and Indonesian-Banjarese Malay with around 2,000 word pairs experienced increases.

5.5.2 Random 2

The training order of Random 2 is Palembang Malay-Banjarese Malay-Malay-Minangkabau. First, the baseline Indonesian-Palembang Malay model is used as the starting point. The Indonesian-Banjarese Malay model reuses the encoder of Indonesian-Palembang Malay, and then its encoder is in turn reused by Indonesian-Malay, and so on.

Table 17 Average accuracies of the baseline models and Random 2, presented in order of Random 2 training. Language similarities are also shown under the language pair.

	IND-MUI / 68%	**IND-BJN / 72%**	**IND-ZLM / 85%**	**IND-MIN / 62%**
Baseline	64.06	64.48	64.82	84.85
Random 2	64.06	64.53	64.49	84.68
Delta	0.00	+0.05	-0.33	-0.17

There were decreases for both Indonesian-Malay and Indonesian-Minangkabau, while there was a slight increase for Indonesian-Banjarese Malay. No pattern seems evident from the results of Random 2.

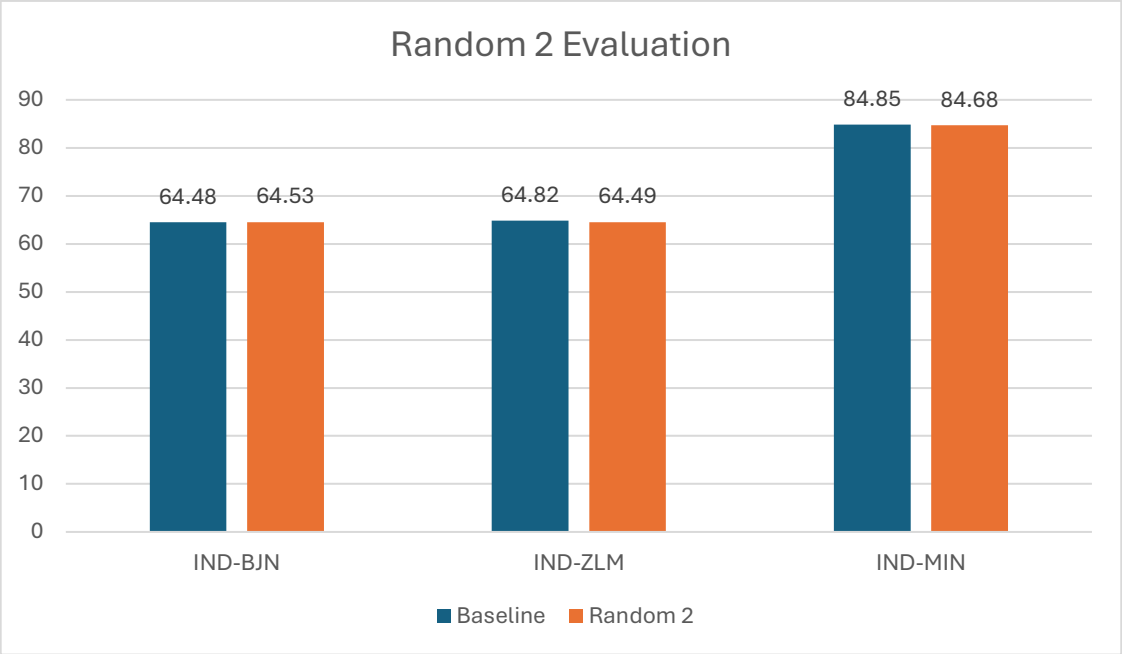

Figure 32 Comparison of the baseline and Random 2 results.

5.5.3 Descending Similarity

The training order of Descending Similarity starts from the language most similar to Indonesian to the least similar. Specifically, the training order is Malay-Banjarese Malay-Palembang Malay-Minangkabau. First, the baseline Indonesian-Malay model is used as the starting point. The Indonesian-Banjarese Malay model reuses the encoder of Indonesian-Malay, and then its encoder is in turn reused by Indonesian-Palembang Malay, and so on.

Compared with the baseline models, Indonesian-Banjarese Malay and Indonesia-Palembang Malay saw slight increases, while the opposite was true for Indonesian-Minangkabau. However, the changes are all very low. These results are similar to comparisons with Random 2. When compared with Random 1, the results were the exact opposite with Indonesian-Banjarese Malay and Indonesian-Palembang Malay seeing decreases while Indonesian-Minangkabau saw a decent increase.

Table 18 Average accuracies of the baseline models, Randoms 1 and 2, and Descending Similarity, presented in order of descending similarity. Language similarities are also shown under the language pair.

