Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6530–6539
Marseille, 20-25 June 2022

Language Identification for Austronesian Languages

Jonathan Dunn, Wikke Nijhof
University of Canterbury, Department of Linguistics
and the New Zealand Institute for Language, Brain and Behaviour
Christchurch, New Zealand
jonathan.dunn@canterbury.ac.nz, wni20@uclive.ac.nz

Abstract

This paper provides language identification models for low- and under-resourced languages in the Pacific region with a focus on previously unavailable Austronesian languages. Accurate language identification is an important part of developing language resources. The approach taken in this paper combines 29 Austronesian languages with 171 non-Austronesian languages to create an evaluation set drawn from eight data sources. After evaluating six approaches to language identification, we find that a classifier based on skip-gram embeddings reaches a significantly higher performance than alternate methods. We then systematically increase the number of non-Austronesian languages in the model up to a total of 800 languages to evaluate whether an increased language inventory leads to less precise predictions for the Austronesian languages of interest. This evaluation finds that there is only a minimal impact on accuracy caused by increasing the inventory of non-Austronesian languages. Further experiments adapt these language identification models for code-switching detection, achieving high accuracy across all 29 languages.

Keywords: language identification, Austronesian languages, code-switching detection, low-resource languages

1. Identifying Pacific Languages

Language identification (LID) remains an important problem within natural language processing because it is a central component in the creation of many corpora. The result is that languages which are not covered by a LID model also lack the data and resources necessary for many applications. This lack of data cannot be solved by bootstrapping methods (Baroni et al., 2009; Goldhahn et al., 2012; Benko, 2014) because it is not possible to identify new samples of the language. This means that accurate LID for minority languages has been and continues to be an important challenge (Jauhiainen et al., 2019), often the first step in developing resources for low-resource languages.

This paper addresses the problem of language identification for the Pacific region, with a special focus on low- and under-resourced Polynesian and Austronesian languages. We provide a LID model capable of accurately identifying 29 Polynesian/Austronesian languages against a selection of languages likely to occur in the Pacific region (200 in total). Of these Austronesian languages, 9 have not been previously available in LID models. An additional challenge is that the Austronesian languages of interest are closely related: similar, for example, to the challenge of identifying related Uralic languages (Chakravarthi et al., 2021). This makes the identification task more difficult because the languages themselves are relatively similar. Even for Austronesian languages that have been included in previous models, then, this work improves our ability to distinguish them from a wider range of closely related languages.

The experiments in this paper first evaluate six LID architectures on an inventory of 200 languages to see which approaches work best with a focus on Austronesian languages. These experiments show that a classifier based on skip-gram character embeddings performs best by a clear margin. We then evaluate this approach against an increasing inventory of languages, from 200 to 800, in order to quantify the trade-off between accuracy and coverage (Majlǐs, 2012; Jauhiainen et al., 2017a).

The problem of language identification is often presented as a trade-off between (i) the number of languages, (ii) the sample size for each document, and (iii) the diversity of data sources included. For example, it is often possible to maintain high accuracy for a very large number of languages if the training and testing data is drawn from a limited set of domains (Brown, 2014). This paper uses data from many different domains and conducts an evaluation on a test set containing 1.1 million samples. The combination of many domains and many test samples helps to ensure that the evaluation represents a realistic context: for example, the application of a LID model in a bootstrapping context (Dunn, 2020) requires accurate identification over a very large number of samples that usually show a highly skewed distribution of labels.

This paper maintains high accuracy at a sample size of 100 characters per document while continuing to cover a large number of Austronesian languages. Previous work has evaluated LID models on sample sizes as small as 40 or 60 characters (Brown, 2014; Jauhiainen et al., 2017b). A smaller sample size like this requires a trade-off, either in reducing the inventory of languages or reducing the diversity of domains used for training and testing. The longer-term goal for this work is to enable bootstrapping methods to develop corpora representing the Pacific region. Such bootstrapped corpora collect data from many domains; thus, our focus is on maintaining a diversity of domains and a large language inventory

Languages	langid.py	CLD3	fastText	polyglot	HeLI	idNet	This Paper
Total	97	107	176	196	285	464	200-800
Austronesian	5	7	9	12	20	10	29

Table 1: Comparison of Total and Austronesian Coverage for Common LID Models

rather than reducing sample size. Further, documents with fewer than 100 characters are often less useful from a corpus-building perspective.

We begin, in Section 2, by reviewing the selection of non-Austronesian languages to include in our initial inventory. We then describe our sources of data (Section 3) and the specific models as they have been implemented here (Section 4). Further, we discuss the evaluation across different models in Section 5 and the evaluation across different language inventories in Section 6. In Section 7 we apply these language identification models to code-switching detection and in Section 8 we evaluate the performance and stability of models after compression.