	IND-ZLM / 85%	**IND-BJN / 72%**	**IND-MUI / 68%**	**IND-MIN / 62%**
Comparison with the Baseline				
Baseline	64.82	64.48	64.06	84.85
Descending Similarity	64.82	64.58	64.10	84.70
Delta	0.00	+0.10	+0.04	-0.05
Comparison with Random 1				
Random 1	64.82	65.27	64.14	84.22
Descending Similarity	64.82	64.58	64.10	84.70
Delta	0.00	-0.69	-0.04	+0.48
Comparison with Random 2				
Random 2	64.49	64.53	64.06	84.85
Descending Similarity	64.82	64.58	64.10	84.70
Delta	+0.33	+0.05	+0.04	-0.15

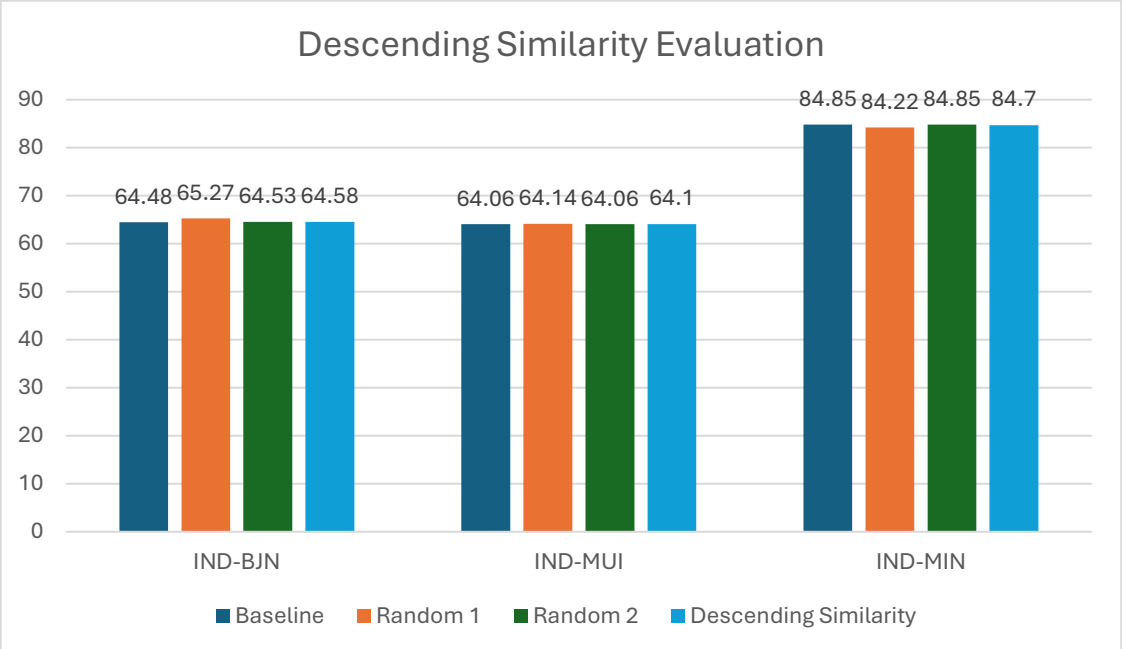

Figure 33 Comparison of the baseline, Randoms 1 and 2, and Descending Similarity results.

5.5.4 Ascending Similarity

The training order of Ascending Similarity starts from the language least similar to Indonesian to the most similar. Specifically, the training order is Minangkabau-Palembang Malay-Banjarese Malay-Malay. First, the baseline Indonesian-Minangkabau model is used as the starting point. The Indonesian-Palembang Malay model reuses the encoder of Indonesian-Minangkabau, and then its encoder is in turn reused by Indonesian-Banjarese Malay, and so on.

Table 19 Average accuracies of the baseline models, Randoms 1 and 2, and Ascending Similarity, presented in order of ascending similarity. Language similarities are also sho`wn under the language pair.

	IND-MIN / 62%	**IND-MUI / 68%**	**IND-BJN / 72%**	**IND-ZLM / 85%**
Comparison with the Baseline				
Baseline	84.85	64.06	64.48	64.82
Ascending Similarity	84.85	64.02	64.38	66.56
Delta	0.00	-0.04	-0.10	+1.74
Comparison with Random 1				
Random 1	84.22	64.14	65.27	64.82
Ascending Similarity	84.85	64.02	64.38	66.56
Delta	+0.63	-0.12	-0.89	+1.74
Comparison with Random 2				
Random 2	84.85	64.06	64.53	64.49
Ascending Similarity	84.85	64.02	64.38	66.56
Delta	0.00	-0.04	-0.15	+2.07

Compared with the baselines, both Indonesian-Palembang Malay and Indonesian-Banjarese experienced slight decreases, but most notable Indonesian-Malay had a significant increase in performance. This is true even when compared with Randoms 1 and 2. This might be due to Indonesian and Malay's high similarity, at 85%.

5.5.5 Descending Dataset Size

The training order of Descending Dataset Size starts from the language with

the most training data to the least. Specifically, the training order is Minangkabau-Malay-Palembang Malay-Banjarese Malay. First, the baseline Indonesian-Minangkabau model is used as the starting point. The Indonesian-Malay model reuses the encoder of Indonesian-Minangkabau, and then its encoder is in turn reused by Indonesian-Palembang Malay, and so on.

Both Indonesian-Malay and Indonesian-Palembang Malay experienced increases in accuracy in this method, while Indonesian-Banjarese Malay saw lower accuracy. The increases in accuracy are quite significant. This is true even when compared with both Random 1 and Random 2. This is possibly due to the influence of dataset size. Specifically, it might be because the model is leveraging the knowledge learnt from the large dataset of Indonesian-Minangkabau. For the case of Indonesian-Banjarese Malay, it might be because it had the least dataset size to begin with.

Table 20 Average accuracies of the baseline models, Randoms 1 and 2, and Descending Dataset Size, presented in descending order of training data size. Language similarities are also shown under the language pair.

	IND-MIN / 62%	**IND-ZLM / 85%**	**IND-MUI / 68%**	**IND-BJN / 72%**
Comparison with the Baseline				
Baseline	84.85	64.82	64.06	64.48
Descending Dataset Size	84.85	65.26	64.76	64.09
Delta	0.00	+0.44	+0.70	-0.39
Comparison with Random 1				
Random 1	84.22	64.82	64.14	65.27
Descending Dataset Size	84.85	65.26	64.76	64.09
Delta	+0.63	+0.44	+0.62	-1.18
Comparison with Random 2				
Random 2	84.85	64.49	64.06	64.53
Descending Dataset Size	84.85	65.26	64.76	64.09
Delta	0.00	+0.77	+0.70	-0.44

5.5.6 Ascending Dataset Size

The training order of Ascending Dataset Size starts from the language with the least training data to the most. Specifically, the training order is Banjarese Malay-Palembang Malay-Malay-Minangkabau. First, the baseline Indonesian-Banjarese Malay model is used as the starting point. The Indonesian-Palembang Malay model reuses the encoder of Indonesian-Banjarese Malay, and then its encoder is in turn reused by Indonesian- Malay, and so on.

Table 21 Average accuracies of the baseline models, Randoms 1 and 2, and Ascending Dataset Size, presented in ascending order of training data size. Language similarities are also shown under the language pair.

	IND-BJN / 72%	**IND-MUI / 68%**	**IND-ZLM / 85%**	**IND-MIN / 62%**
Comparison with the Baseline				
Baseline	64.48	64.06	64.82	84.85
Descending Dataset Size	64.48	64.02	65.60	84.60
Delta	0.00	-0.04	+1.22	-0.25
Comparison with Random 1				
Random 1	65.27	64.14	64.82	84.22
Ascending Dataset Size	64.48	64.02	65.60	84.60
Delta	-0.79	-0.12	+1.22	+0.38
Comparison with Random 2				
Random 2	64.53	64.06	64.49	84.85
Ascending Dataset Size	64.48	64.02	65.60	84.60
Delta	-0.05	-0.04	+1.11	-0.25

Improvements were seen across the board for Indonesian-Malay, while the accuracy of Indonesian-Palembang Malay consistently dropped. For Indonesian-Minangkabau, it was mixed but mostly decreased. There was a slight increase in performance compared to Random 1.