2. Inventory of Languages

One challenge for the identification of low-resource languages is that there are so many such languages that the inventory can become quite large relative to the amount of training data available. This paper first evaluates different methods on an inventory of only 200 languages, including 171 non-Austronesian languages. This section details previous coverage of the relevant languages as well as how we select the non-Austronesian languages to include in the initial model.

We have two types of constraints: First, any Austronesian language, our main focus, is included in the language inventory by default. This is based on genetic classifications. Second, recent work on digital language mapping (Dunn, 2020; Dunn and Adams, 2020) has used web and social media data to determine the inventory of languages most likely to occur in each region, including the Pacific. We take the non-Austronesian languages that are most common in the Pacific region in this previous work[1] and include them in the initial model. This provides an inventory of 200 languages; 29 of these are Austronesian and the remainder are those frequently observed in the Pacific. A complete list is available in the supplementary material.

In Table 1 this inventory is compared with six common LID packages: Google's CLD3[2], langid.py[3], polyglot[4], fastText[5], idNet[6], and HeLI[7]. Most work that depends on language identification indirectly relies on one or another of these packages. The table shows the total inventory of languages for each model as well as the number of Austronesian languages. The base model evaluated here includes 200 languages in total, more than any package other than idNet and HeLI. The largest inventory, 800 languages, includes more than any of the other packages.

[1]https://www.earthLings.io
[2]https://github.com/google/cld3
[3]https://github.com/saffsd/langid.py
[4]https://github.com/aboSamoor/polyglot
[5]https://fasttext.cc/docs/en/language-identification.html
[6]https://github.com/jonathandunn/idNet
[7]https://github.com/tosaja/HeLI

3. Data

The ground-truth corpora used as samples for each language are taken from several sources, detailed in Table 2. Corpora are split into samples of 100 characters and cleaned using the *clean-text* package[8] to remove URLS, numbers, punctuation, and other non-linguistic characters. We divide the data into training, testing, and validation sets. Within each family of models, discussed in Section 4, we first find the best parameters using the test set and then conduct the evaluation on the validation set. These data sources provide a diversity of domains that is important for ensuring a robust evaluation of LID performance. This also provides a very large training set (over 100 million samples) and a very large testing set (over 1.1 million samples).

Corpus	N. Langs
Bible Translations (Brown, 2014)	614
Global Voices News (Tiedemann, 2012)	41
JW 300 (Agić and Vulić, 2019)	380
Open Subtitles (Lison and Tiedemann, 2016)	62
QCRI Educational Domain (Tiedemann, 2012)	42
Tatoeba Sentences (Tiedemann, 2012)	309
Wikipedia Articles TensorFlow DataSets	280
Māori Broadcasts (Boyce, 2006)	1

Table 2: Sources of Data for LID Models

4. Models

Based on previous work on language identification, we implement six main approaches, a combination of neural and non-neural architectures. While many popular packages rely on neural models (such as Google's CLD3), non-neural models often dominate shared tasks (Chakravarthi et al., 2021). Most work on LID uses character n-grams as features, usually trigrams or ranges of

[8]https://pypi.org/project/clean-text/

	NB, InfoGain		SVM, InfoGain		MLP, InfoGain		MLP, Hashing		fastText	
Language	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
W. Average	*0.95*	*0.94*	*0.96*	*0.95*	*0.95*	*0.93*	*0.96*	*0.95*	*0.99*	*0.99*
Acehnese	0.99	0.98	1.00	0.90	1.00	0.84	0.99	0.96	1.00	0.99
Buginese	1.00	0.94	1.00	0.92	1.00	0.93	1.00	0.94	1.00	0.98
Cebuano	0.95	0.88	0.98	0.99	0.69	0.99	0.69	0.89	1.00	1.00
Chamorro	1.00	0.64	1.00	0.31	0.99	0.91	1.00	0.93	0.96	1.00
Chuukese	1.00	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00
Fijian	1.00	0.92	1.00	0.93	1.00	0.96	1.00	0.94	1.00	1.00
Gilbertese	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.97	1.00	1.00
Hawaiian	1.00	0.99	1.00	0.97	0.99	0.99	0.97	1.00	1.00	1.00
Hiligaynon	0.64	0.98	0.97	0.98	0.05	0.00	0.73	0.01	0.99	1.00
Hiri Motu	1.00	1.00	1.00	1.00	0.98	1.00	0.99	1.00	1.00	1.00
Ilocano	0.97	0.97	0.99	0.98	0.99	0.97	0.99	0.97	1.00	0.99
Javanese	0.66	0.98	0.83	0.97	0.99	0.87	1.00	0.74	0.97	0.99
Marshallese	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
Malagasy	0.99	0.99	1.00	1.00	0.98	1.00	0.95	1.00	1.00	1.00
Māori	0.92	0.99	1.00	0.98	0.74	1.00	0.68	0.99	1.00	1.00
Malay	0.97	0.98	0.95	0.99	0.91	0.98	0.84	0.97	0.97	0.99
Niuean	1.00	1.00	1.00	1.00	0.94	1.00	0.97	0.96	1.00	1.00
Pangasinan	0.98	0.86	0.98	0.94	0.99	0.93	0.98	0.89	1.00	0.97
Pohnpeian	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C.I. Māori	0.99	1.00	0.99	0.99	0.55	0.01	0.38	0.00	1.00	1.00
Samoan	1.00	0.97	1.00	0.98	0.96	0.98	0.99	0.96	1.00	0.99
Sundanese	0.97	0.93	0.95	0.93	0.88	0.91	0.94	0.80	1.00	0.96
Tahitian	0.99	0.72	0.99	0.84	1.00	0.94	1.00	0.81	1.00	1.00
Tagalog	0.94	0.98	0.95	0.98	0.98	0.92	0.89	0.97	0.99	0.99
Tongan	1.00	0.92	1.00	0.92	0.86	0.93	0.90	0.93	1.00	0.97
Tuvaluan	1.00	1.00	1.00	1.00	1.00	0.87	1.00	0.86	1.00	1.00
Waray	0.97	0.86	1.00	1.00	1.00	0.94	0.99	0.89	1.00	1.00
Wallisian	1.00	1.00	1.00	0.99	0.97	0.56	0.95	0.61	1.00	1.00
Yapese	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00