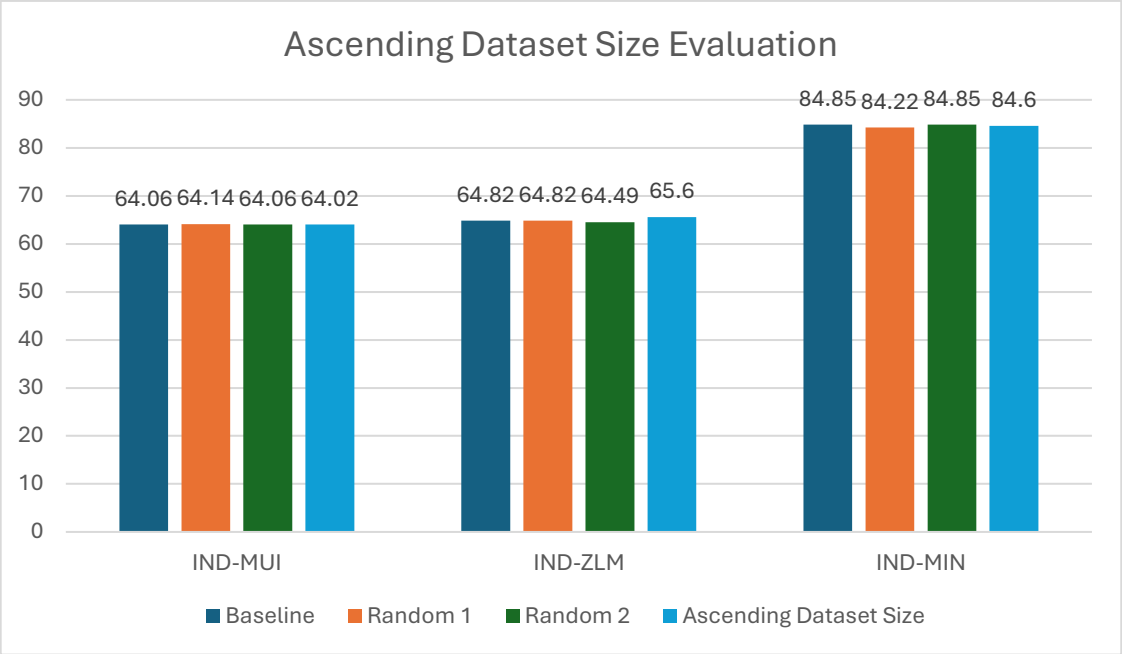

Figure 34 Comparison of the baseline, Randoms 1 and 2, and Ascending Dataset Size results.

5.5.7 Descending Similarity with Largest Dataset as the Start

Based on previous results, dataset size seemed to have a large effect on the accuracy of the models. Therefore, an order combining dataset size and similarity was investigated. Specifically, the first model used is the language pair with the largest amount of data, which is Minangkabau. Afterwards, the languages are in order of descending similarity, which is Malay-Banjarese Malay-Palembang Malay. Indonesian-Malay reused the encoder of the Indonesian-Minangkabau baseline and then Indonesian-Banjarese Malay would in turn reuse the encoder of the Indonesian-Malay model and so on.

This method saw improvements in all aspects. It consistently resulted in the highest-performing models across Indonesian-Malay, Indonesian-Banjarese Malay, and Indonesian-Palembang Malay and performed better than the Baseline, Random 1, and Random 2. Significant improvements were seen for Indonesian-Malay, possibly due to the fact that Indonesian and Malay share an 85% similarity. The trend generally stays true with the less similar languages seeing lower improvements. But it is noteworthy that there were significant improvements for

Indonesian-Banjarese Malay as well despite having the least amount of data.

Table 22 Average accuracies of the baseline models, Randoms 1 and 2, and Descending Similarity with Largest Dataset as the Start, presented in the aforementioned order. Language similarities are also shown under the language pair.

	IND-MIN / 62%	**IND-ZLM / 85%**	**IND-BJN / 72%**	**IND-MUI / 68%**
Comparison with the Baseline				
Baseline	84.85	64.82	64.48	64.06
Descending Similarity with Largest Dataset as the Start	84.85	66.24	65.96	64.43
Delta	0.00	+1.42	+1.48	+0.37
Comparison with Random 1				
Random 1	84.22	64.82	65.27	64.14
Descending Similarity with Largest Dataset as the Start	84.85	66.24	65.96	64.43
Delta	+0.63	+1.42	+0.69	+0.29
Comparison with Random 2				
Random 2	84.85	64.49	64.53	64.06
Descending Similarity with Largest Dataset as the Start	84.85	66.24	65.96	64.43
Delta	0.00	+1.75	+1.43	+0.37

5.5.8 Malay as the Hub Language

To investigate the effect of reusing the encoder of the hub language based on similarity, more models with Malay as the hub language were trained. Unlike the models using Indonesian as the hub language, the models utilising Malay as the hub language have roughly the same amount of data, removing dataset size as a

variable.

Training data was obtained from Nasution et al. [28]. The training data is composed of translation pairs between Malay and Banjarese Malay, Malay and Indonesian, Malay and Minangkabau, and Malay and Palembang Malay. Each contains a list of words in Indonesian followed by the same word in the target language, separated by a tab ("\t"). Pre-processing was done to standardise the various datasets into a single template by replacing the " - " variant of "-" which symbolises reduplication with "-" and the removal of the carriage return ("\r") character. Pairs containing explanations between parentheses, numbers, and the characters "/" and "," were also removed, while capital letters were all made lowercase.