Table 3: Break-Down of LID Performance by Model and Language (200 Language Inventory)

n that include trigrams. Some work has then focused on the problem of feature selection across all possible character n-grams to produce a useful feature space (Lui and Baldwin, 2011). Here we implement a similar information gain feature selection method, experimenting with the number of features and the n-gram range on the test set. That same feature set is then used across all relevant classifiers. Here the best variant uses information gain to choose the top 75k character trigrams.

First, we implement a feed-forward network that is similar to both CLD3 and idNet (Dunn, 2020). The specific architecture of this network is determined on the test set, ultimately containing two layers of 500 ReLU neurons together with a softmax prediction layer. This is listed as *MLP, InfoGain* in Table 3. The specific implementations of this first model and the second model (below) are provided in the supplementary material.

Second, recent work has shown that feature hashing for character n-grams works well for language identification (Malmasi and Dras, 2017; Dunn, 2020), sometimes better than class-based feature selection. For example, two of the comparison systems in Table 1 use feature hashes within a feed-forward network with softmax for predictions (CLD3, idNet). Here the best hash-based classifier uses a hashing space with 150k bins within a feed-forward network with one layer of 200 neurons and a softmax prediction layer. This is listed as *MLP, Hashing* in Table 3. The substantial difference between these first and second variants, then, is whether the character n-grams are derived from feature selection or from a hashing algorithm; the code for both is available in the supplementary material.

Third, an approach based on skip-gram character embeddings, as implemented in the fastText package, has been shown to be effective for language identification (Joulin et al., 2017). Here the best model is based on character skip-gram embeddings, with n-grams ranging from 1 to 4 with 300 dimensions and 100 negative samples. This is listed as *fastText* in Table 3.

Fourth, recent results from the VarDial evaluation campaign focus on a related problem for Uralic languages, in which 29 region-specific languages of interest are combined with 149 non-relevant languages for the task of distinguishing similar languages (Chakravarthi et al., 2021). While Uralic languages are quite different from Austronesian languages, the underlying problem

Languages	Prec. (All)	Prec. (Pacific)	Rec. (All)	Rec. (Pacific)	F1 (All)	F1 (Pacific)
200	0.994	0.995	0.994	0.993	0.994	0.994
300	0.973	0.956	0.973	0.994	0.969	0.974
400	0.970	0.963	0.970	0.988	0.965	0.973
500	0.969	0.967	0.970	0.989	0.965	0.977
600	0.971	0.979	0.963	0.935	0.960	0.946
700	0.969	0.964	0.968	0.993	0.964	0.976
800	0.966	0.970	0.968	0.979	0.962	0.974

Table 4: Decreasing Performance With Increasing Language Inventory

remains comparable. In this evaluation campaign, many of the best approaches used non-neural classifiers such as Naive Bayes or Support Vector Machines together with character n-grams. We evaluate both of these approaches in Table 3, each using the same feature set as the first feed-forward network. While these classifiers have worked well on problems with limited training data, the training set here contains over 100 million samples. SVMs, in particular, are difficult to train in this setting. Thus, we create a more practical sub-set of the training set, containing 50 million samples (for Naive Bayes) and 1 million samples (for the SVM). These variants are listed as *NB* and *SVM* in Table 3.

Fifth, sequence-based models have used LSTM networks to make predictions directly on sequences of characters, avoiding the need for selecting character n-grams as features (Jaech et al., 2016; Kocmi and Bojar, 2017). These models have typically worked with a much smaller inventory of languages, often with a focus on code-switching. Here we have experimented with several variants of a sequence-based LSTM network. However, none of these variants achieved a competitive accuracy. Thus, we focus our evaluation on the five methods above, all based on character n-grams: a feed-forward network, Naive Bayes, and a Support Vector Machine, all with feature selection; a feed-forward network with feature hashing; and a classifier based on skip-gram character embeddings.

5. Evaluation

We first evaluate each of these models using precision and recall, as shown in Table 3. The first row shows the weighted average across all 200 languages. The data is drawn from multiple sources, so that high-resource languages which appear in each source tend to contribute more test samples. The total validation set contains over 1.1 million samples, with each language ranging from 2k to 16k samples. Given the weighted precision and recall, the SGNS-based classifier represented by fastText out-performs the other models, reaching 0.99 for both precision and recall.

The table also shows each of the Austronesian languages separately in order to determine whether this average performance is representative of these low-resource languages. For example, the feature selection MLP model performs poorly for several languages: Cook Islands Māori and Hiligaynon most prominently. And the feature hashing MLP model produces poor results for Javanese and Wallisian. The SGNS model, however, never falls below 0.96 for any Austronesian language. Thus, not only does the model provide overall accurate predictions at a small sample size, but that performance remains robust across the low-resource languages we are concerned with. In this setting, neither the Naive Bayes nor the SVM models exhibit the best performance.

6. Influence of Inventory Size

How well would this approach have worked if we had instead chosen an inventory of 300 or 500 or 800 languages? We evaluate this in Table 4; each model contains the same languages as previous models together with a selection of additional languages. Languages are added in order of the number of samples available in the training data, so that the later languages are those with the fewest available samples. The models with 200 and 800 languages are made available for further use.[9]

The final column in the table shows the weighted f-score, which decreases from 0.994 to 0.962 (all languages) and 0.974 (Austronesian languages) as the number of languages increases. This would seem to show a relatively minor reduction in performance given increased coverage. For example, this performance remains higher than the other methods evaluated in Section 5. While precision and recall show a similar slight reduction, there is variation in the precision for Austronesian languages: the metric both increases and decreases. This indicates that the performance of the SGNS architecture is somewhat unstable, perhaps caused by variability in this type of embedding (Burdick et al., 2021). The fact that the smaller set of Austronesian languages is more subject to variation indicates that variation in the main inventory of languages is averaged out. This variability in performance is investigated further in Section 8; it remains the case that the best-performing model for identifying Austronesian languages is the SGNS approach.

This section has evaluated models for identifying a set of closely related and previously unavailable Austronesian languages. The evaluation has shown that, while the SGNS model is subject to some variation, it remains the best performing approach against other models, with an f-score of 0.974 when evaluated with a total inventory of 800 languages.

[9] https://www.jdunn.name/corpora/

awesome	video	diaries	ka	mau	te	wehi	e	te	whanau
ENG	ENG	ENG	MRI	MRI	MRI	MRI	MRI	MRI	MRI

Table 5: Code-Switching, English and te reo Māori

	MLP, Selection	**fastText**	**fastText (Small)**
Average Across 19 Languages	97.81%	99.07%	99.13%

Table 6: Accuracy of Short-Span Detection by Model (20 characters)

7. Code-Switching Detection

An additional problem closely related to language identification is the detection of code-switching within samples (Solorio et al., 2014; Molina et al., 2016). For example, Table 5 shows a tweet which is roughly one-third English (ENG) and two-thirds te reo Māori (MRI). This type of code-switching is especially common in digital contexts like social media, the precise contexts from which most training sets are derived. Corpora in te reo Māori often contain a significant amount of English words, as do corpora in many other Austronesian languages. For language identification to be useful for the purpose of building corpora, it is important to be able to also identify code-switching within a corpus. This section presents experiments in code-switching detection for Austronesian languages based on the language identification models presented in Section 4.

Given the previous approach to language identification at 100-character spans, the basic approach taken to code-switching detection is based on disaggregation: *First*, we train language identification models for English and each of the 29 languages in Table 7; these models include only two languages, but have a much shorter character span of 15-characters. We evaluate the performance of short-span detection by model type in Table 6 and then by language for the best model in Table 7. *Second*, we use predictions on overlapping spans to convert this span-based prediction to a word-by-word prediction. This section first evaluates different models for short-span language identification before presenting and evaluating two algorithms for converting short-span identification into word-by-word code-switching detection for Austronesian languages.