There are 2,961 translation pairs between Malay and Minangkabau, 2,008 between Malay and Palembang Malay, 2,262 between Malay and Indonesian, and 2,002 between Malay and Banjarese Malay. Each dataset is divided into two sets, with 80% being used for training and 20% being used for testing.

The training order of Descending Similarity starts from the language most similar to Malay to the least similar. Specifically, the training order is Indonesian-Palembang Malay-Banjarese Malay-Minangkabau. First, the baseline Malay-Indonesian model is used as the starting point. The Malay-Palembang Malay model reuses the encoder of Malay-Indonesian, and then its encoder is in turn reused by Malay-Banjarese Malay, and so on.

The results show that there is little consistent improvement when compared to the average random results as well as the baselines. The results using Decreasing Similarity are overall similar to the baselines, though significant improvements for Palembang Malay and especially Banjarese Malay were seen when compared with the average random results.

Table 23 Average accuracies of the baseline models, two randoms, and Descending Similarity, presented in the aforementioned order. Language similarities are also shown under the language pair.

	ZLM-IND / 85%	**ZLM-MUI / 73%**	**ZLM-BJN / 71%**	**ZLM-MIN / 62%**
Comparison with the Baseline				
Baseline	64.94%	48.01%	53.17%	25.76%
Descending Similarity	64.94%	47.81%	53.16%	27.09%
Delta	0.00	-0.20	-0.01	+1.33
Comparison with Random Average				
Random Avg	65.59%	45.25%	43.00%	30.46%
Descending Similarity	64.94%	47.81%	53.16%	27.09%
Delta	-0.65	+2.56	+10.16	-3.37

5.5.9 Minangkabau as the Hub Language

More models with Minangkabau as the hub language were also trained. Similarly, the models utilising Mingkabau as the hub language have roughly the same amount of data, removing dataset size as a variable.

Training data was obtained from Nasution et al. [28]. The training data is composed of translation pairs between Minangkabau and Banjarese Malay, Minangkabau and Indonesian, Minangkabau and Malay, and Minangkabau and Palembang Malay. Each contains a list of words in Indonesian followed by the same word in the target language, separated by a tab ("\t"). Pre-processing was done to standardise the various datasets into a single template by replacing the " - " variant of "-" which symbolises reduplication with "-" and the removal of the carriage return ("\r") character. Pairs containing explanations between parentheses, numbers, and the characters "/" and "," were also removed, while capital letters were all made lowercase.

There are 2,961 translation pairs between Minangkabau and Malay, 1,998 between Minangkabau and Palembang Malay, 2,536 between Minangkabau and Indonesian, and 1,974 between Minangkabau and Banjarese Malay. Each dataset is divided into two sets, with 80% being used for training and 20% being used for

testing.

The training order of Descending Similarity starts from the language most similar to Minangkabau to the least similar. Specifically, the training order is Palembang Malay-Indonesian-Malay-Banjarese Malay. First, the baseline Minangkabau-Palembang Malay model is used as the starting point. The Minangkabau-Indonesian model reuses the encoder of Minangkabau-Palembang Malay, and then its encoder is in turn reused by Minangkabau-Malay, and so on.

The results show that there is little consistent improvement when compared to the average random results as well as the baselines. The results using Decreasing Similarity are overall similar to the baselines and averaged randoms.

Table 24 Average accuracies of the baseline models, two randoms, and Descending Similarity, presented in the aforementioned order. Language similarities are also shown under the language pair.

	MIN-MUI / 64%	**MIN-IND / 62%**	**MIN-ZLM / 62%**	**MIN-BJN / 60%**
Comparison with the Baseline				
Baseline	57.10%	45.11%	51.26%	59.54%
Descending Similarity	57.10%	45.51%	55.20%	54.48%
Delta	0.00	+0.40	+3.94	-5.06
Comparison with Random Average				
Random Avg	54.00%	46.61%	55.03%	54.92%
Descending Similarity	57.10%	45.51%	55.20%	54.48%
Delta	+3.10	-1.10	+0.17	-0.44

Chapter 6 Conclusion

It can be argued that reusing the encoder of the hub language can lead to improvements in the automatic bilingual dictionary induction process. In particular, the size of training data available seems to have a large influence on whether it can improve the performance of models reusing the encoder of the hub language. Training first with the language pair with the most training data and then descending from there shows improvements. The accuracy of Indonesian-Malay at 65.26% outperformed the baseline and random orders. Similarly, Indonesian-Minangkabau at 64.76% outperformed the baseline and random orders. Pure similarity-based orders had mixed results but overall did not improve the performance or had negligible improvements, which was also seen when using Malay and Minangkabau as the hub languages.