Language	**Prec.**	**Rec.**	**Support**
Acehnese	0.99	0.98	11,005
Buginese	0.98	0.99	10,547
Cebuano	1.00	1.00	33,112
Chamorro	1.00	0.99	11,061
Chuukese	1.00	1.00	22,518
Fijian	1.00	1.00	11,185
Gilbertese	1.00	1.00	11,225
Hawaiian	1.00	0.99	22,557
Hiligaynon	1.00	0.99	11,133
Hiri Motu	1.00	1.00	22,593
Ilocano	0.99	0.98	21,620
Javanese	0.99	0.98	22,094
Marshallese	1.00	1.00	11,403
Malagasy	0.99	0.99	32,220
Māori	1.00	0.99	34,023
Malay	0.99	0.98	43,967
Niuean	1.00	1.00	11,325
Pangasinan	1.00	0.95	22,011
Pohnpeian	1.00	1.00	22,182
C.I. Māori	1.00	1.00	11,425
Samoan	1.00	0.98	34,053
Sundanese	0.98	0.96	21,967
Tahitian	1.00	1.00	11,537
Tagalog	0.99	0.98	55,022
Tongan	1.00	0.94	33,790
Tuvaluan	1.00	1.00	11,468
Waray	0.99	1.00	21,422
Wallisian	1.00	1.00	11,455
Yapese	1.00	1.00	23,034

Table 7: Performance of Short-Span Detection by Language with fastText (compressed models, 15 characters)

7.1. Evaluating Short-Span Identification

We begin by evaluating the accuracy for the top two models for language identification when applied to the short-span identification task. Here the short-span task involves distinguishing between two languages (English and an Austronesian language) with a window of 15 characters. The idea behind this algorithm is to support corpus cleaning: first, we identify that a particular document belongs to an Austronesian language; second, we search for English material within that document.

The results for this short-span identification task are shown in Table 6; given the performance reported above, we focus on fastText and the MLP with feature selection methods as the most promising. As before, the average performance for fastText is slightly higher across all Austronesian languages, with 99% vs 97.8%. There are several instances where fastText performs significantly higher than the MLP: Javanese (+6.04%), Malay (+3.87%), and Sundanese (+8.65%). We also include a compressed fastText model for comparison to ensure that model size does not impact prediction accuracy.

After comparing the performance on the short-span task, then, we proceed with code-switching detection using the fastText models. We show the evaluation of compressed fastText models in Table 7 at a span-size of 15 characters. This table shows the precision and recall specifically for the Austronesian languages as well as the number of test samples for each language. The

	Variables
	sample = Current document or sentence *word* = Unit separated by white space after tokenization *character* = Individual symbol within word *character span* = Window of 15 characters that ignores word boundaries *trigram context* = Current word along with one previous word and one following word

	Algorithm 1: Overlapping Character Spans
1	For each word in sample:
2	For each character in word:
3	$p(lang)$ = Probability of English predicted for current character span
4	Return Mean($p(lang)$) = Average probability of English across entire word

	Algorithm 2: Word-Based Detection
1	$p(span)$ = Probability of English for current sample
2	For each word in sample:
3	$p(word)$ = Probability of English for current word alone
4	$p(trigram)$ Probability of English for trigram context around current word
5	If $p(span)$ closer to English:
6	Return max($p(word)$, $p(trigam)$)
7	Else If $p(span)$ further from English:
8	Return min($p(word)$, $p(trigam)$)

Table 8: Code-Switching Detection Algorithms

performance here is consistently high, with only two languages falling below 0.99 precision (Buginese and Sundanese). Tongan is the only language with a recall below 0.95, so that some samples from Tongan are identified as English instead.

7.2. Algorithms for Word-Level Prediction

The short-span prediction task evaluated above allows us to perform language identification on very small samples, but it does not directly provide code-switching detection at the word-level. We therefore evaluate two approaches to converting short-span predictions into word-level predictions, as shown in Table 8.

The first algorithm is based on averaging predictions across overlapping character spans. *First*, the algorithm tokenizes the sample into words and iterates over words; this is required to make predictions about the language for each word in the sample. *Second*, for each word, the algorithm predicts the probability that each span centered on a character within that word belongs to either English or the Austronesian language, using the short-span identification model. Thus, a word containing five characters is represented by five character spans. The algorithm then returns the average probability across all spans as the word's overall value.

The examples in (a) through (d) show a selection of the character spans that are queried for each word. Cases like *awesome* in (a) and *mau* in (d) are straight-forward in the sense that only words from one language fall within the 15-character span. The challenge for this algorithm comes from examples like (b) and (c), in which the character spans around the word contain code-switching within them. A baseline algorithm in the evaluation simply queries each word in the short-span model directly. The intuition behind this first span-based algorithm is that words common in both languages (for example, *i* and *he* in the case of ENG-MRI code-switching) will be identified using information about their immediate context as well.