However, combining dataset size and similarity produced significant results. Using as the first model the language pair with the largest dataset size (Indonesian-Minangkabau) and then training the rest in descending similarity order led to promising results. Across the board, all models improved compared to the baseline and random orders. At 66.24% accuracy, Indonesian-Malay saw improvements ranging from 1.42% to 1.75% compared to the baseline and random orders. Indonesian-Banjarese Malay at 65.96% saw improvements ranging from 0.69% to 1.48% compared to the baseline and random orders. Indonesian-Palembang Malay too at 64.43% saw increases ranging from 0.29% to 0.37%.

In conclusion, reusing the hub language can improve the performance of models taking into account the size of the dataset and similarity of the languages involved. The optimal order is to use the language pair with the largest dataset as the first model and then training based on descending similarity order afterwards.

Acknowledgments

First and foremost, I would like to express my deep gratitude to God for blessing me with the opportunity to pursue this research in Japan and for His blessings throughout it all. Everything was possible due to His grace.

I owe a profound debt to my supervisor, Professor Yohei Murakami, for granting me the chance to conduct this research under his expert guidance. It would not have been possible without his unwavering support. My heartfelt gratitude goes out to Ritsumeikan University and the Konosuke Matsushita Foundation, for without the scholarship they provided none of this would have been possible. I wish to thank the members of the Social Intelligence Laboratory, especially Shella Eldwina Fitri, Zhang Yuxuan, and Ryotaro Yamamoto for their camaraderie, support in our shared endeavours, and assistance in understanding and solving issues in my research. I am also thankful to the other members of the laboratory, especially Assistant Professor Mondheera Pituxcoosuvarn, Ikkyu Nishimura, Mizuki Motozawa, Kantaro Kitagawa, Kenji Matsumoto, Kanaha Mori, Sunada Kaito, and others for their friendship, warm welcome, and invaluable help.

I am deeply grateful to my family for their unwavering love, endless support, and heartfelt prayers, especially my father Victor Herlianto, my mother Debora Muljana, my brother Evan Dwiputra Herlianto, my late grandfather Herlianto, and my grandmother Estiawati. I am also profoundly thankful to my girlfriend Nadya Hartanto for her constant love and encouragement. Additionally, I extend my gratitude to the friends I've made in Japan, especially Marcel Wira, Clervie Beaucousin, Chinatsu Takeda, Vincent Louatron, Benoît Lormeau, Theodosius Sadikin, King Law, Okello John Silas, Thomas Le, Andrew Yap Yuntze, Yunchen Jiang, Jagadish Ankita, Qu Zeping, and Jose Ryan Gonzaga, as well as to my teacher Ayumi Hoki. My thanks go out to St. Agnes' Episcopal Church for welcoming me into their community, especially to Setsuko Reis, John C.J. Chang, Susanna Hontz, Paul Haimes, Debbie Choong, Megumi Kanematsu, Yutaka Matsugu, Rev. Scott Murray, Hiroko Murray, Rev. John Yutaka Kuroda, Yasuko Ikemiyagi, Etsuko Itoh, Rev. Misa Furumoto, and James and Eileen Goltz. I am

also appreciative of Kohei Kikuchi and Miki Ato from the Ritsumeikan University Biwako-Kusatsu Campus International Center for their kindness and unwavering support. My sincere thanks to the Ritsumeikan University Karuta Society and the Otsu Akinotakai Karuta Society for welcoming me and allowing me to play, practice, and improve in competitive karuta.

Lastly, I owe a debt of gratitude to my friends in Indonesia and beyond. I would like to acknowledge those from 7GCraft for their support, namely Kevin Trisnadi, Wryan Cartie Halim, Anthony Gilrandy Theo, William Alex Saputra, Abraham Alvin, Candra Steven, Nicholas Limit, Alfred Shiergetya, Jethro Junaidi, Nicholas Gilbert Samudi, Yoel Gogo, Abraham Simangusong, David Sudjana, Dominick Matthew, Ignasius Irvan, Johannes Elia, Jonathan Hartanto, Reynaldo Lukinanto, Steve Edward Nalasetya, Wincent Liuswinardo, Ryan Nathaniel, Richi Rusli, Farel Arden, and Daniel Agatan. My heartfelt thanks go to my undergraduate university friends William Sebastian, Chandra Delon, Dicky Hertanto, Gerend Tomasouw, Gilbert Chandra, Hans Budiman, Jerico, Nicolas Adicius, Samuel Edsel Fernandez, Adhiwira Lokacarya, and Yosua Maselyno.

I am grateful to everyone who has been part of this journey. Their support, guidance, and friendship have been invaluable and will always be cherished.

References

[1] Central Intelligence Agency, "Country Comparisons - Area," Central Intelligence Agency, 4 July 2024. [Online]. Available: https://www.cia.gov/the-world-factbook/field/area/country-comparison/. [Accessed 6 July 2024].