(a) *awesome* = awesome_video_diarie
(b) *diaries* = video_diaries_ka_ma_
(c) *ka* = _diaries_ka_mau_te_w
(d) *mau* = aries_ka_mau_te_wehi

Language	**Alg. 1**	**Alg. 2**	**Baseline**
Cebuano	24%	100%	94%
Chamorro	96%	100%	100%
Fijian	54%	98%	48%
Hawaiian	64%	100%	94%
Ilocano	74%	100%	100%
Javanese	92%	100%	100%
Malagasy	50%	100%	100%
Māori	40%	100%	60%
Malay	96%	100%	100%
Samoan	34%	100%	44%
Sundanese	92%	100%	100%
Tahitian	56%	98%	86%
Tagalog	94%	100%	100%
Tongan	70%	100%	68%
Average	**67%**	**100%**	**85%**

Table 9: Accuracy of Identified English Spans in Wikipedia

The second algorithm in Table 8 takes a word-level approach: the algorithm iteratres over each word in the sample, as before. In this case, however, the prediction

	Original, 200L		100d, Ftz, 200L		200d, Ftz, 200L		100d, Ftz, 800L		200d, Ftz, 800L	
Language	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
W. Average	*0.99*	*0.99*	*0.99*	*0.98*	*0.99*	*0.98*	*0.95*	*0.95*	*0.95*	*0.95*
Acehnese	1.00	0.99	0.99	0.99	0.99	0.99	0.97	1.00	0.99	0.99
Buginese	1.00	0.98	0.99	0.98	0.99	0.98	0.99	0.97	0.98	0.98
Cebuano	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00
Chamorro	0.96	1.00	0.75	1.00	0.67	1.00	0.58	1.00	0.52	1.00
Chuukese	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Fijian	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Gilbertese	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Hawaiian	1.00	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00
Hiligaynon	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.90
Hiri Motu	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00	0.99	1.00
Ilocano	1.00	0.99	1.00	0.99	1.00	0.99	0.99	0.99	1.00	0.99
Javanese	0.97	0.99	1.00	0.99	0.98	0.99	0.96	0.96	0.98	0.92
Marshallese	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Malagasy	1.00	1.00	0.99	1.00	0.99	1.00	0.99	1.00	1.00	1.00
Māori	1.00	1.00	1.00	0.99	0.99	1.00	1.00	0.99	1.00	0.99
Malay	0.97	0.99	0.98	0.99	0.99	0.99	0.96	0.97	0.85	0.99
Niuean	1.00	1.00	0.99	1.00	0.99	1.00	1.00	0.99	1.00	1.00
Pangasinan	1.00	0.97	1.00	0.97	0.99	0.97	0.99	0.97	1.00	0.96
Pohnpeian	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
C.I. Māori	1.00	1.00	1.00	1.00	1.00	0.99	0.99	1.00	1.00	1.00
Samoan	1.00	0.99	1.00	0.99	1.00	0.99	1.00	0.98	1.00	0.98
Sundanese	1.00	0.96	0.99	0.97	0.99	0.97	0.95	0.97	0.98	0.96
Tahitian	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Tagalog	0.99	0.99	0.99	0.99	0.99	0.99	0.78	0.88	0.77	0.97
Tongan	1.00	0.97	1.00	0.94	1.00	0.95	0.99	0.96	1.00	0.95
Tuvaluan	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00
Waray	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00
Wallisian	1.00	1.00	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00
Yapese	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table 10: Break-Down of LID Performance After Model Compression

is based mainly on word contexts rather than character spans that ignore words. *First*, a baseline prediction is made on the entire sample. *Second*, a prediction is made about the current word in isolation. *Third*, a trigram prediction is made that includes the previous word and the following word. The algorithm combines these two predictions, the word and its context. If the overall sample is predicted to be English, the prediction closest to English is taken. But, if the overall sample is predicted to not be English, preference is given to a non-English prediction.

These two algorithms are used to convert span-level language predictions (covering 15-characters) into word-level predictions. Given the focus of the first portion of the paper on span-level language identification, the goal here is to enable code-switching detection for Austronesian languages using the same set of models.

7.3. Evaluating Code-Switching Detection

We evaluate the two algorithms described above using Wikipedia corpora that represent 14 Austronesian languages. These Wikipedia corpora are written mainly in the language of interest, but English terms remain in nearly every document. We therefore use the two code-switching detection algorithms and a baseline algorithm to predict sequences of three or more English words and then evaluate the accuracy of these sequences.

For each language, we retrieve the first 50 English phrases identified by each algorithm. We then use the short-span identification model to evaluate whether those spans as a whole are English (as they were predicted to be). The results are shown in Table 9. The column *Alg. 1* refers to the span-based algorithm in Table 8. The column *Alg. 2* refers to the word-based algorithm. And the column *Baseline* refers to a simple model which directly queries the short-span model with each individual word (thus not taking any further context into consideration).