[2] Republic of Indonesia Ministry of Home Affairs, "Data Kependudukan [Population Data]," Republic of Indonesia Ministry of Home Affairs, 2023. [Online]. Available: https://dukcapil.kemendagri.go.id/page/read/data-kependudukan. [Accessed 7 July 2024].

[3] The Jakarta Post, "16,000 Indonesian islands registered at UN," *The Jakarta Post,* 21 August 2017.

[4] J. D. Legge, J. F. McDivitt, T. R. Leinbach, G. S. Mohamad, O. W. Wolters and A. W. Adam, "Indonesia," Encylopedia Britannica, 4 July 2024. [Online]. Available: https://www.britannica.com/place/Indonesia. [Accessed 6 July 2024].

[5] Badan Pusat Statistik, Kewarganegaraan, Suku Bangsa, Agama, dan Bahasa Sehari-hari Penduduk Indonesia [Nationalities, Ethnicities, Religions, and Everyday Languages of the Indonesian People], Jakarta, 2012.

[6] D. M. Eberhard, G. F. Simons and C. D. Fennig, Eds., Ethnologue: Languages of the World, 27th ed., Dallas, Texas: SIL International, 2024.

[7] M. Florey, Endangered Languages of Austronesia, Oxford: Oxford University Press, 2010.

[8] S. A. Wurm, Ed., Atlas of the World's Languages in Danger of Disappearing, Paris: UNESCO Publishing, 2001.

[9] J. A. Lopo and R. Tanone, "Constructing and Expanding Low-Resource and Underrepresented Parallel Datasets for Indonesian Local Languages," 2024.

[10] Y. Murakami, "Indonesia Language Sphere: an ecosystem for dictionary development for low-resource languages," *Journal of Physics: Conference Series,* vol. 1192, 2019.

[11] A. H. Nasution, Y. Murakami and T. Ishida, "Constraint-Based Bilingual Lexicon Induction for Closely Related Languages," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, 2016.

[12] A. H. Nasution, Y. Murakami and T. Ishida, "A Generalized Constraint Approach to Bilingual Dictionary Induction for Low-Resource Language Families," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP),* vol. 17, no. 2, pp. 1-29, 2017.

[13] K. Resiandi, Y. Murakami and A. H. Nasution, "A Neural Network Approach to Create Minangkabau-Indonesia Bilingual Dictionary," in *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on*

Under-Resourced Languages, Marseille, 2022.

[14] S. Wichmann, E. W. Holman and C. H. Brown, Eds., *The ASJP Database (version 20),* 2022.

[15] P. Fung, "A statistical view on bilingual lexicon extraction," in *Parallel Text Processing*, Springer, 2000, pp. 1-17.

[16] B. Li and E. Gaussier, "Improving corpus comparability for bilingual lexicon extraction from comparable corpora," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010.

[17] K. Tanaka and K. Umemura, "Construction of a Bilingual Dictionary Intermediated by a Third Language," in *Proceedings of the 15th Conference on Computational Linguistics*, 1994.

[18] M. Wushouer, D. Lin, T. Ishida and K. Hirayama, "A Constraint Approach to Pivot-Based Bilingual Dictionary Induction," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP),* vol. 15, no. 1, pp. 1-26, 2015.

[19] B. M. Rowe and D. P. Levine, A Concise Introduction to Linguistics, New York: Routledge, 2015.

[20] Oxford Latin Desk Dictionary, New York: Oxford University Press Inc., 2005.

[21] V. Louatron, Interviewee, *List of words in French*. [Interview]. 16 July 2024.

[22] Cambridge University Press & Assessment, "Cambridge English-Spanish Dictionary," Cambridge University Press & Assessment, 2024. [Online]. Available: https://dictionary.cambridge.org/dictionary/english-spanish/. [Accessed 16 July 2024].

[23] Cambridge University Press & Assessments, "Cambridge English-Portuguese Dictionary," Cambridge University Press & Assessments, 2024. [Online]. Available: https://dictionary.cambridge.org/dictionary/english-portuguese/. [Accessed 16 July 2024].

[24] Cambridge University Press & Assessments, "Cambridge English-Italian Dictionary," Cambridge University Press & Assessments, 2024. [Online]. Available: https://dictionary.cambridge.org/dictionary/english-italian/. [Accessed 16 July 2024].

[25] L. Campbell, Historical Linguistics: An Introduction, Cambridge: The MIT Press, 2013.

[26] C. Gooskens, "The Contribution of Linguistic Factors to the Intelligibility of Closely Related Languages," *Journal of Multilingual and Multicultural Development,* vol. 28, no. 6, pp. 445-467, 2007.