The word-based context algorithm has the highest accuracy, followed by the word-by-word baseline, with the character-span algorithm performing poorly overall. Three representative examples for the ENG-MRI model are shown below. For the character-based algorithm, words belonging to MRI can be included in the predicted English span if the English words are rather long. For the word-based algorithm, almost all predicted English

phrases are fully English, as shown in the example. Finally, the baseline – which simply queries each word in isolation – always identifies words like *i* and *he*, which could belong to either language, as belonging to English (which has more training data). Thus, sequences like the example below, when viewed as words in isolation, are identified incorrectly as English phrases.

(Algorithm 1) *tuatahi chant from the*
(Algorithm 2) *and are used to generate*
(Baseline) *i kōiwi hua he*

The evaluation in this section has shown that the underlying language identification models used here can also be adapted for short-span identification in a way that supports detection of code-switching in Austronesian languages. This is an important practical tool for supporting corpus creation because it allows us to control the proportion of a document that is in English. These models are available here: https://github.com/jonathandunn/pacific_CodeSwitch

8. Model Size and Model Stability

Using fastText models leads to two practical challenges: first, the size of these models can become quite large (Joulin et al., 2016); second, models based on skip-gram embeddings are subject to instability (Dunn et al., 2022). A number of techniques for reducing model size are available, but each such technique has a chance of also reducing the performance of the model. This section provides an analysis of model performance under different compression strategies as well as the stability of the performance of the resulting models.

The results for compressed models are shown in Table 10 for both 200-language and 800-language inventories. The leftmost column represents the original model (approximately 28 gb); the remaining columns evaluate lower dimensional models (100d, 200d) with 200-languages (200L) and 800-languages (800L). Each of these small models uses a minimum count threshold of 5 as well as having undergone quantization (represented here as *Ftz*). These smaller models range from 600mb (200 languages with 100 dimensions) to 1.2gb (800 languages with 200 dimensions).

With 200 languages in the model, there is a slight decline in recall (from 0.99 in the original to 0.98 in both compressed models). Within Austronesian languages, this impact only is relevant for Tongan, which declines from 0.97 to 0.94 in recall. With 800 languages, there is a decline after compression to 0.95 precision and recall, compared with 0.96 in the original model in Table 4. This slight decline in performance is necessary to have a model size which is small enough for practical use. The final models are available here: https://jdunn.name/corpora/.

Given previous work showing the instability of the skip-gram embeddings that fastText uses, we might think that the performance of the language identification models here would also be unstable. To evaluate this we train five alternate versions of the 200 language compressed model and show the range of precision and recall values in Table 11 for each of the Austronesian languages. This experiment shows that two languages have a rather wide range of performance: Chamorro (which ranges from 0.60 to 0.76 precision), Sundanese (which ranges from 0.88 to 0.99 precision), and Hiligaynon (which ranges from 0.85 to 1.00 recall). Thus, the same instability which has an influence on unsupervised embeddings has an influence on supervised embeddings. In this case, we are able to select the model which has the best performance across all Austronesian languages.

Language	**Precision**		**Recall**	
	Min	*Max*	*Min*	*Max*
Acehnese	0.97	1.00	0.99	0.99
Buginese	0.99	1.00	0.98	0.98
Cebuano	0.99	1.00	1.00	1.00
Chamorro	0.60	0.76	1.00	1.00
Chuukese	1.00	1.00	1.00	1.00
Fijian	1.00	1.00	1.00	1.00
Gilbertese	1.00	1.00	1.00	1.00
Hawaiian	1.00	1.00	1.00	1.00
Hiligaynon	1.00	1.00	0.85	1.00
Hiri Motu	1.00	1.00	1.00	1.00
Ilocano	1.00	1.00	0.99	0.99
Javanese	0.97	1.00	0.99	0.99
Marshallese	1.00	1.00	1.00	1.00
Malagasy	0.99	1.00	1.00	1.00
Māori	0.99	1.00	0.97	1.00
Malay	0.96	0.99	0.99	0.99
Niuean	0.99	1.00	1.00	1.00
Pangasinan	0.99	1.00	0.96	0.97
Pohnpeian	1.00	1.00	1.00	1.00
C. I. Māori	0.97	1.00	1.00	1.00
Samoan	1.00	1.00	0.99	0.99
Sundanese	0.88	0.99	0.96	0.97
Tahitian	1.00	1.00	1.00	1.00
Tagalog	0.98	0.99	0.99	0.99
Tongan	1.00	1.00	0.94	0.96
Tuvaluan	1.00	1.00	0.98	1.00
Waray	1.00	1.00	1.00	1.00
Wallisian	1.00	1.00	1.00	1.00
Yapese	1.00	1.00	1.00	1.00

Table 11: Stability of Compressed Models (5x)

9. Conclusions

This paper has evaluated and made available language identification models for previously unavailable Austronesian languages, both for identifying the language of documents as well as for identifying code-switching within documents. We have shown that the reduction of performance is minimal as the inventory of languages is systematically increased from 200 to 800. This work represents a significant advance in the availability of NLP resources for Austronesian languages.

10. Bibliographical References

Agić, Ž. and Vulić, I. (2019). JW300: A Wide-Coverage Parallel Corpus for Low-Resource Languages. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210. Association for Computational Linguistics, jul.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-crawled Corpora. *Language Resources and Evaluation*, 43(3):209–226.

Benko, V. (2014). Aranea Yet Another Family of (Comparable) Web Corpora. In *Proceedings of 17th International Conference Text, Speech and Dialogue.*, pages 257–264. Springer.

Boyce, M. (2006). *A corpus of Modern Spoken Māori.* Ph.D. thesis, Victoria University of Wellington.

Brown, R. (2014). Non-linear mapping for improved identification of 1300+ languages. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 627–632.

Burdick, L., Kummerfeld, J. K., and Mihalcea, R. (2021). Analyzing the surprising variability in word embedding stability across languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5891–5901, Online and Punta Cana, Dominican Republic, November. Association for Computational Linguistics.

Chakravarthi, B. R., Mihaela, G., Ionescu, R. T., Jauhiainen, H., Jauhiainen, T., Lindén, K., Ljubešić, N., Partanen, N., Priyadharshini, R., Purschke, C., Rajagopal, E., Scherrer, Y., and Zampieri, M. (2021). Findings of the VarDial Evaluation Campaign 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–11, Kiyv, Ukraine, apr. Association for Computational Linguistics.

Dunn, J. and Adams, B. (2020). Geographically-balanced Gigaword corpora for 50 language varieties. In *Proceedings of the Language Resources and Evaluation Conference*, pages 2528–2536. European Language Resources Association, May.

Dunn, J., Li, H., and Sastre, D. (2022). Predicting embedding reliability in low-resource settings using corpus similarity measures. In *Proceedings of the 13th International Conference on Language Resources and Evaluation*. European Language Resources Association.

Dunn, J. (2020). Mapping languages: the Corpus of Global Language Use. *Language Resources and Evaluation*, 54:999–1018.

Goldhahn, D., Eckart, T., and Quasthoff, U. (2012). Building Large Monolingual Dictionaries at the Leipzig Corpora Collection From 100 to 200 Languages. In *Proceedings of the Eighth Conference on Language Resources and Evaluation*, pages 759–765. European Language Resources Association.

Jaech, A., Mulcaire, G., Hathi, S., Ostendorf, M., and Smith, N. (2016). Hierarchical Character-Word Models for Language Identification. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, pages 84–93. Association for Computational Linguistics, nov.

Jauhiainen, T., Lindén, K., and Jauhiainen, H. (2017a). Evaluating HeLI with Non-Linear Mappings. In *Proceedings of the Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–108. Association for Computational Linguistics, apr.

Jauhiainen, T., Lindén, K., and Jauhiainen, H. (2017b). Evaluation of language identification methods using 285 languages. In *Proceedings of the Nordic Conference on Computational Linguistics*, pages 183–191, Gothenburg, Sweden, may. Association for Computational Linguistics.

Jauhiainen, T., Lui, M., Zampieri, M., Baldwin, T., and Lindén, K. (2019). Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, 65.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., and Mikolov, T. (2016). Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431.

Kocmi, T. and Bojar, O. (2017). LanideNN: Multilingual Language Identification on Character Window. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 927–936. Association for Computational Linguistics, apr.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 923–929. European Language Resources Association, may.

Lui, M. and Baldwin, T. (2011). Cross-domain Feature Selection for Language Identification. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 553–561. Asian Federation of Natural Language Processing, nov.

Majlǐs, M. (2012). Yet Another Language Identifier. In *Proceedings of the EACL Student Research Workshop*, pages 46–54. Association for Computational Linguistics.

Malmasi, S. and Dras, M. (2017). Feature Hashing for Language and Dialect Identification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 399–403.

Molina, G., AlGhamdi, F., Ghoneim, M., Hawwari, A.,

Rey-Villamizar, N., Diab, M., and Solorio, T. (2016). Overview for the Second Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas, nov. Association for Computational Linguistics.

Solorio, T., Blair, E., Maharjan, S., Bethard, S., Diab, M., Ghoneim, M., Hawwari, A., AlGhamdi, F., Hirschberg, J., Chang, A., and Fung, P. (2014). Overview for the First Shared Task on Language Identification in Code-Switched Data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar, oct. Association for Computational Linguistics.

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the International Conference on Language Resources and Evaluation*, page 2214–2218. European Language Resources Association.

11. Language Resource References

Dunn, J. (2022). *Pacific Code-Switch: Python Package*. https://github.com/jonathandunn/pacific_CodeSwitch.

Dunn, J. (2022). *Pacific Language Identification Models*. https://www.jdunn.name/corpora.