[27] R. A. Blust, "Austronesian languages," Encyclopedia Britannica, 4 July 2024. [Online]. Available: https://www.britannica.com/topic/Austronesian-languages. [Accessed 19 July 2024].

[28] A. H. Nasution, Y. Murakami and T. Ishida, "Plan optimization to bilingual

dictionary induction for low-resource language families," *Transactions on Asian and Low-Resource Language Information Processing,* vol. 20, no. 2, pp. 1-28, 2021.

[29] Balai Bahasa Banjarmasin, Kamus Banjar Dialek Hulu - Indonesia [Hulu Dialect Banjarese - Indonesian Dictionary], Banjarbaru: Balai Bahasa Banjarmasin, 2008.

[30] D. Mulyana, Interviewee, *List of words in Sundanese.* [Interview]. 19 July 2024.

[31] V. Herlianto, Interviewee, *List of words in Javanese.* [Interview]. 19 July 2024.

[32] A. Adelaar and N. P. Himmelmann, Eds., The Austronesian Languages of Asia and Madagascar, Oxon: Routledge, 2005.

[33] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," *Transactions of the Association for Computational Linguistics,* vol. 5, pp. 339-351, 2017.

[34] J. Lee, K. Cho and T. Hofmann, "Fully Character-Level Neural Machine Translation without Explicit Segmentation," *Transactions of the Association for Computational Linguistics,* vol. 5, pp. 365-378, 2017.

[35] D. Dong, H. Wu, W. He, D. Yu and H. Wang, "Multi-Task Learning for Multiple Language Translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, Beijing, 2015.

[36] B. Nothofer, "Javanese," in *Concise Encyclopedia of Languages of the World*, Oxford, Elsevier, 2009, pp. 560-561.

[37] L. Brown, "A Grammar of Nias Selatan," 2001.

[38] The Editors of Encyclopaedia Britannica, "Malay language," Encyclopedia Britannica, 15 July 2024. [Online]. Available: https://www.britannica.com/topic/Malay-language. [Accessed 20 July 2024].

[39] "Guidelines for Preparing 40-word Lists for Languages to be Included in the ASJP Database," 18 September 2007. [Online]. Available: https://asjp.clld.org/static/Guidelines.pdf. [Accessed 20 July 2024].

[40] C. H. Brown, E. W. Holman, S. Wichmann and V. Velupillai, "Automated classification of the world′s languages: a description of the method and preliminary results," *Language Typology and Universals,* vol. 61, no. 4, pp. 285-308, 2008.

[41] International Phonetic Association, Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet, Cambridge: The Press Syndicate of the University of Cambridge, 1999.

[42] S. Wichmann, E. W. Holman, D. Bakker and C. H. Brown, "Evaluating linguistic distance measures," *Physica A: Statistical Mechanics and its Applications,* vol. 389, no. 17, pp. 3632-3639, 2010.

[43] A. H. Nasution, Y. Murakami and I. Toru, "Generating similarity cluster of Indonesian languages with semi-supervised clustering," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 9, no. 1, pp. 531-538, 2019.

[44] J. Cui, J. Liu and Z. Liao, "Research on K-means clustering algorithm and its implementation," Atlantis Press, Paris, 2013.

[45] F. Wickelmaier, "An Introduction to MDS," 2003.

[46] M. S. Hossain, "Evaluation of clustering algorithms: Measure the quality of a clustering outcome," [Online]. Available: https://computing4all.com/courses/introductory-data-science/lessons/evaluation-of-clustering-results/. [Accessed 24 July 2024].

[47] H. Hammarström, R. Forkel, M. Haspelmath and S. Bank, "Glottolog 5.0," Max Planck Institute for Evolutionary Anthropology, Leipzig, 2024.

[48] A. Struyf, M. Hubert and P. Rousseeuw, "Clustering in an Object-Oriented Environment," *Journal of Statistical Software,* vol. 1, no. 4, pp. 1-30, 1997.

[49] I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Montreal, 2024.

[50] A. H. Ribeiro, K. Tiels, L. A. Aguirre and T. B. Schön, "Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020*, Palermo, 2020.

[51] S. Hochreiter, "Long Short-term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-80, 1997.

[52] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks,* vol. 18, no. 5-6, pp. 602-610, 2005.

[53] F. Koto and I. Koto, "Towards Computational Linguistics in Minangkabau Language: Studies on Sentiment Analysis and Machine Translation," in *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*, Hanoi, 2020.

[54] scikit-learn developers, "3.1. Cross-validation: evaluating estimator performance," scikit-learn developers, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Accessed 26 July 2024].